

Scheduling Algorithm Design in Multiuser Wireless Networks

by

Yi Chen

B.Sc., Northwestern Polytechnical University, 2008

M.Sc., Northwestern Polytechnical University, 2011

A Dissertation Submitted in Partial Fulfillment of the  
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

© Yi Chen, 2016

University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Scheduling Algorithm Design in Multiuser Wireless Networks

by

Yi Chen

B.Sc., Northwestern Polytechnical University, 2008

M.Sc., Northwestern Polytechnical University, 2011

Supervisory Committee

---

Dr. L. Cai, Supervisor

(Department of Electrical and Computer Engineering)

---

Dr. X.D. Dong, Departmental Member

(Department of Electrical and Computer Engineering)

---

Dr. S. Ganti, Outside Member

(Department of Computer Science)

## Supervisory Committee

---

Dr. L. Cai, Supervisor  
(Department of Electrical and Computer Engineering)

---

Dr. X.D. Dong, Departmental Member  
(Department of Electrical and Computer Engineering)

---

Dr. S. Ganti, Outside Member  
(Department of Computer Science)

---

## ABSTRACT

In this dissertation, we discuss throughput-optimal scheduling design in multiuser wireless networks. Throughput-optimal scheduling algorithm design in wireless systems with flow-level dynamics is a challenging open problem, especially considering that the majority of the Internet traffic are short-lived TCP controlled flows. In future wireless networks supporting machine-to-machine and human-to-human applications, both short-lived dynamic flows and long-lived persistent flows coexist. How to design the throughput-optimal scheduling algorithm to support dynamic and persistent flows simultaneously is a difficult and important unsolved problem.

Our work starts from how to schedule short-lived dynamic flows in wireless systems to achieve throughput-optimality with queue stability. Classic throughput-optimal scheduling algorithms such as the Queue-length based Maxweight scheduling algorithm (QMW) cannot stabilize systems with dynamic flows in practical communication networks. We propose the Head-of-Line (HOL) access delay based scheduling algorithm (HAD) for flow-level dynamic systems, and show that HAD is able to obtain throughput-optimality which is validated by simulation.

As the Transmission Control Protocol (TCP) is the dominant flow and congestion control protocol for the Internet nowadays, we turn our attention to the compatibility

between throughput-optimal schedulers and TCP. Most of the existing throughput-optimal scheduling algorithms have encountered unfairness problem in supporting TCP-controlled flows, which leads to undesirable network performance. Motivated by this, we first reveal the reason of the unfairness problem, then study the compatibility between HAD and TCP with different channel assumptions, and finally analyze the mean throughput performance of HAD. The result shows that HAD is compatible with TCP.

Since the assumption of an infinite buffer size in the existing theoretical analysis of throughput-optimality is not practical, we analyze the queueing behaviour of the proposed throughput-optimal scheduling algorithm to provide useful guidelines for real system design by using the Markov chain analytic model. We propose the analytic model for the queueing and delay performance for the HAD scheduler, and then further develop an approximation approach to reduce the complexity of the model.

Finally, we propose a throughput-optimal scheduling algorithm for hybrid wireless systems with the coexistence of persistent and dynamic flows. Then, to generalize the throughput-optimal scheduling, the control function in the scheduling rule is extended from a specific one to a class of functions, so that the scheduling design can be more flexible to make a tradeoff between delay, fairness, etc. We show that the hybrid wireless networks with coexisting persistent flows and dynamic flows can be stabilized by our proposed scheduling algorithm which can obtain throughput-optimality.

In summary, we solve the challenging problem of designing throughput-optimal scheduling algorithm in wireless systems with flow-level dynamics. Then we show that our algorithm can support TCP regulated flows much better than the existing throughput-optimal schedulers. We further analyze the queueing behaviour of the proposed algorithm without the assumption of infinite buffer size that is often used in the throughput-optimality analysis in the literature, and the result provides a guideline for the implementation of our algorithm. At last, we generalize the proposed scheduling algorithm to support different types of flows simultaneously in practical wireless networks.

# Contents

<b>Supervisory Committee</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Abbreviations</b>	<b>x</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Dedication</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation . . . . .	3
1.3 Research Objectives and Contributions . . . . .	4
1.3.1 HOL Access Delay Based Scheduling (HAD) for Flow-level Dynamics . . . . .	4
1.3.2 On Achieving Fair and Throughput-optimal Scheduling for TCP Flows . . . . .	5
1.3.3 Queueing Behavior and Delay Analysis of HAD . . . . .	6
1.3.4 Scheduling Design for Coexistence of Persistent and Dynamic Flows . . . . .	7
1.4 Agenda . . . . .	8
1.5 Bibliographic Notes . . . . .	9
<b>2 HOL Access Delay Based Scheduling in Wireless Networks with Flow-Level Dynamics</b>	<b>10</b>

2.1	Introduction and Related Work . . . . .	11
2.2	System Model . . . . .	12
2.2.1	Arrival Model . . . . .	13
2.2.2	Channel Model . . . . .	13
2.2.3	System Capacity Region . . . . .	13
2.3	HOL Access Delay Based Scheduling . . . . .	15
2.4	Performance Evaluation . . . . .	21
2.5	Conclusion . . . . .	26
<b>3</b>	<b>On Achieving Fair and Throughput-optimal Scheduling for TCP Flows in Wireless Networks</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	TCP Compatibility Problem . . . . .	28
3.2.1	System Capacity and Throughput-optimal Scheduling Algorithms	28
3.2.2	An Example . . . . .	29
3.2.3	F-D-MW . . . . .	31
3.2.4	Further Discussion . . . . .	32
3.3	System Model . . . . .	33
3.3.1	Networks with Flow-level Dynamics . . . . .	34
3.3.2	Channel Model . . . . .	34
3.3.3	Queueing Model . . . . .	34
3.4	HOL Access Delay Based Scheduling . . . . .	36
3.4.1	Throughput-optimality . . . . .	37
3.4.2	Fairness Analysis . . . . .	38
3.4.3	Throughput Analysis . . . . .	43
3.5	Performance Evaluation . . . . .	44
3.5.1	Network Setting . . . . .	44
3.5.2	Homogeneous Networks with HAD . . . . .	46
3.5.3	Heterogeneous Networks with HAD . . . . .	49
3.5.4	System Stability . . . . .	55
3.6	Conclusion . . . . .	56
<b>4</b>	<b>Queueing Behavior Analysis of HAD Scheduling</b>	<b>58</b>
4.1	Introduction . . . . .	58
4.2	Queueing Behavior Analysis . . . . .	59

4.2.1	State-dependent Markov Model . . . . .	60
4.2.2	Approximation . . . . .	62
4.3	Performance Evaluation . . . . .	64
4.4	Conclusion . . . . .	66
<b>5</b>	<b>Throughput-optimal Scheduling for Hybrid Systems</b>	<b>68</b>
5.1	Introduction . . . . .	68
5.2	Related Works . . . . .	69
5.3	System Model . . . . .	70
5.3.1	Arrival Model . . . . .	71
5.3.2	Channel Model . . . . .	72
5.3.3	System Capacity Region and Throughput-optimality . . . . .	72
5.4	QHAD Scheduling Algorithm . . . . .	73
5.4.1	QHAD . . . . .	74
5.4.2	Stability with QHAD . . . . .	74
5.5	HADGe Scheduling Algorithm . . . . .	79
5.6	Performance and Discussion . . . . .	84
5.6.1	QHAD . . . . .	84
5.6.2	HADGe . . . . .	89
5.7	Conclusion . . . . .	89
<b>6</b>	<b>Conclusions and Future Research Issues</b>	<b>91</b>
6.1	Conclusions . . . . .	91
6.2	Future Research Issues . . . . .	92
	<b>Bibliography</b>	<b>94</b>

# List of Figures

Figure 2.1	System illustration of class- $k$ flows from slot $t$ to $t + 1$ . . . . .	14
Figure 2.2	Number of flows $N(t)$ with $\rho = 0.999$ . . . . .	22
Figure 2.3	System backlog $ Q(t) $ with $\rho = 0.999$ . . . . .	23
Figure 2.4	Average queue length with $\rho = 0.999$ . . . . .	23
Figure 2.5	Number of flows $N(t)$ with different $\rho$ at $t = 50,000$ . . . . .	24
Figure 2.6	Backlog $ Q $ with different $\rho$ at $t = 50,000$ . . . . .	25
Figure 3.1	Incompatibility between TCP and QMW scheduling. . . . .	30
Figure 3.2	The downlink of a wireless network with different classes of flows. . . . .	33
Figure 3.3	Network topology. . . . .	45
Figure 3.4	Throughput performance in the 2-flow homogeneous network. . . . .	47
	(a) HAD . . . . .	47
	(b) QMW . . . . .	47
	(c) F-D-MW . . . . .	47
Figure 3.5	Throughput performance in the 8-flow homogeneous network. . . . .	48
	(a) HAD . . . . .	48
	(b) QMW . . . . .	48
	(c) F-D-MW . . . . .	48
Figure 3.6	HOL access delay ratio in the heterogeneous network. . . . .	51
	(a) Without channel variations . . . . .	51
	(b) With channel variations . . . . .	51
Figure 3.7	Throughput performance in the 8-flow heterogeneous network with channel variations. . . . .	52
	(a) HAD . . . . .	52
	(b) MR . . . . .	52
	(c) QMW . . . . .	52
	(d) F-D-MW . . . . .	52
Figure 3.8	Jain's fairness index in terms of channel occupation. . . . .	54

(a) Homogeneous network . . . . .	54
(b) Heterogeneous network . . . . .	54
Figure 3.9 System throughput of different algorithms with increasing number of flows. . . . .	55
Figure 3.10 Throughput-optimality test. . . . .	56
Figure 4.1 Markov chain of $N(t)$ . . . . .	60
Figure 4.2 Number of flows in the system with varying $\rho$ . . . . .	65
Figure 4.3 Average queue delay with varying $\rho$ . . . . .	66
Figure 5.1 System backlog, persistent flow workload 30%. . . . .	85
Figure 5.2 System backlog, persistent flow workload 50%. . . . .	85
Figure 5.3 System backlog, persistent flow workload 80%. . . . .	85
Figure 5.4 Number of flows, persistent flow workload 30%. . . . .	86
Figure 5.5 Number of flows, persistent flow workload 50%. . . . .	86
Figure 5.6 Number of flows, persistent flow workload 80%. . . . .	86
Figure 5.7 System backlog, persistent flow workload 95%. . . . .	87
Figure 5.8 Number of flows, persistent flow workload 95%. . . . .	88
Figure 5.9 Number of flows for $f_1(H_i(t)) = H_i(t)^{1/2}$ . . . . .	90
Figure 5.10 Number of flows for $f_2(H_i(t)) = \log(H_i(t) + 1)$ . . . . .	90
Figure 5.11 Number of flows for $f_3(H_i(t)) = \log \log(H_i(t) + e)$ . . . . .	90

## List of Abbreviation

<b>ACK</b> .....	ACKnowledgement packet
<b>BS</b> .....	Base Station
<b>COF</b> .....	Control Objective Function
<b>CSMA</b> .....	Carrier Sense Multiple Access
<b>cwnd</b> .....	Congestion Window
<b>F-D-MW</b> .....	Flow Delay MaxWeight scheduler
<b>HAD</b> .....	Head-of-line Access Delay based scheduler
<b>HADGe</b> .....	General HAD scheduler
<b>HOL</b> .....	Head-Of-Line
<b>LTE</b> .....	Long Term Evolution
<b>MAC</b> .....	Media Access Control
<b>MR</b> .....	MaxRate scheduler
<b>MSS</b> .....	Maximum Segment Size
<b>MU</b> .....	Mobile User
<b>PF</b> .....	Proportional Fairness scheduler
<b>QHAD</b> .....	Queue-length associated HAD scheduler
<b>QMW</b> .....	Queue-length based MaxWeight scheduler
<b>SYN</b> .....	SYNchronization packet
<b>TCP</b> .....	Transmission Control Protocol
<b>UE</b> .....	User Equipment

## ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to supervisor Prof. Lin Cai for her support of my research and study at University of Victoria in the past years, for her patience, inspiration and technical advice. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank Prof. Xiaodai Dong and Prof. Sudhakar Ganti for serving as my thesis committee, Prof. Lei Ying from the School of Electrical, Computer and Energy Engineering at Arizona State University to serve as my external examiner. My sincere thanks also goes to Prof. Jianping Pan for his valuable comments, insights and guidance.

I thank my fellow labmates in Communication Networks Lab: Dr. Zhe Yang, Dr. Yuanqian Luo, Dr. Siyuan Xiang, Dr. Xuan Wang, Dr. Lei Zheng, Dr. Min Xing, Dr. Kan Zhou, Zhe Wei, Haoyuan Zhang, Dr. Jianping He, Dr. Yongmin Zhang, Yue Li, Yuanzhi Ni, Jiayi Chen and Mohammad Ghasemahmadi for the stimulating discussions and for the funs we have in the last years. Also I thank my friends at Northwestern Polytechnical University for the encouragement.

I would also like to thank my significant other, Yue Yin, for the encouragement and criticism, and most importantly, your accompany along the time. Last and certainly not least, I would like to thank my parents, for their endless love and support throughout my life.

*Yi Chen, Victoria, BC, Canada*

DEDICATION

To my parents,  
And all of my friends,  
Without whom none of this would be possible

# Chapter 1

## Introduction

In this dissertation, we discuss the design of throughput-optimal scheduling algorithms in multiuser wireless networks. First, the Head-of-line (HOL) Access Delay based scheduling algorithm (HAD) is proposed in our work for wireless networks with flow-level dynamics, and its throughput-optimality compared with the existing schedulers is demonstrated. Second, motivated by the unfairness problem of the existing throughput-optimal schedulers when scheduling the Transmission Control Protocol (TCP) controlled flows, the compatibility between TCP and HAD is analyzed and presented. Third, to fill the gap between theoretical analysis and implementation, the queueing behaviour and the average delay performance of HAD are analyzed using a Markov chain model. Finally, the throughput-optimal scheduling algorithm for hybrid systems where persistent and dynamic flows coexist is proposed and investigated, which is then generalized from one specific algorithm to a class of algorithms. In this chapter, we will discuss the background and motivation of our work, and introduce the research objectives and contributions.

### 1.1 Background

In wireless networks, data flows are competing for limited wireless channel resources. Hence how to efficiently and fairly schedule the transmission for data flows using limited bandwidth is one of the key problems in the operation of wireless systems.

In this dissertation, we mainly consider a one-hop multiuser wireless system working in slotted time with one base station and a number of users. The essential job of a scheduling algorithm is to determine in every time slot which user should be selected

to transmit. One key performance metric of scheduling algorithm design is the queue stability. Assume that data flows in a wireless communication system have infinite buffer size. When  $t \rightarrow \infty$ , if the time average of the total system queue length is finite, the queues are considered to be stable, i.e., the corresponding scheduler is able to stabilize the system. The higher the data arrival rate is, the harder the scheduling algorithm can stabilize the queues, and there is a boundary of traffic arrival rate vector for each system, beyond which there is no scheduling algorithm that can stabilize the system. The region bounded by this boundary is described as the *capacity region* of the system. At the viewpoint of queueing network and stability, in a given network, the arrival traffic rate should be smaller than the service rate in order to stabilize the queues in the network. Correspondingly, in a wireless system, the sufficient condition to stabilize the system is that the arrival rate lies inside the capacity region, which gives the upper bound of the achievable arrival rate that can be supported by any possible scheduling algorithm, either offline or online, with queue stability given a system channel profile.

However, when the above sufficient condition of system stability is satisfied, not all scheduling algorithms can lead to system stability. Typically, a scheduling algorithm is throughput-optimal if it can always achieve the network queue stability given any traffic arrival rate vector that lies strictly within the capacity region. For example, the work in [58] shows that there are a number of widely used scheduling algorithms such as the Proportional Fair scheduling algorithm that cannot stabilize the system when the traffic rate lies in the capacity region. Motivated by this, different kinds of throughput-optimal schedulers have been proposed.

The calculation of the capacity region may not be the same when dealing with different systems. In this dissertation, the type of a system is determined by its flows. We consider two types of flows in our work. One is *persistent flow*, and the other is *dynamic flow*. Persistent flows are long-lived and have continuous data injection. For machine-type applications, such as sensor networks or vehicular networks, persistent flows may exist. The number of persistent flows is a constant value, and does not change over time. Although at certain point, one persistent flow may have zero backlogged packet in the buffer, it remains in the system and waits for the future arrival traffic. On the other hand, dynamic flows have finite amount of service requests upon arrival in the network, and leave the system once the demanded services are fulfilled. Since dynamic flows arrive and leave the system over time, the number of dynamic flows in the system may change from one slot to the next. Dynamic flows are

commonly observed in human-to-human communication applications. The examples of applications with dynamic flows include the email/text messages delivered from one people to another, web browsing, etc.

## 1.2 Motivation

The classic Queue-length based MaxWeight scheduling algorithm (QMW) [50] is the first throughput-optimal scheduling algorithm in the literature for systems with persistent flows only. Due to its desirable feature of throughput-optimality with the simple scheduling rule, it attracted a lot of research interests and has been extensively studied in the literature. However, it fails to achieve throughput-optimality with the presence of flow-level dynamics [52]. How to design throughput-optimal scheduling algorithms for systems of dynamic flows is a critical and practical issue given that dynamic flows are commonly observed in real life.

On the other hand, TCP is the dominant flow and congestion control protocol of the Internet nowadays, and will be remaining so in the foreseeable future. The existing throughput-optimal scheduling algorithms, however, are not compatible with TCP. More specifically, the coexistence of TCP and these algorithms leads to severe unfair resource allocation for flows. The Proportional Fairness scheduling algorithm (PF) is widely adopted in the industry to achieve fair scheduling among different flows, but the existing results show that PF is not throughput-optimal. It is very hard for a throughput-optimal scheduling algorithm to be widely used in the Internet unless it can be compatible with TCP.

In the theoretical analysis of all the throughput-optimal schedulers, it is always assumed that the system buffer size is infinite. However, this assumption is not valid in practice. Considering the implementation of a scheduling algorithm in real systems, it is important to analyze the queueing behaviour and the delay performance of the given algorithm to avoid potential performance deterioration.

In the existing works, when analyzing the throughput-optimal scheduling algorithms, most of the system models consider persistent flows only, or dynamic flows only. Few works have been completed to investigate the throughput-optimal scheduling for the coexistence of persistent and dynamic flows. The approach of separating the two types of flows and scheduling them independently is not the best choice, because separating the resources for two types of flows will result in a lower multiplexing gain and lower efficiency. How to design a throughput-optimal scheduler that is able

to deal with persistent and dynamic flows simultaneously remains an open problem.

The above mentioned issues motivate this dissertation.

## 1.3 Research Objectives and Contributions

This dissertation has made the following main contributions: it designed the HAD scheduling algorithm aiming to achieve the throughput-optimality for wireless systems with the presence flow-level dynamics; analyzed the compatibility issue between various throughput-optimal schedulers and the TCP protocol; derived the analytic model for the queueing behaviour and delay performance of the proposed HAD scheduling algorithm, which is also a performance reference of the other throughput-optimal schedulers; designed a throughput-optimal scheduling algorithm for systems with the coexistence of both persistent and dynamic flows, which is then extended to a general form of scheduling rule for throughput-optimality. The details of the research objectives and contributions are discussed below.

### 1.3.1 HOL Access Delay Based Scheduling (HAD) for Flow-level Dynamics

The most fundamental resource in a wireless network is the physical spectrum of the wireless channel, which is limited. In order to provide high data-rate services for the data flows in a multiuser network, an efficient scheduling algorithm should be able to fully use the multiuser diversity gain so that the utilization of the spectrum resource can be maximized.

Consider the downlink of a time-slotted wireless network, where only one user can be scheduled for transmission in one slot. Due to the channel fading and interference, the channel condition is changing over time. Hence the scheduling and resource allocation algorithm should consider both the real-time user demand and the dynamic channel condition. Scheduling algorithms only explore the dynamic feature of wireless channel may suffer from performance degradation [58].

The pioneer works of Tassiulas and Ephremides [49–51] proposed QMW and proved that QMW is a throughput-optimal strategy. QMW prioritizes the flows with the largest product of the queue length (backlog) and the current transmission rate. It became a popular research topic since the scheduling strategy of QMW is simple while it can achieve throughput-optimality, and the properties of QMW has

been extensively studied in the literature [3, 37]. Other works extended throughput to other metrics such as the delay performance [39], energy consumption [36, 41], and fairness [17, 40], etc. Although QMW presents desirable throughput performance, one necessary condition is that the network adopting QMW consists of a fixed number of persistent flows only. If some or all of the flows are dynamic which have a finite amount of data to transmit (flows are short-lived), and the number of users in the system is not a constant, i.e., the system has flow-level dynamics, the QMW scheduling algorithm is not throughput-optimal [52]. The queue-length based scheduling has other variations in the literature besides QMW, and it has been further shown that all these algorithms are no longer throughput-optimal in systems with flow-level dynamics.

Considering that dynamic flows are ubiquitous among human-to-human communications, it is valuable to design a scheduling algorithm that is throughput-optimal for systems with flow-level dynamics. Different from the existing works mentioned above which are mainly queue-length-based, we aim to propose an online algorithm, namely HAD, using the head-of-line (HOL) access delay rather than queue length in the control function of the scheduling rule. In this dissertation, we investigate the condition for a scheduling algorithm to achieve throughput-optimality, based on which we prove that, with the sufficient usage of the multiuser diversity gain, our proposed scheduling algorithm can achieve system queue stability with any arrival traffic rate in the capacity region.

### 1.3.2 On Achieving Fair and Throughput-optimal Scheduling for TCP Flows

TCP is the dominant transport layer protocol in the Internet, and it has been extensively studied in the literature [4]. It performs decently even with the scale of the Internet growing by several orders of magnitude in the past three decades. The basic congestion control mechanism of TCP was designed to probe for the available bandwidth while maintaining certain level of fairness among co-existing flows. In practice, all TCP variants, both the widely used loss-based variants and the other delay-based ones, have their own clock timing, which relies on the end-to-end acknowledgement packets (ACKs). Based on the received ACKs, a TCP sender determines whether and how many packets should be injected into the network by updating the size of the congestion window (*cwnd*). This protocol was originally designed for wired net-

works. As an increasing number of wireless devices are involved in the Internet, it becomes increasingly important to investigate the compatibility between TCP and the lower layer wireless scheduling algorithms [6, 56]. The adjustment of *cwnd* was designed to achieve fairness among all TCP flows. However, the control mechanism of some existing throughput-optimal scheduling algorithms conflicts with TCP congestion control. How TCP can be compatible with throughput-optimal scheduling algorithms in wireless networks when multiple users share a radio link is the research interest of our work.

In the link layer, scheduling algorithms in wireless networks have been extensively studied in the literature. For example, considering the differentiated services, a number of scheduling algorithms and MAC protocols have been designed according to the QoS requirements of various applications [61]. On the other hand, considering the wireless channel dynamics, opportunistic scheduling algorithms can exploit the multiuser diversity gain to improve the overall performance.

Unfortunately, some TCP flows will suffer from a severe unfairness or starvation problem if most of the existing throughput-optimal scheduling algorithms are used [47], which gives the motivation of our study. In this work, we reveal why the existing throughput-optimal scheduling algorithms are not compatible with TCP flows. Second, we apply the proposed HAD scheduling algorithm to schedule TCP flows in wireless networks. We prove that it can schedule TCP flows under homogeneous or heterogeneous channel conditions with certain level of fairness guarantee. Performance evaluations using OMNeT++4 have been conducted to validate our theoretical findings, which show that the HAD algorithm can outperform the other throughput-optimal scheduling algorithms in supporting TCP flows in wireless networks.

### 1.3.3 Queueing Behavior and Delay Analysis of HAD

Given an arbitrary traffic arrival rate in the capacity region, a throughput-optimal scheduling algorithm can maintain the queue stability, i.e., the time average of the amount of backlogged packets in the system can be bounded. When a system of persistent flows is studied, to achieve queue stability, each individual flow in the system must have a finite queue length, which is achievable by QMW. In systems of dynamic flows working in slotted time, the system stability requires the number of flows  $N(t)$  in the system to be finite in any time slot. In the literature, the

typical approach to prove that a given scheduler is throughput-optimal for flow-level dynamics is to find the existence of an upper bound of  $N(t)$ . Whether the upper bound is a loose or tight one does not affect the result of throughput-optimality. Since the upper bound is not explicitly calculated, the existing analyses always assume that the buffer size is infinite.

In practice, however, it is impossible to implement an infinite buffer. As a result, an arbitrarily designed buffer size may be too small resulting in potential performance deterioration, or too large such that the cost of building a system is too high leading to unnecessary waste. This motivates us to investigate the queueing behavior of our proposed HAD scheduler to give a guideline of implementation. In our work, we adopt the Markov chain analytic model, and obtain the explicit description of the Markov state probability, the stationary state, etc., to analyze HAD. Since the result of directly solving the Markov chain model is very complex, to reduce the complexity of the analysis, we further study two approximation methods corresponding to different arrival traffic intensity. With the queueing behaviour analysis, the delay performance is also investigated. The simulation and analytic results match each other very well.

### 1.3.4 Scheduling Design for Coexistence of Persistent and Dynamic Flows

Although there have been a few solutions to design scheduling algorithms for systems with various flows, the majority of the existing works considered networks that exclusively include the flows of only one type, i.e., either persistent flows or dynamic flows, rather than both of them. Few works have been done jointly considering both types of flows. However, the coexistence of the persistent and dynamic flows cannot be ignored in practice, e.g., in 5G cellular systems, both machine-to-machine and human-to-human applications share the same spectrum, and thus it is important to design the corresponding throughput-optimal scheduling algorithms. The approach of separating the two types of flows and scheduling them independently is not the best choice, because separating the resources for two types of flows will result in a lower multiplexing gain and lower efficiency. On the other hand, the type of each flow is a prerequisite for the scheduler if the separate scheduling approach is adopted, and it may be costly to distinguish persistent and dynamic flows in reality. Meanwhile, the recent result for the hybrid system scheduling is inadequate for real networks because the rate variation of the wireless channel is not sufficiently considered, and

the delay performance is not desirable. All of the above motivate us to investigate the scheduling algorithm which is flow-type-insensitive for hybrid systems with both persistent and dynamic flows. To make the scheduler more flexible and adaptive to different systems, we further study how to generalize the proposed scheduling design.

## 1.4 Agenda

This section provides a map of the rest of the dissertation to show the reader where and how it validates the claims previously made.

Chapter 2 discusses the throughput-optimal scheduling algorithm in wireless systems with flow-level dynamics. We obtain the condition to achieve system queue stability, propose and analyze our simple online scheduling algorithm, the HAD scheduling algorithm. The advantage of HAD over the other existing works is explained. Simulation results are given to validate our analytic results.

Chapter 3 describes the reason of the incompatibility between the existing scheduling algorithms and TCP, and then investigates the properties of the HAD scheduler we proposed. We prove that the proposed HAD can fairly schedule TCP flows in wireless networks with time-varying channel conditions while achieving throughput-optimality with flow-level dynamics. Simulations using OMNeT++ 4 have been conducted to validate our analytic results, and compare the performance of different scheduling algorithms comprehensively.

Chapter 4 gives the approach of analyzing the queueing behaviour and delay performance of HAD. The explicit expression of the algorithm performance has been obtained. Further simplification of the calculation has been made in this chapter by proposing two approximation methods to reduce the complexity of the computation.

Chapter 5 is where the definition of capacity region is given for hybrid systems with the coexistence of persistent and dynamic flows, followed by our proposed online scheduling algorithm to achieve throughput-optimality for hybrid systems. We also generalize the scheduling algorithm to provide better flexibility and adaptiveness in scheduling algorithm design. The simulation result not only validates the throughput-optimality of the proposed schedulers in various types of network settings, but also offers the interesting observation that the offline MaxRate scheduler fails to achieve throughput-optimality in hybrid systems.

Chapter 6 concludes this dissertation.

## 1.5 Bibliographic Notes

Most of the works reported in this dissertation have appeared or been submitted as research papers. The works in Chapter 2 have been published in [9]. The works in Chapter 3 have been accepted in [12]. The works in Chapter 4 have been accepted as [11]. The works in Chapter 5 have been prepared for submission as [13].

## Chapter 2

# HOL Access Delay Based Scheduling in Wireless Networks with Flow-Level Dynamics

Scheduling algorithm design is an important and challenging problem which influences whether wireless networks can efficiently use the limited channel resource or not. The queue-length based QMW has been proved to be throughput-optimal by L. Tassiulas, etc. It was first proposed in [50], and has been extensively studied in the literature [49, 51]. QMW schedules the flows with the maximal product of the current channel transmission rate and the flows backlog in each time slot. If the number of flows is fixed in the system, and the traffic generated by each flow is long-lived, i.e., the per-flow traffic is infinitely long, the QMW scheduler is proved to be throughput-optimal. A throughput-optimal scheduling algorithm can stabilize the queues in the system if the arrival rates are within the system capacity region. The application of throughput-optimal scheduling can be found in [7, 59, 63]. However, if some or all of the flows have a finite amount of data to transmit (flows are short-lived), and the number of flows in the system is not a constant, i.e., the system has flow-level dynamics, the QMW scheduling algorithm is not throughput-optimal. The other queue length based scheduling algorithms include the Exponential rule [48] and the Log rule [44], etc. It has been shown that all these algorithms are no longer throughput-optimal in the systems with flow-level dynamics [52]. In practice, the number of flows may vary dynamically. Thus, it is valuable to design a scheduling algorithm to be throughput-optimal in such systems, which motivates the work in this chapter.

## 2.1 Introduction and Related Work

Considering flow-level dynamics, several scheduling algorithms have been proposed in the past few years. In [52], the authors first explained the instability of the QMW scheduling algorithm in wireless systems with flow-level dynamics, and designed a new algorithm that can stabilize such systems. However, this algorithm requires the prior knowledge of the channel rate distribution and the incoming traffic of each flow. Thus it is too complicated to implement. The authors further investigated the spacial inefficiency of the QMW scheduling in [53, 54] and proposed a grouping strategy as a possible solution. More examples and numerical experiments about the instability and inefficiency of the QMW scheduling in multihop networks were discussed in [24].

A Flow (File) Delay based MaxWeight (F-D-MW) scheduling algorithm was proposed in [46] aiming to stabilize the system with dynamic flows. However, it only proved that F-D-MW is throughput-optimal if the distribution of the channel transmission rate of each user is i.i.d., and lacked the results to show the performance in a heterogeneous channel network. F-D-MW uses the flow (or HOL file) delay as the weight of the channel transmission rate, and thus the flows (HOL files) entering the system earlier will always have a higher priority for transmission. In this situation, it may be difficult to provide good quality-of-service for the flows entering the system later with stringent delay requirement. A MaxRate (MR) scheduling algorithm was designed in [33] for flow-level dynamic systems. In each time slot, the MR scheduling algorithm opportunistically selects the user which has the best channel condition according to each flow's channel profile. The designed algorithm requires either the prior knowledge of the channel and traffic distributions to guarantee the throughput-optimality, which make the algorithm an offline one, or a sufficiently long learning period to learn the best channel rate seen by the user so far to make it an online algorithm.

Delay based scheduling has been investigated in many existing works. Using delay as the weight in a MaxWeight type of scheduling algorithm was introduced in [35], and has been extended to providing throughput-optimal scheduling algorithms for wireless networks [18, 48]. The utility maximization using delay-based scheduling was studied in [38]. Based on the works above, a delay-based back-pressure routing for multihop wireless networks was developed in [23]. Regarding HOL delay based scheduling, [3] studied the HOL file delay based scheduling without flow-level dynamic and discussed a framework for stable scheduling algorithms. The throughput-optimality of

this scheduling algorithm was further investigated in [18]. However, none of these works considers to adopt the HOL access delay based scheduling in flow-level dynamic systems. What we also need to clarify here is that the head-of-line file delay, on which the existing works [23, 46] are focusing, is calculated from the moment when the HOL file (packet) arrives in the system, which is different from the concept of the HOL access delay in this work. Our definition of the head-of-line access delay will be given in this chapter, and the advantage of our work over the other delay based scheduling will also be explained later.

In this work, we investigate the scheduling algorithm based on the HOL access delay. Different from the F-D-MW in [46], we give the proof of the throughput-optimality without the assumption of the i.i.d. distribution of the channel transmission rate. Unlike the QMW scheduling, the HOL access delay based scheduling has one desirable property for flow-level dynamic systems. Consider a flow with the last packet waiting for transmission. By adopting QMW, without new traffic arrival, the queue length of this flow will remain small, which will result in a long (or possibly infinite) delay of the flow since the other flows with a larger queue length will have a higher priority. This problem is often referred as the “last packet problem”, which will cause instability with flow-level dynamics. This problem is solved if our HOL access delay based scheme is used.

## 2.2 System Model

We consider a time-slotted heterogeneous wireless network with one base station (BS) and multiple mobile users (MUs). Each MU is associated with one or a few distinct dynamic flows, each of which is a traffic burst with finite number of bits. Flow size is defined as the size of traffic burst upon its arrival. Since the scheduling objective is each flow, we will use the concept of “flow” instead of “user” thereafter. A dynamic flow can enter the system at any time slot, and will leave the system after all the bits are transmitted.

The system has multiple classes of flows, which makes the system a heterogeneous network. Within each class, flows have i.i.d. arrival and channel rate distributions. Assume that all the flows can be assorted into  $K$  classes. The  $i$ -th flow of class- $k$  at time  $t$  is denoted by  $Q_{ki}(t)$ . The amount of the remaining bits of  $Q_{ki}(t)$  waiting for transmission at the beginning of time slot  $t$  is denoted by  $|Q_{ki}(t)|$  and called the residual bits. The number of flows of class- $k$  at the beginning of time slot  $t$  is  $N_k(t)$ ,

and the total number of flows at the beginning of time slot  $t$  is  $N(t) = \sum_{k=1}^K N_k(t)$ . Within each class, the flows are indexed by their arrival time.

### 2.2.1 Arrival Model

New flows can arrive at any time. Let  $A_k(t) \in \{0\} \cup \mathbb{Z}_+$  denote the number of class- $k$  flows arriving during time slot  $t$ , which is a random variable.  $A_k(\cdot)$  is i.i.d. with the mean  $\lambda_k = \mathbb{E}[A_k(1)]$ . We suppose that the scheduling decision is made at the beginning of every time slot, so all the flows that arrive after the beginning of slot  $t$  can only be scheduled at the beginning of slot  $t+1$ . Let  $B_{ki}(t)$  denote the flow size of the  $i$ -th class- $k$  flow which arrives during slot  $t$ . In class- $k$ , we assume that  $B_{ki}(t)$  is the i.i.d. copy of some integer random variable  $B_k$  and has a finite mean  $\beta_k = \mathbb{E}[B_k]$ . The second moments of  $A_k(\cdot)$  and  $B_k(\cdot)$  are both finite. We define  $|Q_k(t)| = \sum_{i=1}^{N_k(t)} |Q_{ki}(t)|$  as the class- $k$  backlog and  $|Q(t)| = \sum_{k=1}^K |Q_k(t)|$  as the system backlog.

### 2.2.2 Channel Model

Let  $r_{ki}(t)$  denote the transmission rate of the wireless channel at time  $t$  between  $Q_{ki}(t)$  and the BS. The unit of the channel rate is bit/slot. The BS can transmit at most  $r_{ki}(t)$  bits at time  $t$  for  $Q_{ki}(t)$ .  $r_{ki}(t)$  may vary over time as a result of fading. For class- $k$ , we assume  $r_{ki}(\cdot)$  are i.i.d. copies of positive integer random variable  $R_k$  with finite supports, i.e.,  $R_k \in \{R_{k1}, R_{k2}, \dots, R_{km_k}\}$ . Different classes may have heterogeneous channel condition distributions. The maximum possible transmission rate of the class- $k$  flows is defined as  $R_k^{\max} = \sup\{r : \mathbb{P}\{R_k = r\} > 0\}$ , and the maximum possible transmission rate of the system is defined as  $R^{\max} = \max_{1 \leq k \leq K} \{R_k^{\max}\}$ .

An example of class- $k$  dynamic flows of the network is illustrated in Fig. 2.1, which shows the evolution from time slot  $t$  to  $t+1$ . At the beginning of time slot  $t$ , there are four flows in the system. After that, there is one new flow which has the head-of-line access delay  $H_{B_1(t)} = 0$ . Suppose that  $Q_{k3}(t)$  is scheduled at time slot  $t$ , and  $r_{k3}(t) \geq |Q_{k3}(t)|$ ,  $Q_{k3}(t)$  will finish all its transmission and leave the system, and we have four flows in class- $k$  at the beginning of slot  $t+1$ .

### 2.2.3 System Capacity Region

Let  $\gamma_k$  represent the expected number of time slots that are required for the service of a class- $k$  flow if served with  $R_k^{\max}$ , and we have  $\gamma_k = \mathbb{E}\left[\frac{B_k}{R_k^{\max}}\right]$ . Let  $\rho_k = \lambda_k \gamma_k$  denote the

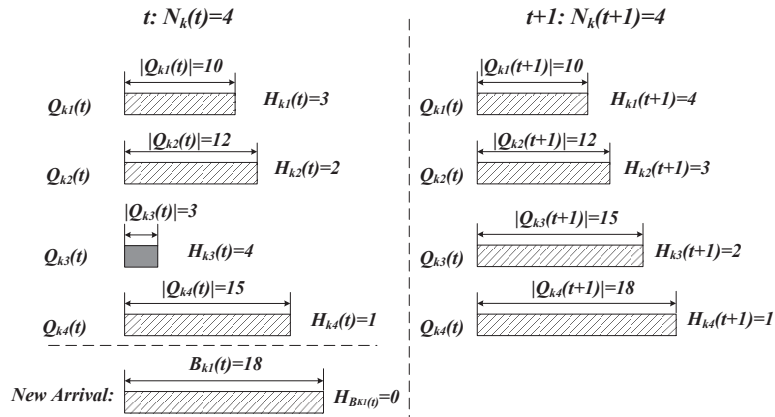


Figure 2.1: System illustration of class- $k$  flows from slot  $t$  to  $t + 1$

traffic intensity of class- $k$  flows, and  $\rho = \sum_{k=1}^K \rho_k$  denote the system traffic intensity. The system capacity region is defined as  $\mathcal{S} = \{(\lambda_1, \lambda_2, \dots, \lambda_K), (\gamma_1, \gamma_2, \dots, \gamma_K) : \rho < 1\}$ . For any arrival process that lies in the capacity region, if the system is strongly stable, i.e.,  $\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}|Q(t)| < \infty$ , then the corresponding scheduling algorithm is throughput-optimal.

Intuitively, for a system working in slotted time, if the system is stable, the total amount of the queued data should be finite at any time slot, while if unstable, the total amount of the queued data will grow into infinity when  $t \rightarrow \infty$  given infinite system buffer size. Considering that the physical meaning of the traffic intensity  $\rho$  is the average number of time slots that are required to transmit the arrival traffic in one time slot when the maximum possible transmission rate is always adopted, the sufficient condition for stability to be achievable is  $\rho < 1$  [52]. In other words, if the average amount of arrival traffic in one time slot can be transmitted in less than one time slot by the maximum possible transmission rate, there exists at least one scheduling algorithm to achieve system stability. If  $\rho > 1$ , on average more than one slot is required to transmit the amount of arrival data in one slot, and the residual data will accumulate into infinity over time which results in instability. From this perspective, the system capacity region is defined as  $\rho < 1$ , and any arrival rate that is in the capacity region can be stably transmitted by the throughput-optimal algorithms.

If the system has no flow-level dynamics, i.e., the number of flows is fixed, we can use  $|Q(t)|$  as the metric of the system stability. For the systems with flow-level

dynamics, each flow has a finite amount of data to transmit upon arrival, and leaves the system once all the data are transmitted. When the system stability is achieved, the number of flows in the system is finite, i.e.,  $N(t) < \infty$  at any time slot. If  $N(t) \rightarrow \infty$  when  $t \rightarrow \infty$ , we have  $|Q(t)| \rightarrow \infty$  as well, and thus the system is unstable. As a result, we can also use  $N(t)$  as the metric of the system stability when considering flow-level dynamics.

## 2.3 HOL Access Delay Based Scheduling

Since we consider a delay based scheduling algorithm, we give the definition of the Head-Of-Line (HOL) access delay which we will use in our scheduling.

**Definition 1** (The HOL Access Delay  $H_{ki}(t)$ ). *Let  $I_{ki}^H(t)$  denote the head bit in  $Q_{ki}(t)$  which will be the first bit to be transmitted. The HOL access delay of  $Q_{ki}(t)$  is defined as  $H_{ki}(t) = t - t_0$ , where  $t$  is the current time, and  $t_0$  is the time at which  $I_{ki}^H(t)$  becomes the first bit in  $Q_{ki}(t)$ .*

HOL access delay can be calculated according to the following equation:

$$H_{ki}(t+1) = (H_{ki}(t) + 1) (1 - \mathbb{1}_{\{Q_{ki}(t)\}}(t)), \quad (2.1)$$

where  $\mathbb{1}_{\{Q_{ki}(t)\}}(t)$  is the indicator function such that  $\mathbb{1}_{\{Q_{ki}(t)\}}(t) = 1$  only when  $Q_{ki}(t)$  is scheduled at time slot  $t$ . With the system model and the definition of HOL access delay in the above, we adopt the following HOL access delay based scheduling algorithm.

**Algorithm 1.** *HOL Access Delay Based MaxWeight Scheduling algorithm (HAD) seeks the flow  $\{k, i\}$  to transmit that satisfies the following condition at the beginning of time slot  $t$ :*

$$\{k, i\}^*(H_{ki}(t), r_{ki}(t)) \in \arg \max_{1 \leq k \leq K, 1 \leq i \leq N_k(t)} H_{ki}(t) \cdot r_{ki}(t), \quad (2.2)$$

*with uniform tie-breaking if there are a number of flows satisfying the condition. The scheduling decision is made in every time slot independently.*

Next we prove that, with flow-level dynamics, HAD scheduling is throughput-optimal, i.e., the system is stable with the HAD scheduling, by three steps. In the

first step, we explain that in an unstable system, there are countless flows that have infinite HOL access delays, and we investigate a sufficient condition for stability. Second, we prove one property of HAD scheduling algorithm when the system is unstable. Third, based on the above results, we further prove that a system with flow-level dynamics is stable when HAD is adopted so long as the traffic intensity lies inside the system capacity region.

**Theorem 1.** *Given an infinite buffer, for a flow-level dynamic multiuser wireless system, if the HAD scheduling algorithm cannot stabilize the system, there will be an infinite number of flows in the system that have infinite HOL access delays when the system time goes to infinity.*

*Proof.* Suppose that the system is unstable when  $t \rightarrow \infty$ , and then at least there is one flow with an infinite HOL access delay. If there is only one flow that is unstable associated with the infinite HOL access delay, and all the other flows have finite HOL access delays, the only flow with the infinite HOL access delay will be scheduled according to (2.2) at a certain time slot, for example  $t_1$ , and its HOL access delay in the next time slot  $t_1 + 1$  is 0. Because we have the assumption here that all the other flows have finite HOL access delay, we can come to the conclusion that at time slot  $t_1 + 1$ , all the flows have finite HOL access delays. This conclusion is contradicted with the instability condition that there is at least one flow with an infinite HOL access delay.

Similarly, we can prove that if there are only a finite number of flows associated with infinite HOL access delays, the system is also stable. Finally we can conclude that if the system is unstable, an infinite number of flows in the system must have infinite HOL access delays when  $t \rightarrow \infty$ .  $\square$

**Theorem 2.** *(Sufficient condition) Let  $r(t)$  denote the real transmission rate of the network at time  $t$ . If a class- $k$  flow  $Q_{ki}(t)$  is scheduled, i.e.,  $r(t) = r_{ki}(t)$ , the sufficient condition for the network with flow-level dynamics to be stable for any arrival rate that lies in the capacity region is*

$$\lim_{t \rightarrow \infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0. \quad (2.3)$$

*Proof.* We use the following Lyapunov function  $L(t) = (W(t))^2$  to prove Theorem 2, where  $W(t)$  is defined as the workload of the system at time  $t$ , i.e.,  $W(t) = \sum_{k=1}^K \sum_{i=1}^{N_k(t)} \left\lceil \frac{|Q_{ki}(t)|}{R_k^{\max}} \right\rceil$ . The workload is apparently a direct reflection of the total queue

length of the system. We define  $W_A(t) = \sum_{k=1}^K \sum_{i=1}^{A_k(t)} \left\lceil \frac{B_{ki}(t)}{R_k^{\max}} \right\rceil$  as the amount of the new workload injected in the network at time  $t$ , and  $W_R(t) = \sum_{k=1}^K \sum_{i=1}^{N_k(t)} \left\lceil \frac{r_{ki}(t)}{R_k^{\max}} \right\rceil \cdot \mathbb{1}_{\{Q_{ki}(t)\}}(t)$  as the decrease of the workload if  $Q_{ki}(t)$  is scheduled for transmission at time  $t$ , i.e.,  $r(t) = r_{ki}(t)$ , where  $\mathbb{1}_{\{Q_{ki}(t)\}}(t) = 1$  if  $Q_{ki}(t)$  is scheduled, and  $\mathbb{1}_{\{Q_{ki}(t)\}}(t) = 0$  otherwise. Based on the above notations, the evolution of the workload in the system can be described as  $W(t+1) = [W(t) + W_A(t) - W_R(t)]^+$ . Then we calculate the square of this equation and after some manipulation we can obtain

$$\begin{aligned}
& (W(t+1))^2 - (W(t))^2 \\
& \leq (W_A(t))^2 + (W_R(t))^2 - 2W(t)W_R(t) + \\
& \quad 2W_A(t)(W(t) - W_R(t)) \\
& \leq (W_A(t))^2 + (W_R(t))^2 - 2W(t)(W_R(t) - W_A(t)). \tag{2.4}
\end{aligned}$$

Since the arrival rates lie in the capacity region, and the second moments of the arrival rates are bounded, we can conclude that there exists a  $U = \mathbb{E}[(W_A(t))^2] + \mathbb{E}[(W_R(t))^2] < \infty$ . By taking the expectation of (2.4), we can calculate the Lyapunov drift of the Lyapunov function as follows:

$$\mathbb{E}[(W(t+1))^2] - \mathbb{E}[(W(t))^2] \leq U - 2\mathbb{E}[W(t)(W_R(t) - W_A(t))].$$

The above holds for all  $t \in \{0, 1, 2, \dots\}$ . Summing over  $t \in \{0, 1, \dots, T-1\}$  for some integer  $T > 0$  yields

$$\mathbb{E}[L(W(T))] - \mathbb{E}[L(W(0))] \leq U \cdot T - \sum_{t=0}^{T-1} \mathbb{E}[W(t)]\mathbb{E}[W_R(t) - W_A(t)].$$

Note that  $\mathbb{E}[L(W(T))] \geq 0$ , taking a lim sup yields

$$\begin{aligned} & \limsup_{T \rightarrow \infty} \frac{1}{T\varepsilon} \sum_{t=0}^T \left( \sum_{k=1}^K \sum_{i=1}^{N_k(t)} \mathbb{E}[W(t)] \right) \\ & \left( \sum_{k=1}^K \sum_{i=1}^{N_k(t)} \left\lceil \frac{r_{ki}(t)}{R_k^{\max}} \right\rceil \cdot \mathbb{1}_{\{Q_{ki}(t)\}}(t) - \sum_{k=1}^K \sum_{i=1}^{A_k(t)} \frac{B_{ki}(t)}{R_k^{\max}} \right) \\ & \leq \limsup_{T \rightarrow \infty} \frac{\mathbb{E}[L(W(0))]}{T\varepsilon} + \frac{U}{\varepsilon}. \end{aligned}$$

From the definition of the capacity region in the previous section, we have

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_{k=1}^K \sum_{i=1}^{N_k(t)} \mathbb{E}[B_{ki}(t)/R_k^{\max}] = 1 - \varepsilon.$$

Note that

$$\lim_{t \rightarrow \infty} \mathbb{E}[r_{ki}(t)/R_k^{\max}] = 1,$$

and also  $\mathbb{E}[L(W(0))]$  is bounded, we have

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[W(t)] < \infty,$$

which indicates that the total queue length in the system is bounded and hence the system is stable.  $\square$

The intuitive explanation of the above theorem is as follows. If the scheduling algorithm always tries to schedule a flow when it has its possible maximum transmission rate, the system is stable thanks to the efficient utilization of resource. From the definition of the capacity region (in Sec. 2.2) we can tell that if a flow is scheduled when it is not in its maximum transmission channel rate, it probably needs more time slots for transmission and hence leads to waste of resource. However, the above is not a necessary condition for a system to be stable. For example, if there is a large gap between the arrival rate vector and the capacity region, i.e., the traffic intensity of the system is quite low, it is possible that the system is able to deliver all the arrival bits though some transmission is associated with a low transmission rate. But for a network with a very high traffic intensity, i.e., there is a very small gap between the arrival vector and the system capacity, the condition in (2.3) is almost a necessary

condition.

**Lemma 1.** *Given infinite buffer, for a single-class ( $K = 1$ ) flow-level dynamic multi-user wireless system with the HAD scheduling algorithm as in (2.2), if the system is unstable, we have*

$$\lim_{t \rightarrow \infty} \mathbb{P}\{r(t) < R_1^{\max}\} = 0.$$

*Proof.* In a homogeneous system, without loss of generality, we suppose  $Q_1(t)$  has the maximum HOL access delay in the system at time  $t$ , i.e.,  $H_1(t) = H^{\max}(t)$ . For the simplicity of presentation, suppose that  $r_i(t)$  are copies of a positive integer random variable  $R \in \{R_1, R_2\}$  and  $R_1 < R_2$ . One can extend the proof to the multi-rate case with the same approach.

Let  $\mathcal{U}(t)$  denote the set of flows in which all the flows have the HOL access delay larger than  $(H_1(t) \cdot R_1/R_2)$  at time  $t$ , and let  $\tilde{N}(t)$  denote the number of flows in  $\mathcal{U}(t)$ . From (2.2) we know that the probability  $\mathbb{P}\{r(t) < R^{\max}\}$  is equivalent to the probability of the event that  $\forall Q_i(t) \in \mathcal{U}(t)$  are associated with  $R_1$  at time  $t$ . Denoting  $p = \mathbb{P}\{r_i(t) = R_1\}$ , we have

$$\mathbb{P}\{r(t) < R^{\max}\} = p^{\tilde{N}(t)}. \quad (2.5)$$

Next we prove that  $\lim_{t \rightarrow \infty} \tilde{N}(t) \rightarrow \infty$ .

Suppose that we can find a positive integer  $\mathcal{M}$  such that  $\tilde{N}(t) < \mathcal{M}$  for all the time  $t$ . If  $r_1(t) = R_2$ ,  $Q_1(t)$  will be scheduled, so the probability for  $\{Q_i(t) : i = \arg \max_{1 \leq i \leq N(t)} H_i(t)\}$  to be scheduled is larger than  $1 - p$  for any time slot. Since  $\mathcal{M}$  and  $p$  are not time coupled, we can find a positive integer  $\mathcal{T}$  which is also irrelevant with time, and for any arbitrarily small positive  $\varepsilon$ , the probability that  $\forall Q_i \in \mathcal{U}(t)$  can be scheduled within  $\mathcal{T}$  slots is larger than  $1 - \varepsilon$ . In other words, the maximum HOL access delay at time  $t + \mathcal{T}$ , denoted by  $H_1(t + \mathcal{T})$ , is smaller than  $(H_1(t)R_1/R_2) + \mathcal{T}$  with a probability larger than  $1 - \varepsilon$ . Since we can find a positive value  $\mathcal{H}$  such that we have  $\mathbb{P}\{\frac{H_1(t)R_1}{R_2} + \mathcal{T} < H_1(t)\} > 1 - \varepsilon$  when  $H_1(t) > \mathcal{H}$ . So we have  $\mathbb{P}\{H_1(t + \mathcal{T}) < H_1(t)\} > (1 - \varepsilon)^2$ . This indicates that when the maximum HOL access delay is larger than  $\mathcal{H}$ , and after  $\mathcal{T}$  slots, the maximum HOL access delay is not likely to increase beyond and bounded by  $\mathcal{H}$ . This conclusion contradicts with the system instability that we discuss here. So the assumption that we can find  $\mathcal{M}$  such that  $\tilde{N}(t) < \mathcal{M}$  for all the time  $t$  is not true, which leads to  $\lim_{t \rightarrow \infty} \tilde{N}(t) \rightarrow \infty$ .

Consequently from (2.5) we have  $\lim_{t \rightarrow \infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0$ .  $\square$

**Theorem 3.** *Given infinite buffer, for a flow-level dynamic multiuser wireless system with the HAD scheduling algorithm as in (2.2), if the arrival rates lie in the capacity region, the system is stable when  $K \geq 1$ .*

*Proof.* First we consider the case that  $K = 1$ . Based on Lemma 1, we have  $\lim_{t \rightarrow \infty} \mathbb{P}\{r(t) < R_k^{\max}\} = 0$ . Since the arrival rates lie in the capacity region, based on Theorem 2, we have the conclusion that the system is stable.

Now consider a heterogeneous system where  $K > 1$ . Suppose this system is unstable with the HAD scheduling. Without loss of generality, we assume  $R_{k1} < R_{k2} < \dots < R_{k(m_k-1)} < R_{km_k}$  if the class- $k$  flows have  $m_k$  rates.

Suppose that class 1 is the unstable class, i.e.,  $N_1(t) \rightarrow \infty$  when  $t \rightarrow \infty$ . Similar to the proof of Theorem 1, we can have the conclusion that if  $\exists i \in \{1, 2, \dots, K\}$ , such that  $N_i(t) \rightarrow \infty$ , then  $\forall k \in \{1, 2, \dots, K\}$ , we have  $N_k(t) \rightarrow \infty$ , i.e., if one class is unstable, then all the classes are unstable. If a flow in class-1 is scheduled, from the proof of Lemma 1, we can directly come to  $\lim_{t \rightarrow \infty} \mathbb{P}\{r_{1i}(t) < R_{1m_k}\} = 0$  if  $r(t) = r_{1i}(t)$ . This conclusion can be extended to a more general one that for any class- $k$  in the system, if it is unstable,  $\mathbb{P}\{r_{ki}(t) < R_{1m_k}\} = 0$  if  $r(t) = r_{ki}(t)$ . However, from Theorem 2, the system is stable. This is a contradiction to the assumption of instability, and hence this assumption is not true. Combined with the case where  $K = 1$ , we have Theorem 3 proved.  $\square$

With the above analysis, we can draw the conclusion that HAD can stabilize the systems when the number of flows is not fixed as long as the arrival rate lies in the system capacity region, so it is throughput-optimal for the systems with flow-level dynamics. Note that essentially the traffic arrival characteristics do not influence the algorithm's throughput-optimality through the analysis above. In the system model of our work, however, because the definition of the system capacity region is related to both of the traffic and the channel profile, we put the flows that have the same traffic arrival characteristic and the same channel profile into one class just for the convenience of the presentation.

HAD scheduling has several advantages compared with the existing scheduling algorithms. First, in the MR scheduling, either the pre-requisite of channel condition distribution is required, or a learning period is necessary to learn the possible maximum channel rate which has an unknown influence on the system performance; while HAD scheduling is an online scheduling and its decision-making process is simple. It

is more practical in situations where the channel distribution may not be available in advance. Second, in the F-D-MW, the necessary condition in the proof of stability [46] is that all the flows have i.i.d. channel condition, or at least the maximum channel rates among all the flows are identical, which narrows the utilization; while in HAD, heterogeneous channel condition distributions of different classes are supportable. Even when the  $R_K^{\max}$  is different among the  $K$  classes, HAD is still able to stabilize the system according to Theorem 3. Also in F-D-MW, the flows that come into the system earlier will always have a higher priority to win the chance for transmission than the flows that enter the system later, so that the new flows may suffer long start-up latency. While in HAD scheduling, the new flows can have more opportunities to be served. Last but not least, in the F-D-MW scheduling algorithm, each flow has to record the delay for every packet, while HAD only needs a simple counter for the HOL access delay, thus the overhead is reduced.

The implementation of our proposed HAD scheduler is similar to that of the classic QMW scheduler, and HAD does not bring more signalling overhead than the widely adopted PF and the other online throughput-optimal scheduling algorithms such as QMW and F-D-MW. In the existing works, the MaxWeight type of scheduler has been implemented and tested, e.g., by the work from the Bell Laboratories, Alcatel-Lucent in 2011 [26]. Moreover, according to [31], different types of the MaxWeight scheduling components have already been adopted and implemented in practice, e.g., data center bridging by Cisco [14] and Qualcomm's Flashling peer-to-peer wireless networks [43], etc.

## 2.4 Performance Evaluation

In this section, we evaluate the performance of HAD scheduling along with the other scheduling algorithms, including the Queue-length based MaxWeight (QMW) [50], the Flow-Delay based MaxWeight (F-D-MW) [46], the Max-Rate (MR) scheduling algorithms [33], and the Proportional Fairness algorithm (PF) [2].

In the simulation, we have two classes of short-lived flows. The traffic burst size of the class-1 and class-2 flows is 30 (units) and 60 (units), respectively. We adopted Good-Bad channel model, i.e., each class has two transmission rates. The channel rate for class-1 flows is  $R_1 = \{9, 10\}$  (units/slot), and  $\mathbb{P}\{R_1 = 9\} = 0.1$ ,  $\mathbb{P}\{R_1 = 10\} = 0.9$ ; while  $R_2 = \{16, 20\}$  (units/slot), with  $\mathbb{P}\{R_2 = 16\} = 0.2$ ,  $\mathbb{P}\{R_2 = 20\} = 0.8$ . The arrival probability is calculated according to the traffic intensity  $\rho$ . The history rate

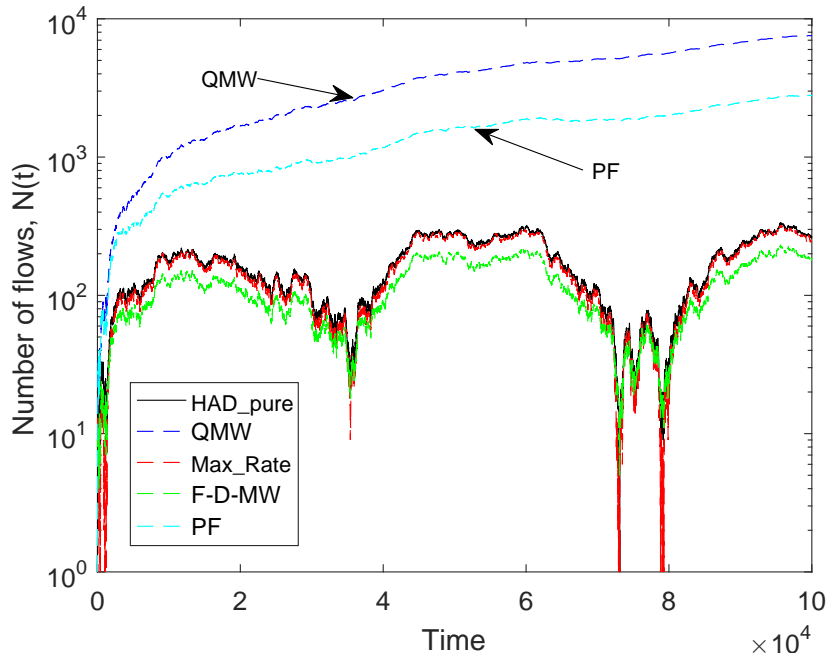


Figure 2.2: Number of flows  $N(t)$  with  $\rho = 0.999$

window size for PF is 1000 time slots [2]. The simulation tool is Matlab. With traffic intensity  $\rho = 0.999$ , the throughput-optimality of HAD scheduling is compared with the other algorithms, illustrated in Figs. 2.2-2.4.

The results shown in Fig. 2.2 are the evolution of the number of flows  $N(t)$  in the system with y-axis in the logarithmic form. We can observe that  $N(t)$  of QMW and PF increases with time and from the increasing trend we can tell that the system cannot be stabilized with either the QMW or the PF scheduling. While with the other three scheduling algorithms,  $N(t)$  is bounded and the system can be stabilized. We can observe that  $N(t)$  with HAD is slightly larger than that with MR, and F-D-MW has the smallest  $N(t)$  which is the result of using flow delay as the scheduling weight so that the old flows in the system have more chances to transmit. By allowing more flows to co-exist in the system, HAD scheduling may allocate more resources to the newer flows and hence can achieve a lower start-up latency and a better fairness between the old and new flows.

Fig. 2.3 shows the evolution of the system backlog  $|Q(t)|$  with y-axis in the logarithmic form. Similar to Fig. 2.2, we can observe that  $|Q(t)|$  with the QMW and PF scheduling algorithm keeps increasing with time, and  $|Q(t)|$  with the other three algorithms are bounded, and are almost identical. The same conclusion can be drawn

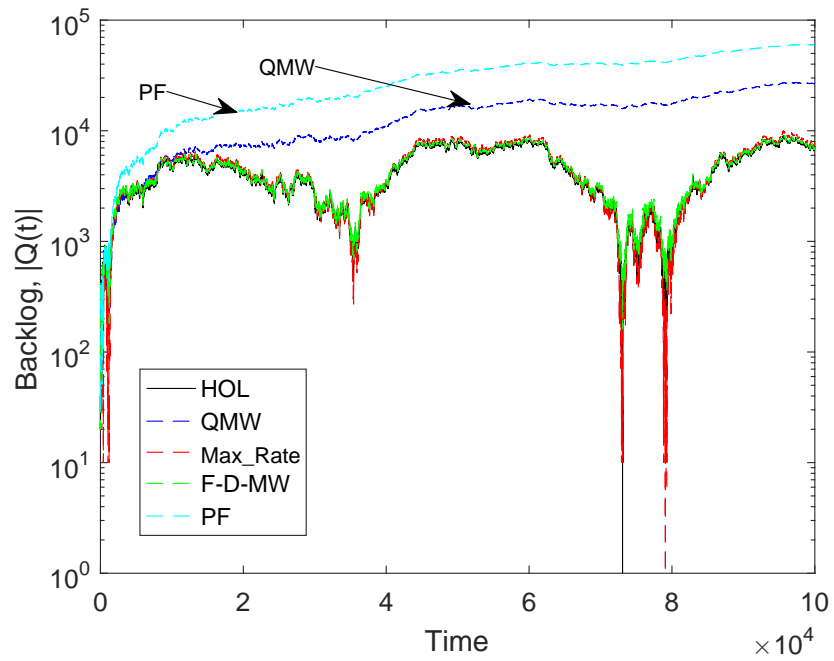


Figure 2.3: System backlog  $|Q(t)|$  with  $\rho = 0.999$

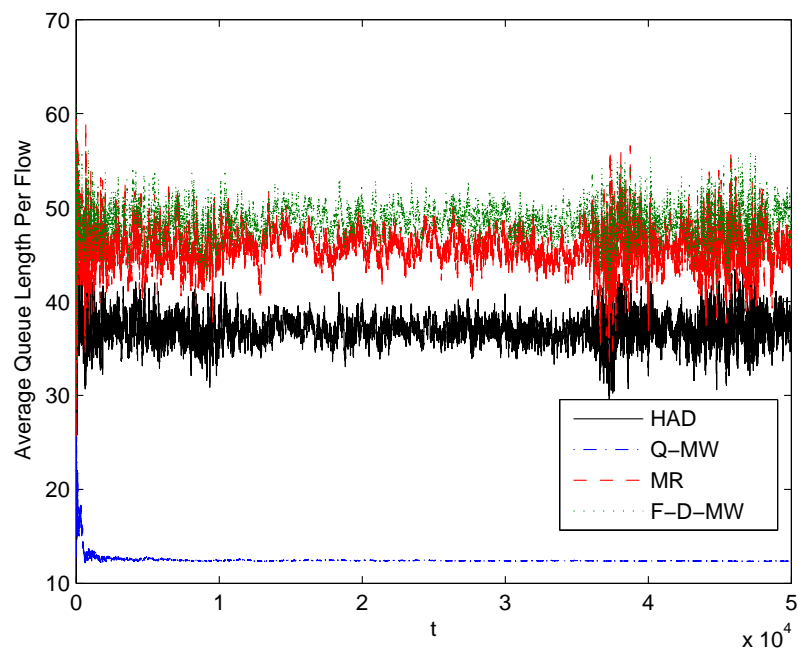


Figure 2.4: Average queue length with  $\rho = 0.999$

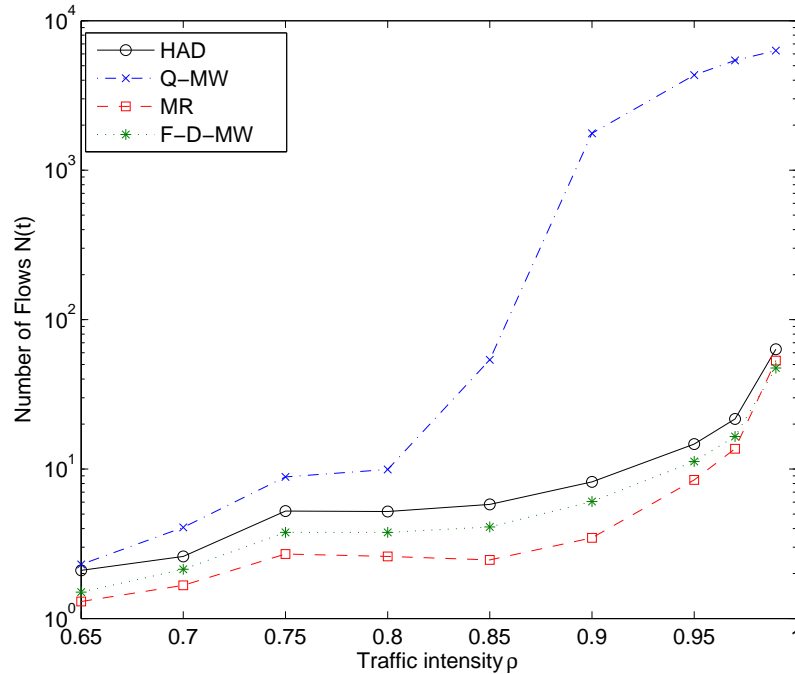


Figure 2.5: Number of flows  $N(t)$  with different  $\rho$  at  $t = 50,000$

that the system can be stabilized by all the scheduling algorithms except QMW and PF.

The evolution of the average queue length per flow, which is defined as  $\bar{Q}(t) = |Q(t)|/N(t)$ , is illustrated in Fig. 2.4. The QMW scheduling has the smallest  $\bar{Q}(t)$ , while the other three algorithms all have a larger  $\bar{Q}(t)$ . The reason is that, the QMW scheduling computes the weight of each flow proportional to the individual queue length. Thus once a flow has only a small amount of data left for transmission, it will have a small weight during the scheduling process, which results in two consequences: 1) there will be an increasing number of flows accumulated in the system as shown in Fig. 2.2; 2) these large number of flows with the small number of tail bits will hardly get a chance for transmission, which makes the average queue length maintained at a low level. While for the other three throughput-optimal scheduling algorithms,  $N(t)$  is kept to be very low, and the average queue length (mainly associated with the new arriving flows) is relatively high. It can also be observed that the  $\bar{Q}(t)$  of F-D-MW is the largest, and the  $\bar{Q}(t)$  of HAD is the smallest. As the system backlog is almost identical for all these three algorithms,  $\bar{Q}(t)$  has an inverse relationship with  $N(t)$ .

In Figs. 2.5 and 2.6, we compare the performance of HAD scheduling and the

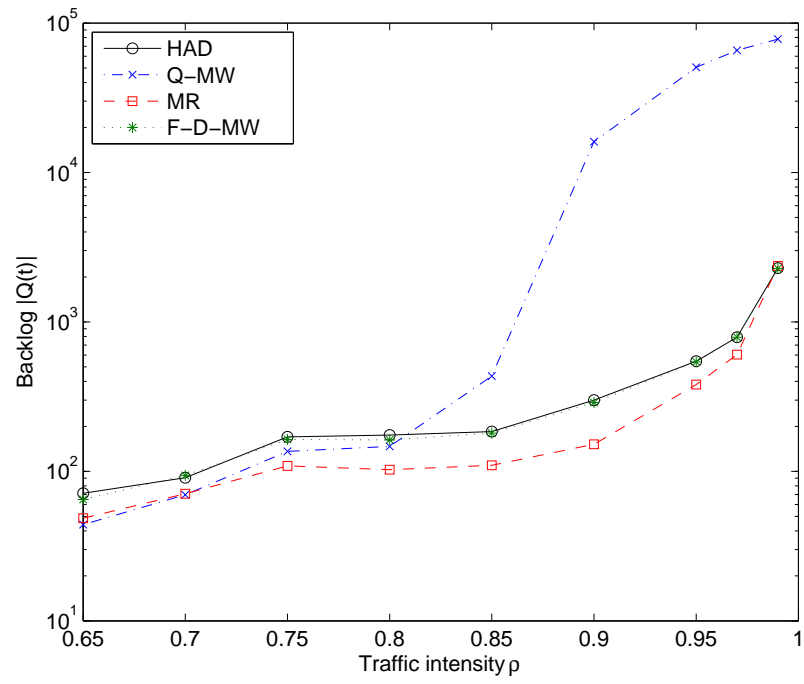


Figure 2.6: Backlog  $|Q|$  with different  $\rho$  at  $t = 50,000$

other three scheduling algorithms with the traffic intensity varying from 0.65 to 0.995. The  $x$ -axis is the traffic intensity  $\rho$  which is defined in the system model. Figs. 2.2 and 2.3 show the evolution of the system  $N(t)$  and  $|Q(t)|$  when  $\rho = 0.999$ , while here we take the snapshots of the evolution of  $N(t)$  and  $|Q(t)|$  with  $\rho$  increasing from 0.65 to 0.995 at the moment of  $t = 50,000$ . The results are averaged over 10 simulations. When  $\rho \leq 0.8$ , all of the four algorithms are stable because of the low traffic intensity. Because the weight is proportional to the queue length, the QMW has a good performance on system backlog  $|Q(t)|$ , although it has more flows in the system as we can observe in Fig. 2.5. It is noticeable in Fig. 2.5, when  $\rho \geq 0.8$ , the number of flows of the QMW scheduling algorithm in the system grows fast, and the same trend on the system backlog  $|Q(t)|$  can be also observed in Fig. 2.6. While the  $N(t)$  and  $|Q(t)|$  of HAD and the other two algorithms only experience a slow increase. When  $\rho$  is small, the MR scheduling has the best performance thanks to the full knowledge of the system channel information, while the performance of the three throughput-optimal algorithms tend to converge when  $\rho$  approaches one.

## 2.5 Conclusion

In this chapter, we have studied the HAD scheduling algorithm in a multiuser wireless network with flow-level dynamics. The sufficient condition for the flow-level stability of a network with short-lived flows has been provided. Based on this result, we have proved the throughput-optimality of the HAD scheduling algorithm in heterogeneous wireless networks. The HAD is an online scheduling algorithm with no requirement of prior knowledge of the statistics of the arrival traffic and channel state information, and hence it is more practical and simpler to implement compared to the other existing throughput-optimal scheduling algorithms considering flow-level dynamics. The performance of HAD has been evaluated through extensive simulations, which demonstrated that it outperforms the QMW scheduling, and has a similar performance to the other throughput-optimal scheduling algorithms, such as MR and F-D-MW.

In this chapter, we have considered the scheduling design for flow-level dynamic systems. In the Internet, the dominant transport layer protocol, TCP, has its end-to-end control function, which may interact with link-layer resource management solutions. Thus it is important to study whether the proposed throughput-optimal scheduling algorithm is compatible with TCP, which motivates the work in the following chapter.

## Chapter 3

# On Achieving Fair and Throughput-optimal Scheduling for TCP Flows in Wireless Networks

### 3.1 Introduction

TCP has been extensively studied in the literature [4]. In the past three decades, the scale of the Internet has grown by several orders, and TCP still performs decently well. The basic congestion and flow control mechanism of TCP was to probe for the available bandwidth while maintaining certain level of fairness among coexisting flows. In practice, all TCP variants, both the widely used loss-based variants and the other delay-based ones, have their own clock timing, which relies on the end-to-end acknowledgement packets (ACKs). Based on the received ACKs, a TCP sender determines whether and how many packets should be injected into the network by updating the size of the congestion window (*cwnd*). This protocol was originally designed for the wired networks. As an increasing number of wireless devices are involved in the Internet, it becomes increasingly important to investigate the compatibility between TCP and the lower layer wireless scheduling algorithms [6, 56]. The adjustment of *cwnd* was designed to achieve fairness among all TCP flows. However, the control mechanism of some existing throughput-optimal scheduling algorithms conflicts with TCP congestion control. How TCP can be compatible with throughput-optimal scheduling algorithms in wireless networks when multiple users share a radio link is the research interest of this chapter.

## 3.2 TCP Compatibility Problem

In this section the existing throughput-optimal scheduling algorithms are introduced in detail. Since QMW is the origination of these algorithms, we use QMW as an example to investigate the unfairness problem with TCP flows. We also reveal the unfairness problem of the delay-based scheduling algorithm with TCP flows. With the observation of unfairness in the example, we further discuss the motivation and the approach to find the throughput-optimal scheduling algorithm for fair TCP flow scheduling.

### 3.2.1 System Capacity and Throughput-optimal Scheduling Algorithms

The wireless network capacity region  $\Lambda$  is defined as the closure of all arrival rate vectors that can be stably transmitted in the network, considering all possible scheduling policies. An arrival rate vector can be stably transmitted when the queueing stability is assured. The queueing stability of a discrete time process  $Q(t)$  is defined as that  $Q(t)$  is strongly stable if it satisfies  $\limsup_{t \rightarrow \infty} (1/t) \sum_{\tau=0}^{t-1} \mathbb{E}[|Q(\tau)|] < \infty$  [37].  $\Lambda$  is fixed and only depends on the channel statistics of the system. A scheduling algorithm is throughput-optimal if it is able to ensure the queueing stability as long as the vector of average arrival rates is within the capacity region [3].

QMW is provable to be throughput-optimal with the condition that the number of users in the system does not change over time [50]. Assuming that only one user can be scheduled in every time slot, the scheduling rule of QMW can be found in Algorithm 2, in which the scheduler tries to maximize the selected transmission rate, weighted by the queue length.

**Algorithm 2.** Let  $Q_i(t)$  denote the  $i$ -th flow at time  $t$ , and the corresponding queue length is  $|Q_i(t)|$ . QMW seeks user  $i$  to transmit which satisfies the following condition at the beginning of time slot  $t$ :

$$i^*(|Q_i(t)|, r_i(t)) \in \arg \max_{1 \leq i \leq N(t)} |Q_i(t)| \cdot r_i(t), \quad (3.1)$$

with uniform tie-breaking if there are more than one users satisfying the condition.

In (3.1),  $r_i(t)$  is the transmission rate of  $Q_i(t)$  at time  $t$ , and  $N(t)$  is the total number of users in the system at time  $t$ . The scheduling decision is made in every

time slot independently. Due to its desirable throughput-optimality feature and low complexity to implement, its performance has been extensively studied [3, 37, 57]. Other queue-length based scheduling algorithms including the Exponential rule and Log rule were proposed in [44, 48] to improve the delay performance. The applications of throughput-optimal scheduling algorithms can be found in [7, 8, 59, 63].

In the networks with a dynamic number of flows over time referred as flow-level dynamics, QMW is no longer applicable due to the instability problem [52]. The capacity region for systems with flow-level dynamics is different from that without flow-level dynamic, which will be given in Sec. 2.2. Several scheduling solutions were proposed for systems with flow-level dynamics. The Max-Rate scheduling algorithm (MR) was designed in [33], but the pure MR scheduling is an off-line algorithm, and requires the full knowledge of the channel distribution in advance, which is difficult and sometimes impossible to obtain in practical systems. A modified MR in the same paper uses the history information to learn the channel variance, but how to design the learning window is an open question. The Flow-Delay based MaxWeight (F-D-MW) scheduling algorithm was studied in [46] to stabilize the systems with flow-level dynamics. The proof shows that F-D-MW is throughput-optimal, but the drawbacks are the complexity of implementation and the undesirable delay performance.

The above scheduling algorithms mainly focus on how to achieve throughput-optimality, and have no special consideration of how to schedule TCP controlled flows. In the next subsection we will use an example to show the incompatibility between TCP and the queue-length based scheduling.

### 3.2.2 An Example

Fig. 3.1 illustrates the interaction of QMW and TCP. We assume that the packet arrivals are regulated by a loss-based TCP congestion controller (TCP-Reno [42] or TCP-SACK [34]). For the simplicity of the explanation, we assume that only one packet will be transmitted when a flow is scheduled, and that all the packets have the same size of one maximum segment size (MSS). The queueing time and transmission delay in the wireless access links dominate the variation of the Round Trip Time (RTT).

Suppose that before time  $t$ , the second flow  $Q_2(t)$  has been in the system for a while and its TCP congestion window size at time  $t$  has already been increased to be larger than one MSS, i.e.,  $cwnd_2(t) > 1$  MSS; while the first flow  $Q_1(t)$  is

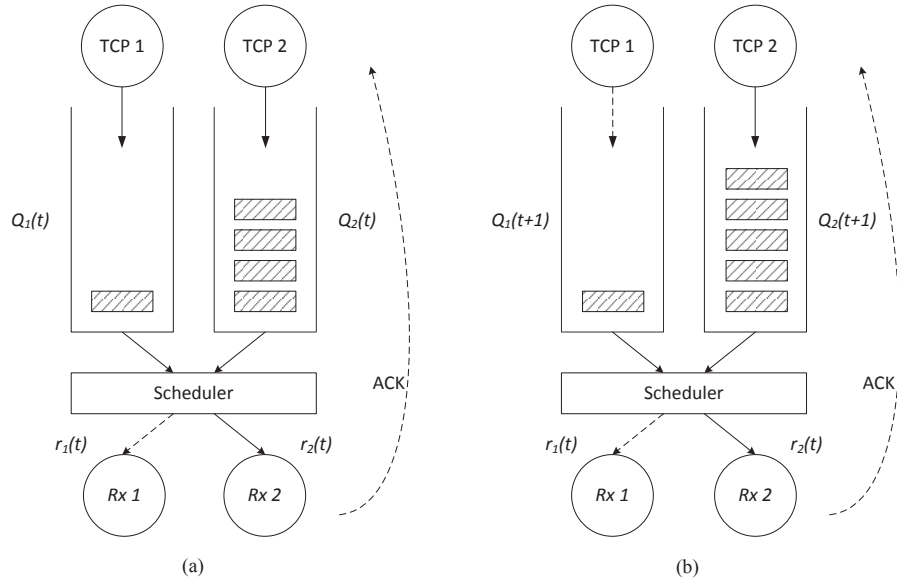


Figure 3.1: Incompatibility between TCP and QMW scheduling.

a new one entering the system, and its TCP congestion window size is small, e.g.,  $cwnd_1(t) = 1$  MSS. Fig. 3.1(a) shows the example that the queue lengths of these two flows are  $|Q_1(t)| = 1$  MSS and  $|Q_2(t)| = 4$  MSS at time  $t$ . Assume  $r_1(t) = r_2(t)$ . Since  $|Q_2(t)| > |Q_1(t)|$ , according to the scheduling policy of QMW in Algorithm 2, a packet from  $Q_2(t)$  is transmitted. Rx 2, the receiver of  $Q_2(t)$ , generates an ACK after receiving the packet, and sends the ACK to the TCP sender, i.e., TCP 2 in Fig. 3.1. After receiving the ACK, TCP 2 slides and increases the congestion window size at time slot  $t + 1$ , i.e.,  $cwnd_2(t + 1) > cwnd_2(t)$ , and sends one or more packet(s) into  $Q_2(t)$ . On the other hand, since no packet is transmitted in flow 1 at time slot  $t$ , TCP 1 receives no ACK, and thus its congestion window size remains one. As a result, no new packet is added to  $Q_1(t)$  and we still have  $|Q_2(t + 1)| > |Q_1(t + 1)|$  in time slot  $t + 1$  as shown in Fig. 3.1(b). Eventually,  $|Q_1(t)|$  will hardly increase and the first flow suffers from the starvation, while the second flow dominates the usage of the resources.

QMW also makes the old flows suffer from a long delay before their last few packets are transmitted. This problem is considered as the last-packet problem of QMW, which is also the reason why QMW is not applicable with flow-level dynamics [24]. Consider that flow one has a finite amount of data to transmit. Before finishing the whole transmission, it is possible that only one or a couple of packets are left in its

queue. If another flow has many packets waiting in the queue at this time, the last few packets in flow one have to wait without being scheduled until the number of packets in the other flow's queue decreases to a sufficiently small value.

Simulation results in Sec. 3.5 will show the severe unfairness problem of the joint behavior of TCP and QMW both at the beginning and the end of each flow's transmission. Other variants of QMW, such as the Exponential rule and the Log rule [44, 48], all directly or indirectly use the queue length as the weight for the scheduling decision, and thus they can be categorized as queue-length based scheduling. They encounter the same unfairness and starvation problem when working with TCP flows. In the rest of this chapter, we only take QMW as the representative one in this category.

### 3.2.3 F-D-MW

The scheduling rule of F-D-MW can be found in Algorithm 3.

**Algorithm 3.** *Let  $Q_i(t)$  denote the  $i$ -th flow at time  $t$ .  $D_i(t)$  is the sojourn time of  $Q_i(t)$  which is measured from the time instant when  $Q_i(t)$  arrives in the network waiting for being scheduled. F-D-MW seeks user  $i$  to transmit which satisfies the following condition at the beginning of time slot  $t$ :*

$$i^*(D_i(t), r_i(t)) \in \arg \max_{1 \leq i \leq N(t)} D_i(t) \cdot r_i(t),$$

*with uniform tie-breaking if there are more than one users satisfying the condition.*

F-D-MW is throughput-optimal with a dynamic number of users in the system [3, 28]. However, F-D-MW is not compatible with TCP flows in wireless networks either. Because an F-D-MW scheduler always assigns a higher weight to the existing TCP flows in the network, the new flows have a much lower instantaneous throughput when they enter the system, and thus they suffer the long start-up latency and may even be starved at the beginning. This may not be desirable for the applications with stringent delay requirements. In most operating systems (Windows, MacOS, etc.), when a node begins to establish a TCP connection, it sends out the TCP SYN control packet and will wait up to 75 seconds (20 seconds in Unix) for the SYN-ACK packet from the destination node. When the timer expires, the attempt to establish the TCP connection will be abandoned. With F-D-MW (and QMW), it is possible that the connection may be abandoned due to the long start-up latency.

### 3.2.4 Further Discussion

The objective of an optimal scheduler is to allocate resources to stabilize the system whenever possible. Typically, a resource allocation problem can be modelled as a utility maximization problem, and solved by the approaches as those in [17, 30, 62]. By using dual-decomposition, the problem can be decomposed into a rate control problem and a Maximum Weighted Matching problem (scheduling problem). In such an approach, the rate control is explicitly performed for the scheduling algorithm, and the congestion signal of the rate control is the queue length, which is a required feedback information to the sender. Since the throughput-optimal scheduling (including QMW) can be explained as a generalization of the scheduling algorithm developed by this approach, a rate control may be needed to cooperate in real operation to prevent undesirable performance degradation.

In the Internet, however, the queue-length based rate control is not likely to be widely used, so long as TCP is the dominant transport layer protocol [47]. The popular TCP variants, such as TCP Reno, TCP New Reno [19] and TCP SACK, are all window-based congestion control using the packet loss as the congestion signal. Since it is not likely to drastically modify TCP to be compatible with the scheduling algorithms due to the backward compatibility concern, how to design an efficient scheduling algorithm in the link layer to be compatible with the existing TCP protocol is a critical issue.

We have two types of methods to design scheduling algorithms to be compatible with the current window-based TCP. The first is to use a utility based non-throughput-optimal scheduling algorithm in the MAC layer, whose stability region is less than the system capacity region, which results in less efficient channel utilization. A typical example of such algorithms is the Proportional Fairness scheduling algorithm (PF), which has been widely adopted in the cellular systems such as the LTE networks. PF is able to fairly allocate the channel resources for all the users in the system according to their previous resource allocation while considering the multi-user diversity gain. But it has been shown that PF is not throughput-optimal [2]. [58] has proven that utility based scheduling, including the PF scheduling, is not throughput-optimal, and thus in general the stability region is less than the capacity region.

The second is to develop new throughput-optimal scheduling algorithms compatible with TCP. There are two main approaches to design the throughput-optimal scheduling algorithms, i.e., the queue-length-based and the delay-based approaches.

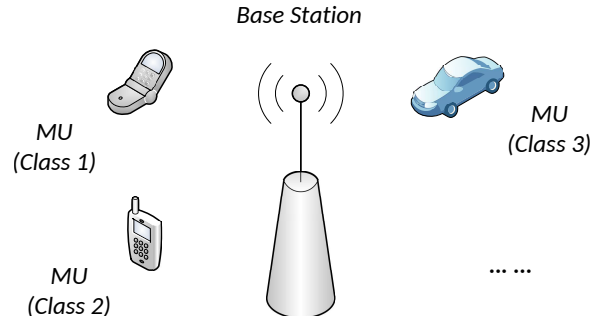


Figure 3.2: The downlink of a wireless network with different classes of flows.

As QMW is not desirable for supporting TCP flows, a newly designed queue-length based scheduling was proposed in [47], which uses network coding and the computation of a threshold when deciding the weight of each user. The algorithm shows throughput-optimality and fairness with a fixed number of long-lived TCP flows, but the design of the threshold and network coding brings the complexity in implementation. Furthermore, this algorithm is not designed for the networks with flow-level dynamics. Therefore we focus on the second type, the delay-based approach. The existing delay-based solution, F-D-MW, however, is not compatible with TCP either as explained earlier. This motivates us to study our own delay-based scheduling algorithm.

### 3.3 System Model

In this chapter, we consider the downlink of a single-hop centralized wireless access network as illustrated in Fig. 3.2, which works in slotted time. Our work can also be applied to uplink scheduling if a centralized scheduler exists, which is omitted due to space limit. The network consists of one central controller, such as the base station (BS), and  $N(t)$  mobile users (MU) at time  $t$ . Each MU is associated with a distinct TCP flow. As each user is associated with one flow, we do not distinguish the concept of “user” and “flow” thereafter. Flows are categorized into different classes according to their channel profiles, and the central controller in the network selects one flow to transmit for each time slot.

### 3.3.1 Networks with Flow-level Dynamics

In the network with flow-level dynamics [52], we have the arrival of new flows and the departure of old ones. We consider that there are  $K$  classes of flows in the system, and each class of flows is defined according to the channel profiles. At time slot  $t$ , the  $i$ -th flow of class  $k$  is denoted by  $Q_{ki}(t)$ , the number of flows of class- $k$  is  $N_k(t)$ , and the total number of flows in the system is  $N(t) = \sum_{k=1}^K N_k(t)$ . For every  $Q_{ki}(t)$ , there is a finite amount of data to transmit. After all the data are delivered through the radio link, the corresponding flow will leave the system. As a result, the number of flows in the current time slot may not be the same as that in the next time slot. Thus the number of flows in the system is time-varying.

### 3.3.2 Channel Model

Let  $r_{ki}(t)$  denote the transmission rate of the wireless channel at time  $t$  for  $Q_{ki}(t)$ . The unit of the channel rate is *bit/slot*. The BS can transmit at most  $r_{ki}(t)$  bits at time slot  $t$  for  $Q_{ki}(t)$ .  $r_{ki}(t)$  may vary over time as a result of channel fading. For each class  $k$ , we assume that  $r_{ki}(\cdot)$  are i.i.d. copies of positive random variable  $R_k$  with finite first and second order moments, and  $r_{ki}(t) \in \{R_{k1}, R_{k2}, \dots, R_{km_k}\}$ . Different classes have different channel profiles, which give the channel rate distributions, i.e., what is the probability that the channel rate is equal to a certain value. The maximum possible transmission rate of the class- $k$  flows is defined as  $R_k^{\max} := \sup\{r : \mathbb{P}\{R_k = r\} > 0\}$ , and the maximum possible transmission rate of the system is defined as  $R^{\max} := \max_{1 \leq k \leq K} \{R_k^{\max}\}$ . The flows in the same class have the same channel rate distribution, and thus they have the same upper bound, lower bound, the average rate and the channel rate variance. It is possible that the flows in the same class have different instantaneous channel rates as the channel rate is a random variable.

### 3.3.3 Queueing Model

We assume that new flows can arrive at the scheduler at any time in a time slot. The number of new class- $k$  flows arriving during time slot  $t$  is  $A_k(t)$ , which is the i.i.d. copy of a random variable  $A_k$  with a finite mean  $\lambda_k = \mathbb{E}[A_k(\cdot)]$ , where  $\mathbb{E}[\cdot]$  denotes expectation. The packets of the  $i$ -th flow in class  $k$  are stored in a dedicated buffer. We consider that the amount of data stored at the sender side is  $B_{ki}(t)$ , and the buffer is large enough to avoid buffer overflow.  $B_{ki}(t)$  is the i.i.d. copy of an

integer random variable  $B_k$  and has a finite mean  $\beta_k = \mathbb{E}[B_{ki}(\cdot)]$ . We assume that the second moments of  $A_k$  and  $B_k$  are both finite. TCP is used as the end-to-end transport protocol. For each flow, TCP determines the amount of data delivered from  $B_{ki}(t)$  to the transmission queue  $Q_{ki}(t)$  of the scheduler. The amount of data delivered by TCP from  $B_{ki}(t)$  to  $Q_{ki}(t)$  is denoted by  $s_{ki}(t)$ . We suppose that the scheduling decision is made at the beginning of every time slot, so that any of the data packets that arrives after the beginning of slot  $t$ , i.e., any  $B_{ki}(t)$  of  $\forall k = \{1, 2, \dots, K\}$  and  $\forall i = \{1, 2, \dots, N_k(t)\}$  can only be transmitted in the following slots. We define  $|Q_k(t)| := \sum_{i=1}^{N_k(t)} |Q_{ki}(t)|$  as the class- $k$  backlog and  $|Q(t)| := \sum_{k=1}^K |Q_k(t)|$  as the system backlog. The queue dynamic is given by

$$|Q_{ki}(t+1)| = \max[|Q_{ki}(t)| - r_{ki}(t) + s_{ki}(t), 0]. \quad (3.2)$$

If there is no more traffic arrival for  $Q_{ki}(t)$  from the current TCP session,  $Q_{ki}(t)$  will leave the system. With the above model, we can define the capacity region of a flow-level dynamic network. Let  $\gamma_k$  represent the expected number of time slots required for the service of a class- $k$  flow if served with  $R_k^{\max}$ , and then we have  $\gamma_k = \mathbb{E}\left[\frac{B_k}{R_k^{\max}}\right]$ . Let  $\rho_k = \lambda_k \gamma_k$  denote the traffic intensity of class- $k$  flows, and  $\rho = \sum_{k=1}^K \rho_k$  denote the system traffic intensity. The system capacity region is defined as  $\mathcal{S} = \{(\lambda_1, \lambda_2, \dots, \lambda_K), (\gamma_1, \gamma_2, \dots, \gamma_K) : \rho < 1\}$ . For any arrival process that lies in the capacity region, if the system is strongly stable, i.e.,  $\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[|Q(t)|] < \infty$ , then the correspondingly adopted scheduling algorithm is throughput-optimal.

With the models above, intuitively, if the system is stable, the total amount of data in the system should be finite. If the system is unstable, the total amount of data will grow into infinity when  $t \rightarrow \infty$  considering infinite buffer size. Note that the traffic intensity represents on average how many slots are required to transmit the arrived data in one time slot if the maximum transmission rate is adopted. If the traffic intensity is smaller than 1, it means that the average amount of the data arrived in one time slot can be transmitted in less than one time slot by the maximum transmission rate, and thus there exists at least one scheduling algorithm to obtain the system stability. When the traffic intensity is larger than 1, it means that on average more than one time slot is required to transmit the amount of arrival data in one time slot, and thus the packets that cannot be transmitted in time will accumulate, which leads to system instability. From this perspective, the traffic intensity  $\rho < 1$  is defined as the system capacity region. Any arrival rate in the capacity region

can be stably transmitted by the throughput-optimal scheduling algorithms without admission control [52].

### 3.4 HOL Access Delay Based Scheduling

The definition of the Head-Of-Line (HOL) access delay has been given in Definition 1 in Chapter 2. HOL access delay can be viewed as the waiting time of  $Q_{ki}(t)$  being served from  $Q_{ki}(t)$ 's previous transmission, and is calculated according to the following equation:

$$H_{ki}(t+1) = (H_{ki}(t) + 1)(1 - \mathbf{1}_{ki}(t)), \quad (3.3)$$

where  $\mathbf{1}_{ki}(t)$  is the indicator function such that  $\mathbf{1}_{ki}(t) = 1$  only when  $Q_{ki}(t)$  is scheduled at time slot  $t$ , and  $\mathbf{1}_{ki}(t) = 0$  otherwise. With the system model and the definition of HOL access delay, we propose to adopt the HAD scheduling algorithm for the TCP flows. The definition of HAD scheduling algorithm can be found in Algorithm 1 in Chapter 2.

#### Remarks

Similar to QMW and F-D-MW, we also assume that we schedule one flow in each time slot in HAD. In the current LTE systems, resource blocks can be assigned to different flows in the same time slot, and this does not conflict with the main results of our work. Taking OFDMA as an example, although multiple flows can be scheduled in the same time slot, in any sub-channel (any logic resource block), we can only schedule one flow at a time. According to the structure of the latest 3GPP framework for LET system (shown in Fig. 6.4-1 of [1]), each UE has a dedicated buffer for data storage, and multiple queues exist in this data storage for different types of flows. All the flows are connected to the scheduler which selects the most desirable flow. After each scheduling decision, the queues are updated and ready for the next round of scheduling. Different flows can be categorized into different classes, and thus it is possible to implement the proposed HAD scheduling algorithm considering the system model in our work.

In the following we first investigate the throughput-optimality of HAD, and then study the fairness issue using HAD to schedule TCP flows. The fairness is closely related to HOL access delay. It can be known from Definition 1 in Chapter 2 that the

HOL access delay is actually the access waiting time, which indicates how long one flow has to wait between two consecutive transmissions. If two flows in the network need to equally share the channel time, their average access waiting time should be the same. Longer HOL access delay means that less channel time is allocated to the corresponding flow. Thus, we will study the fairness performance of HAD by investigating the HOL access delay. To measure the fairness of HAD, we use the fairness defined for weighted fair queueing in terms of allocated channel time, in which each flow can be allocated a share of channel time proportional to its weight. In HAD, the weight of the allocated channel time of each flow is its maximum channel rate, i.e., for  $Q_{ki}(t)$ , the weight  $w_{ki}$  is  $R_k^{\max}$ . With this definition, and considering the variance of channel condition and the uniform tie-breaking rule, all the flows should share the channel time with the average access waiting time correlated by  $w_{ki}$  in order to achieve a fair scheduling. According to [16], if

$$H_{ki}(t)/H_{mj}(t) = w_{mj}/w_{ki}, \quad (3.4)$$

the weighted average fairness of HAD in terms of the allocated channel time can achieve the maximum, considering that the allocated channel time can be viewed as the reciprocal of HOL access delay.

### 3.4.1 Throughput-optimality

The throughput-optimality of HAD in flow-level dynamic systems has been explained in Chapter 2. Here we only present the most important steps. The details and simulation results can be found in Chapter 2.

The proof of the throughput-optimality of HAD involves three steps. Let  $r(t)$  denote the real transmission rate of the network at time  $t$ . First, if a class- $k$  flow  $Q_{ki}(t)$  is scheduled, i.e.,  $r(t) = r_{ki}(t)$ , the sufficient condition for the network with flow-level dynamics to be stable for any arrival rate that lies in the capacity region is shown in (2.3). Second, we draw the conclusion that, for a single-class ( $K = 1$ ) flow-level dynamic multi-user wireless system with HAD, we can obtain  $\mathbb{P}\{r(t) < R^{\max}\} = p^{\tilde{N}(t)}$ , and thus (2.3) is true if the system is unstable. Here  $\tilde{N}(t)$  is an increasing function of  $N(t)$ , and  $p = \mathbb{P}\{r_i(t) \neq R_i^{\max}\}$ . Third, we obtained the result that HAD is throughput-optimal for a single-class ( $K = 1$ ) flow-level dynamic system by following the result in step two, and then we extended the throughput-optimality to a heterogeneous system with  $K > 1$  by explaining that if  $\exists k$  such that  $N_k(t) \rightarrow \infty$ ,

we have  $\forall i \in \{1, 2, \dots, K\}$ ,  $N_i(t) \rightarrow \infty$ , i.e., if one class is unstable, then all the classes in the network are unstable.

### Remarks

The throughput that we investigate here is the link layer throughput, and with the assumption that the only bottleneck link of the end-to-end path for the TCP flow is the wireless link with a dedicated buffer for each flow, the TCP end-to-end throughput can converge to the bottleneck link throughput [5]. However, TCP end-to-end throughput may be affected by multiple bottleneck links and the TCP flow may share the buffer with other flows, then the end-to-end TCP throughput may be different from the wireless link throughput we investigated here [55], which is out of the scope of this work.

The proof of (2.3) involves the definition of a Lyapunov function regarding the workload of the system at time  $t$ . The intuitive explanation of the corresponding theorem is as follows. If the scheduling algorithm always tries to schedule a flow when it has its possible maximum transmission rate, the system is stable thanks to the maximum utilization of resources. From the definition of the capacity region, we can tell that if a flow is scheduled when it is not in its maximum transmission channel rate, it probably needs more time slots for transmission and hence leads to a waste of resources. However, the above is not a necessary condition for a system to be stable. For example, if there is a large distance between the arrival rate vector and the capacity region boundary, i.e., the traffic intensity of the system is quite low, it is possible that the system is able to deliver all the arrival bits though some transmissions associated with a low transmission rate. But for a network with a very high traffic intensity, i.e., there is a very small gap between the arrival rate vector and the system capacity, the condition in (2.3) becomes necessary.

### 3.4.2 Fairness Analysis

**Proposition 1.** *Given i.i.d. channel transmission rate distribution for all the system flows, HAD can achieve fair resource sharing, so that all flows obtain an equal share of the channel time.*

*Proof.* In this proof, we only have one class of flows in the system, and thus the class index  $k$  in the subscript is omitted for simplicity. We first investigate the simplified scenario in which there are only 2 flows. The proof can be extended to the  $N$ -flow

cases. Since we have the assumption that all the system flows have i.i.d. channel transmission rate distribution, we can tell  $\mathbb{E}[r_1(t)] = \mathbb{E}[r_2(t)]$ . Next we will show that  $\mathbb{E}[H_1^{sch}(t)] = \mathbb{E}[H_2^{sch}(t)]$ , where  $H_i^{sch}(t)$  is the HOL access delay of flow  $i$  when it is scheduled at time  $t$ . To prove this, without loss of generality, we just need to show that  $\mathbb{E}[H_1^{sch}(t)] > \mathbb{E}[H_2^{sch}(t)]$  is impossible.

We assume that  $\mathbb{E}[H_1^{sch}(t)] > \mathbb{E}[H_2^{sch}(t)]$  is true, i.e., flow 1 has an on average larger head-of-line access delay than that of flow 2. The channel rate of  $Q_i(t)$  at time slot  $t$ , denoted by  $r_i(t)$ , only depends on the SINR of the wireless channel at  $t$  and the corresponding modulation and coding scheme. As a result, we have  $\Pr\{r_i(t) = x | H_i(t) = y\} = \Pr\{r_i(t) = x\}$ , which indicates that  $H_i(t)$  and  $r_i(t)$  are independent. Given that  $\mathbb{E}[r_1(t)] = \mathbb{E}[r_2(t)]$  and  $r_i(t)$  is independent of  $H_i(t)$ , according to HAD which seeks the maximum product of  $r_i(t) \cdot H_i(t)$  in every time slot, flow 1 has more chance to transmit than flow 2. Consequently, the average number of time slots that flow 1 has to wait between two of its transmissions is less than that of flow 2. This implies  $\mathbb{E}[H_1^{sch}(t)] < \mathbb{E}[H_2^{sch}(t)]$ , which contradicts to our assumption here. Thus the assumption that  $\mathbb{E}[H_1^{sch}(t)] > \mathbb{E}[H_2^{sch}(t)]$  cannot HAD. Similarly we can prove that  $\mathbb{E}[H_1^{sch}(t)] < \mathbb{E}[H_2^{sch}(t)]$  is also not possible. Thus we have  $\mathbb{E}[H_1^{sch}(t)] = \mathbb{E}[H_2^{sch}(t)]$ . This result can be extended to  $\mathbb{E}[H_i^{sch}(t)] = \mathbb{E}[H_j^{sch}(t)]$  if we have more than 2 flows in the system. Since the proof is similar to the 2-flow case, the details are omitted.  $\square$

Next we consider the fairness performance of HAD in heterogeneous networks. For a single flow  $Q_{ki}(t)$ , it is possible that  $H_{ki}^{sch}$  varies from time to time, even with the deterministic channel profile. Considering the variance of HOL access delay, we define  $\bar{H}_{ki}^{sch}$  as the average value of  $H_{ki}^{sch}(t)$  over time. To measure the fairness of HAD, we define  $\eta_{k,l}^H$  as the ratio of the average HOL access delay of class- $k$  and class- $l$  flows, i.e.,  $\eta_{k,l}^H = \bar{H}_{ki}^{sch} / \bar{H}_{lj}^{sch}$ . We assume that the choice of  $i$  in class  $k$  and  $j$  in class  $l$  does not affect the value of  $\eta_{k,l}^H$ . Similarly, we define  $\eta_{k,l}^R$  as the ratio of the channel rates of class  $k$  and class  $l$ , so as to describe the relationship between the HOL access delay and channel rate with heterogeneous and deterministic channel rate profile.

**Proposition 2.** *Given non-identical (heterogeneous) constant channel rates for the flows, when the number of flows in the system is sufficiently large, HAD can achieve fair channel time allocation among flows proportionally to their channel rates.*

*Proof.* We assume that every flow in the network has a non-empty queue. For simplicity, we consider that all the flows can be categorized into 2 classes, and  $\bar{N}_1$  and

$\bar{N}_2$  are the average number of flows in class 1 and class 2, respectively. The deterministic channel rates of class-1 and class-2 flows are  $R_{11}$  and  $R_{21}$ , respectively. We assume that the channel rate of class-1 flows is smaller than that of class-2 flows, i.e.,  $R_{11} \leq R_{21}$ . Note that  $\eta_{2,1}^R = R_{21}/R_{11}$  and  $\eta_{1,2}^H = \bar{H}_{1i}^{sch}/\bar{H}_{2j}^{sch}$ . Because the channel rates of class 1 and class 2 are constant values, we have  $H_{ki}^{sch}(t) = \max_{1 \leq i \leq N_k(t)} H_{ki}(t)$ . If a flow  $Q_{1i}(t)$  from class 1 is scheduled in time slot  $t$ , the earliest time for a flow  $Q_{2j}(t)$  to be scheduled is time slot  $t + 1$ , thus we have the following relationship:

$$\bar{H}_{1i}^{sch} \cdot R_{11} \geq (\bar{H}_{2j}^{sch} - 1) \cdot R_{21}. \quad (3.5)$$

We further clarify that  $\bar{H}_{1i}^{sch}$  is expected to be  $\bar{H}_{1i}^{sch} = \bar{N}_1 + \eta_{12}^H \bar{N}_2$ , and the explanation is as follows.  $\bar{H}_{1i}^{sch}$  is the average time that  $Q_{1i}(t)$  waits between the previous and next transmissions. During this time period, each of the other class-1 flows is expected to have one transmission considering that their HOL access delays are larger than  $Q_{1i}(t)$  in the first time slot after  $Q_{1i}(t)$ 's transmission, and thus  $\bar{H}_{1i}^{sch} \geq \bar{N}_1$ . Meanwhile, since we assume  $R_{11} \leq R_{21}$ , during the time period of  $\bar{H}_{1i}^{sch}$ , each of the class-2 flows can be scheduled once or more. As  $\bar{H}_{1i}^{sch}$  and  $\bar{H}_{2j}^{sch}$  are the average access waiting times for class-1 and class-2 flows to be scheduled, respectively,  $\bar{H}_{1i}^{sch}/\bar{H}_{2j}^{sch}$  represents on average how many times that a class-2 flow can be scheduled during  $\bar{H}_{1i}^{sch}$ , and thus  $(\bar{H}_{1i}^{sch}/\bar{H}_{2j}^{sch}) \cdot \bar{N}_2(t)$  means on average how many times that class-2 flows can be scheduled during  $\bar{H}_{1i}^{sch}$ . By calculating on average how many times all the other flows can transmit between  $Q_{1i}(t)$ 's previous transmission and next transmission, i.e., during the period of  $\bar{H}_{1i}^{sch}$ , we have  $\bar{H}_{1i}^{sch} = \bar{N}_1 + (\bar{H}_{1i}^{sch}/\bar{H}_{2j}^{sch})\bar{N}_2$ . With (3.5), we can further have:

$$\frac{\bar{N}_1 + \eta_{1,2}^H \cdot \bar{N}_2}{\frac{\bar{N}_1 + \eta_{1,2}^H \cdot \bar{N}_2}{\eta_{1,2}^H} - 1} \geq \eta_{2,1}^R.$$

The solution of  $\eta_{1,2}^H(t)$  can be found by solving the following inequality:

$$\eta_{1,2}^H \geq \frac{\eta_{2,1}^R(\bar{N}_2 - 1) - \bar{N}_1}{2\bar{N}_2} + \frac{\sqrt{(\bar{N}_1 - \eta_{2,1}^R(\bar{N}_2 - 1))^2 + 4\eta_{2,1}^R\bar{N}_1\bar{N}_2}}{2\bar{N}_2}. \quad (3.6)$$

When  $\bar{N}_2$  is large enough ( $\bar{N}_2 \gg 1$ ), the right-hand-side of (3.6) converges to

$$\frac{\eta_{2,1}^R \bar{N}_2 - \bar{N}_1 + \sqrt{(\bar{N}_1 - \eta_{2,1}^R \bar{N}_2)^2 + 4\eta_{2,1}^R \bar{N}_1 \bar{N}_2}}{2\bar{N}_2} = \eta_{2,1}^R.$$

Similarly, if the flow  $Q_{2j}(t)$  from class 2 is scheduled, we have

$$(\bar{H}_{1i}^{sch} - 1) \cdot R_{11} \leq \bar{H}_{2j}^{sch} \cdot R_{21},$$

which indicates:

$$\frac{\bar{N}_1 + \eta_{1,2}^H \bar{N}_2 - 1}{\frac{\bar{N}_1 + \eta_{1,2}^H \bar{N}_2}{\eta_{1,2}^H}} \leq \eta_{2,1}^R.$$

By solving this inequality, we have

$$\eta_{1,2}^H \leq \frac{\bar{N}_2 \eta_{2,1}^R - \bar{N}_1}{2(\bar{N}_2 - 1)} + \frac{\sqrt{(\bar{N}_2 \eta_{2,1}^R - \bar{N}_1)^2 + 4\eta_{2,1}^R \bar{N}_1 (\bar{N}_2 - 1)}}{2(\bar{N}_2 - 1)}. \quad (3.7)$$

When  $\bar{N}_2$  is sufficiently large, (3.7) converges to  $\eta_{1,2}^H \leq \eta_{2,1}^R$ . Hence we come to the conclusion that  $\eta_{1,2}^H = \eta_{2,1}^R$  by combining the results above.  $\square$

**Proposition 3.** *Given independent and non-identical (heterogeneous) channel rate distributions for the flows, when the number of flows is sufficiently large, HAD can achieve fair resource sharing among flows proportional to their maximum channel rates.*

*Proof.* We still consider a 2-class system, where the channel rate of flow  $Q_{ki}(t)$  from class  $k$  in time slot  $t$  is denoted by  $r_{ki}(t)$ . We use  $r_{ki}^{sch}(t^{(k)})$  to denote the channel rate when  $Q_{ki}(t^{(k)})$  is actually scheduled in time slot  $t^{(k)}$ , in which we specifically use  $H_{ki}^{sch}(t^{(k)})$  to denote the HOL access delay of  $Q_{ki}(t^{(k)})$ . Thus we have  $H_{1i}^{sch}(t^{(1)})r_{1i}^{sch}(t^{(1)}) \geq \max\{H_{2j}(t^{(1)})r_{2j}(t^{(1)})\}$  when  $Q_{1i}(t^{(1)})$  is scheduled.

In [9], it has been proved that when the number of flows in the system is sufficiently large,  $\mathbb{P}\{r_{ki}^{sch}(t) = R_k^{\max}\} = 1$ , which means that the scheduler is able to fully utilize the multi-user diversity gain to improve the throughput performance, and hence we have  $H_{1i}^{sch}(t^{(1)})R_1^{\max} \geq \max\{H_{2j}(t^{(1)})r_{2j}(t^{(1)})\}$ . Because the earliest following time for a flow of class 2 to be scheduled is  $t^{(1)} + 1$ , we have  $H_{1i}^{sch}(t^{(1)})R_1^{\max} \geq (H_{2j}^{sch}(t^{(2)})) -$

1)  $R_2^{\max}$ .

By taking the time average over the above inequality, we have

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T H_{1i}^{sch}(t) R_1^{\max} \geq \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T (H_{2j}^{sch}(t) - 1) R_2^{\max}. \quad (3.8)$$

Because the number of flows in the system is sufficiently large, and thus  $H_{2j}^{sch}(t^{(2)}) \gg 1$ . From (3.8) we have  $\mathbb{E}[H_{1i}(t)^{sch} \cdot R_1^{\max}] \geq \mathbb{E}[H_{2j}(t)^{sch} \cdot R_2^{\max}]$ . Similarly, we can obtain  $\mathbb{E}[H_{2j}^{sch}(t) \cdot R_2^{\max}] \geq \mathbb{E}[H_{1i}^{sch}(t) \cdot R_1^{\max}]$ . This indicates that when  $t \rightarrow \infty$ , we have

$$\frac{\mathbb{E}[H_{1i}^{sch}(t)]}{\mathbb{E}[H_{2j}^{sch}(t)]} = \frac{R_2^{\max}}{R_1^{\max}}. \quad (3.9)$$

□

## Remarks

Since the arrival rate of networks with flow-level dynamics refers to the number of new flows generated in one time slot, the arrival rate influences the total number of flows in the network. Given a wireless network, the larger the arrival rate is, the more flows we have in the network. The weighted fairness between flows remains the same no matter whether the arrival rates for HAD are small or large. The throughput ratio of two flows will change only when their channel profiles change. Considering that the arrival rate does not affect the fairness and the increase of HOL access delay of each individual flow in one time slot so long as each queue is non-empty, in summary, HAD is able to achieve fair scheduling among flows and thus can be adopted with TCP control schemes given various channel conditions.

One thing that we need to emphasize is that a scheduling algorithm with good fairness performance does not always necessarily mean the same HOL access delay for every flow if the flows belong to different classes. In practical networks, the resource allocation may be related to how much a customer pays for the service, and thus the scheduling algorithm should also accordingly assign the channel resources. To provide this type of differentiated services, we can simply assign a weight to each class, and use the multiplication of the weight and  $R_k^{\max}$  of each class to ensure the portion of channel time allocated to this class.

On the other hand, with TCP flows, if a throughput-optimal scheduling algorithm suffers the unfairness problem, such as QMW, the desirable throughput-optimality

may no longer hold. This is because the TCP controlled flow tends to avoid quick queue length increase, so with QMW scheduling, some flows may not have a sufficiently large size of the queue length to be scheduled. The HAD scheduling is compatible with TCP as it aims to allocate channel time to flows less dependent on its packet arrival process.

### 3.4.3 Throughput Analysis

With the analysis of the HOL access delay, we can further analyze the throughput relationship between different flows, which follows a  $\eta^2$ -rule as explained in the analysis below. The throughput analysis will be based on the HOL access delay analysis above, and thus we also follow the three cases discussed above.

#### Case 1

Given i.i.d. channel transmission rate distribution for all the system flows, we have  $\mathbb{E}[W_i(t)]/\mathbb{E}[W_j(t)] = 1$ , where  $W_i(t)$  denotes the throughput of flow  $Q_i(t)$  in time slot  $t$ .

This conclusion can be drawn from the result that  $\mathbb{E}[H_i^{sch}] = \mathbb{E}[H_j^{sch}]$ . Since the probability that flow  $Q_i(t)$  is scheduled, denoted by  $p_i$ , can be calculated as  $p_i = 1/\mathbb{E}[H_i^{sch}]$ , we know that  $p_i = p_j$ . Because the flows have i.i.d. channel distribution, considering that  $\mathbb{E}[W_i(t)] = p_i \cdot \mathbb{E}[R_i]$ , we can come to the conclusion that  $\mathbb{E}[W_i(t)]/\mathbb{E}[W_j(t)] = 1$ .

#### Case 2

Consider a 2-class network. Given non-identical (heterogeneous) constant channel rates for the flows, when the number of flows in the system is sufficiently large, we have  $\mathbb{E}[W_{1i}(t)]/\mathbb{E}[W_{2j}(t)] = 1/(\eta_{2,1}^R)^2$ , where  $\eta_{2,1}^R = R_{21}/R_{11}$ .

This conclusion can be drawn from the result in Proposition 2 that  $\eta_{1,2}^H = \eta_{2,1}^R$ . Given this relationship, we have  $p_{1i}/p_{2j} = \mathbb{E}[H_{2j}^{sch}]/\mathbb{E}[H_{1i}^{sch}] = 1/\eta_{2,1}^R$ , and thus the ratio of throughput  $\mathbb{E}[W_{1i}(t)]/\mathbb{E}[W_{2j}(t)] = (R_1/R_2) \cdot (p_{1i}/p_{2j}) = 1/(\eta_{2,1}^R)^2$ .

#### Case 3

Consider a 2-class network. Given independent and non-identical (heterogeneous) channel rate distributions for the flows, when the number of flows in the system is

sufficiently large, we have  $\mathbb{E}[W_{1i}(t)]/\mathbb{E}[W_{2j}(t)] = 1/(\tilde{\eta}_{2,1}^R)^2$ , where  $\tilde{\eta}_{2,1}^R = R_2^{\max}/R_1^{\max}$ .

This conclusion can be drawn from the result of (3.9) in the proof of Proposition 3. Given this relationship, we have  $p_{1i}/p_{2j} = \mathbb{E}[H_{2j}^{sch}]/\mathbb{E}[H_{1i}^{sch}] = 1/\tilde{\eta}_{2,1}^R$ , and the ratio of throughput  $\mathbb{E}[W_{1i}(t)]/\mathbb{E}[W_{2j}(t)] = (p_{1i}/p_{2j}) \cdot (R_{1i}^{\max}/R_{2j}^{\max}) = 1/(\tilde{\eta}_{2,1}^R)^2$ .

## 3.5 Performance Evaluation

To investigate the performance of the HAD scheduling algorithm for TCP flows, we conducted simulations in OMNeT++ 4.4.1 with the existing TCP Tahoe protocol implementation. We compared the performance of HAD with the MR, QMW, F-D-MW, and PF scheduling algorithms.

### 3.5.1 Network Setting

We consider centralized wireless networks such as the cellular networks in our simulation. The network topology in our simulation is shown in Fig. 3.3. In this network, a server is connected to the base station (BS) through a router. The BS can exchange messages with a number of wireless devices through the shared wireless channel. In the simulation, each client tries to establish a TCP connection with the server at a certain time, and then sends requests to the server. If the TCP connection is established successfully, the server will send the requested data back to the clients. For each TCP connection, the number of requests per TCP session follows exponential distribution with the mean value of 10 (requests); the request length follows truncated normal distribution with mean value of 20B and standard deviation of 5B; the reply length follows exponential distribution with mean value of 100MB; the re-connection interval is 10 seconds. More details of the network settings can be found in [20].

In our simulation, the scheduler is implemented in the BS. Each client has a dedicated queue in the BS. The packets sent to cli[i] is stored in Queue[i] ( $i \in \{0, 1, 2, \dots\}$ ). Each queue can store 100 packets, which is sufficiently large to avoid frequent packet dropping. How to minimize the buffer size while ensuring no degradation of performance is another important issue and out of the scope of this chapter [5, 33]. The scheduler determines which queue is chosen to transmit, and how many packets can be transmitted. When the packets are sent out, they will be delivered to the clients over the wireless channel. Based on the practice that wireless links tend to be the bottleneck link in a network, we assume that the bandwidth between the intermedi-

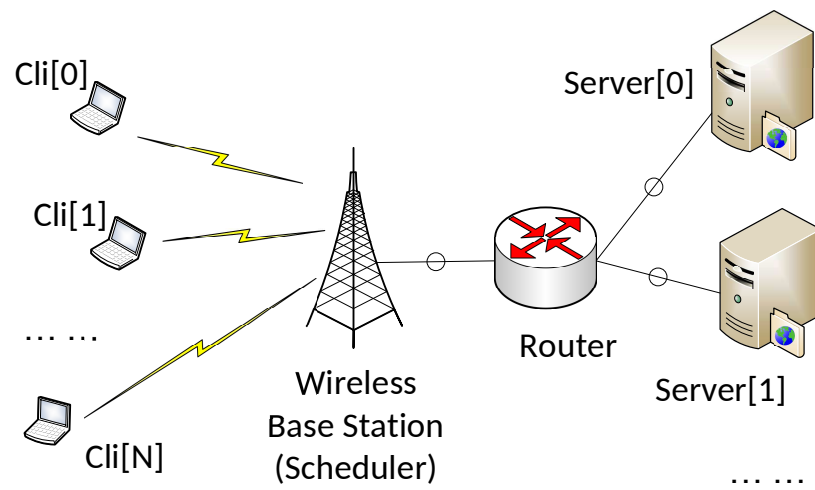


Figure 3.3: Network topology.

ate routers and the servers are large enough, and the bottleneck link is the wireless channel to/from the client, such that the packet delay jitter in other hops can be ignored compared to the delay in the wireless access links.

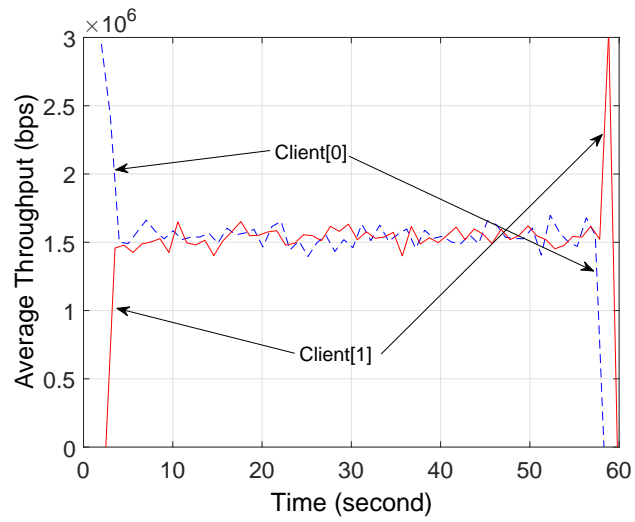
### 3.5.2 Homogeneous Networks with HAD

In this simulation, we focus on the fairness performance of the scheduling algorithms in a homogeneous network, in which the channel rate distribution of each flow is the same. The Jain's fairness index in terms of HOL access delay of each flow is always close to 1 in our simulation with various number of flows in the system, which validates our analysis of the HOL access delay in the homogeneous networks in Proposition 1. More information can be found in the comprehensive simulation results in Fig. 3.8.

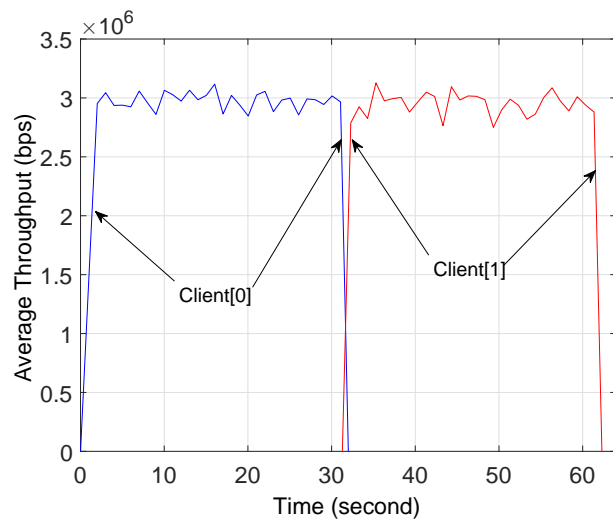
Next we investigate the throughput performance, and we begin with the simplest simulation scenario, in which there are only 2 clients (cli[0,1]). The channel rate of the wireless link in the network can be randomly selected from the set of {2Mbps, 3Mbps, 4Mbps} with the same probability. The starting time of the TCP connection of cli[0] and cli[1] follows exponential distribution with mean value of 0s and 2.5s, respectively. We compare the performance of HAD, QMW and F-D-MW in the 2-flow network scenario in Fig. 3.4. Fig. 3.4(a) shows the performance of HAD in terms of throughput, and the y-axis is the throughput of each client averaged over a sliding time window of 1 second. We can tell from the figure that before the beginning of cli[1]'s TCP session, cli[0] used the channel exclusively, but as long as cli[1] started its data request, the two clients in the network began to evenly share the network bandwidth.

Fig. 3.4(b) shows the fairness performance of classic QMW in terms of throughput, with the same network settings as those in Fig. 3.4(a). In Fig. 3.4(b), after the TCP connection between cli[0] and the server has been established, cli[0] dominated the channel usage. cli[1], however, suffered from the starvation, and was able to begin the transmission only after cli[0] received all the data. Fig. 3.4(c) shows the performance of F-D-MW in the same setting. Although cli[1] did not have the starvation problem and was able to have data transmission at the same time with cli[0], they could not share the channel resource evenly since the throughput of cli[1] is only about half of that of cli[0], which shows a noticeable unfairness.

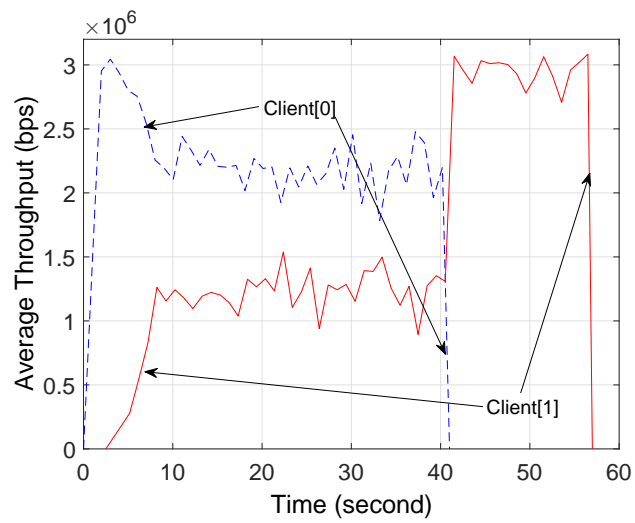
To further verify the fairness performance, we increased the number of flows in the system to 8, and the results are shown in Fig. 3.5. In this simulation, the channel



(a) HAD

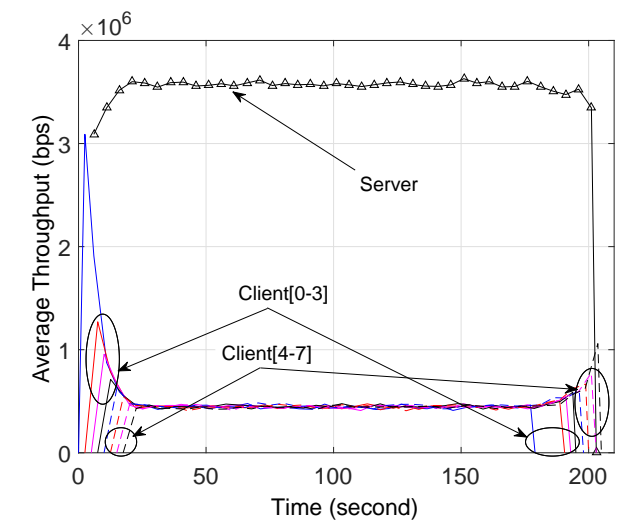


(b) QMW

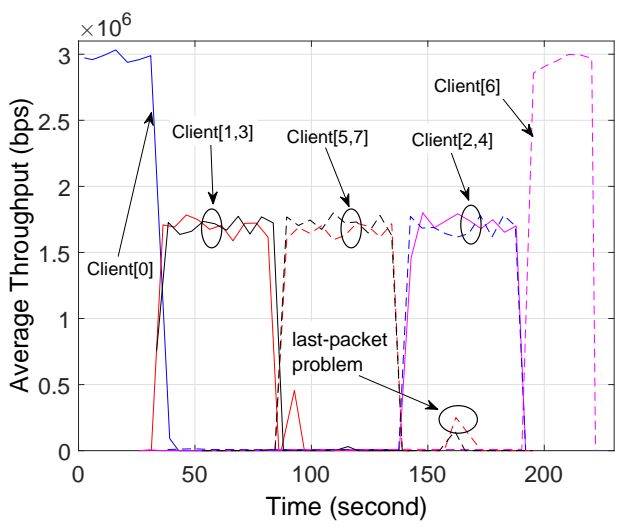


(c) F-D-MW

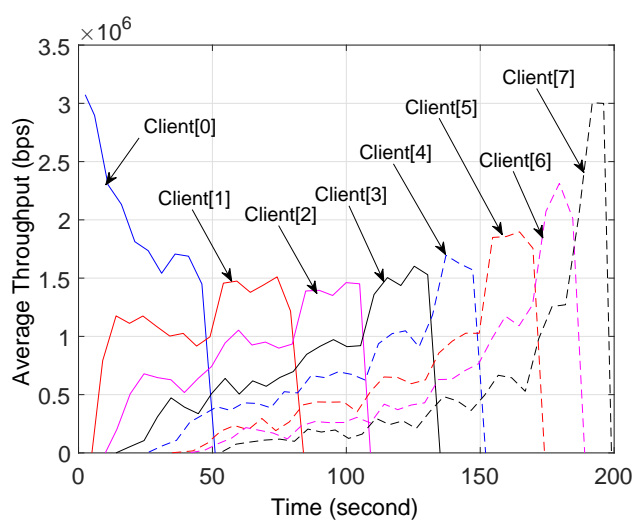
Figure 3.4: Throughput performance in the 2-flow homogeneous network.



(a) HAD



(b) QMW



(c) F-D-MW

Figure 3.5: Throughput performance in the 8-flow homogeneous network.

setting is the same as that in Fig. 3.4. The starting time of the TCP connection for cli[0-7] follows exponential distribution with mean values of 0s, 2.5s, 5s, 7.5s, 10s, 12.5s, 15s and 17.5s, respectively.

We can observe the performance of HAD in Fig. 3.5(a) that before the data request of cli[1], the average throughput of cli[0] kept increasing. But after cli[1-7] began their TCP sessions, instead of dominating the usage of the channel, cli[0] shared the channel resource with the other clients and its throughput decreased quickly to around 0.5Mbps, while the other 7 clients were also able to increase the throughput to about 0.5Mbps. This trend stayed stable for the rest of the simulation time until the end of the transmission. This observation verifies the desirable fairness performance of HAD. Furthermore, from the server's throughput performance, which indicates the total throughput of the network, we can observe that the network throughput is above the average channel rate. In fact, thanks to the opportunistic scheduling feature of HAD, we can achieve the multi-user diversity gain. Fig. 3.5(b) shows the performance of QMW, where cli[0] dominated the channel usage from the beginning to the end of its transmission. The last-packet problem of QMW mentioned in Sec. 3.2-B can also be observed in Fig. 3.5(b). Fig. 3.5(c) shows the performance of F-D-MW. The priority of the older flows in F-D-MW can be observed through the whole simulation. As a result, if a long-lived TCP flow exists, other flows may starve.

### 3.5.3 Heterogeneous Networks with HAD

In this simulation, we focus on the fairness performance of the scheduling algorithms in a heterogeneous network, in which the flows are categorized into two classes according to the various channel rate distributions. Fig. 3.6(a) shows the ratio of the average HOL access delay of the two classes, which have non-identical (heterogeneous) constant channel rates. Here the channel rates of the two class are 3Mbps and 5Mbps, so the channel rate ratio is 0.6 which is shown by the blue straight line. We can observe that the ratio of the HOL access delay converges quickly to the blue straight line as the number of flows increases. This verifies Proposition 2.

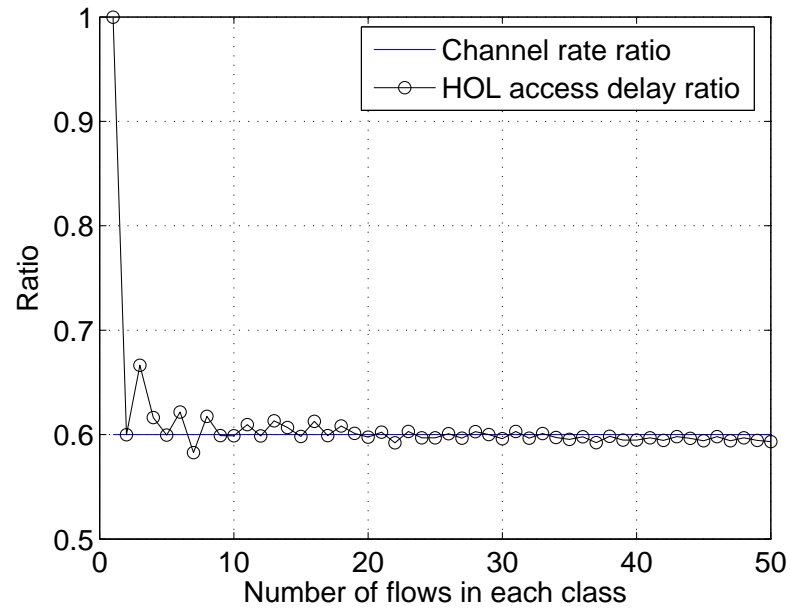
Fig. 3.6(b) shows the ratio of average HOL access delay of the two classes flows which have independent and non-identical (heterogeneous) channel rate distributions. In this figure, we use the two-state Markov channel model, and the available channel rate set for class 1 and 2 is  $\{2\text{Mbps}, 3\text{Mbps}\}$ , and  $\{5\text{Mbps}, 6\text{Mbps}\}$ , respectively. Each rate in the set has the probability of 0.5. The maximum channel rate ratio is

0.5 which is shown by the blue straight line. We can also observe that the ratio of the HOL access delay converges to the blue straight line as the number of flows increases. This also verifies the analysis in Proposition 3.

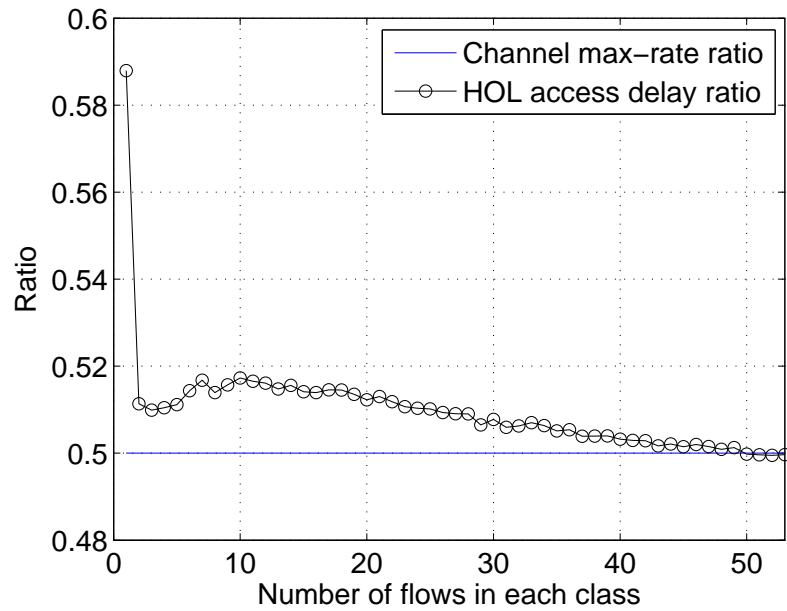
Fig. 3.7(a) shows the fairness performance of HAD in an 8-flow heterogeneous system, in which cli[0-3] belong to class 1 and cli[4-7] belong to class 2. The simulation results are averaged over a sliding time window of 5 seconds. The available channel rate sets for class 1 and 2 are {4Mbps, 5Mbps, 6Mbps} and {2Mbps, 3Mbps, 4Mbps}, respectively. The starting time of the TCP connection for cli[0-7] is 0s, 2.5s, 5s, 7.5s, 10s, 12.5s, 15s and 17.5s, respectively. We can observe that before 10s, cli[0-3] evenly shared the channel resource, which is the same as what we can expect in a homogeneous network. After 10s, cli[4-7] took turns to begin their TCP sessions. In this case, instead of suffering from starvation, cli[4-7] were able to occupy a portion of channel time. After 17.5s, the ratio of the mean window-averaged throughput between cli[0-3] and cli[4-7] is approximately 0.447, which is very close to the square of the maximum rate ratio ( $\eta_{1,2}^R = 0.444$ ). This observation also verifies our throughput analysis in Sec. 3.4.3.

For comparison, Fig. 3.7(b) shows the performance of MR in the heterogeneous network with channel variations. In Fig. 3.7(b), MR shows a very desirable total network throughput performance. Since MR tries to select the flow with the maximum possible channel rate to transmit, the total network throughput is very close to 6Mbps. But the flows from class 1 occupied the major part of the throughput, and the flows from class 2 had very little share of the network throughput due to the difference on the channel conditions. This observation shows the unfairness problem in MR in heterogeneous networks.

The performances of QMW and F-D-MW are presented in Figs. 3.7(c) and 3.7(d). In Fig. 3.7(c) we can see that, for QMW, after the TCP connection of cli[0] was established, the data transmission for cli[0]'s TCP session dominated the whole network throughput, while the other clients had to yield the channel resource to cli[0]. Since all (or most) of the requested data has been received, cli[0] finished its channel usage by 20s. Only after this time, one of the other clients had a chance to be scheduled. But similarly, this client also dominated the transmission in the whole network, and the remaining clients continued to suffer from starvation. This observation remains the same through the whole simulation time. Similar to Fig. 3.5(b), we can see the last-packet problem of QMW. We note that only one flow is scheduled at a time here while two clients can be schedule simultaneously in Fig. 3.5(b). This is because the



(a) Without channel variations



(b) With channel variations

Figure 3.6: HOL access delay ratio in the heterogeneous network.

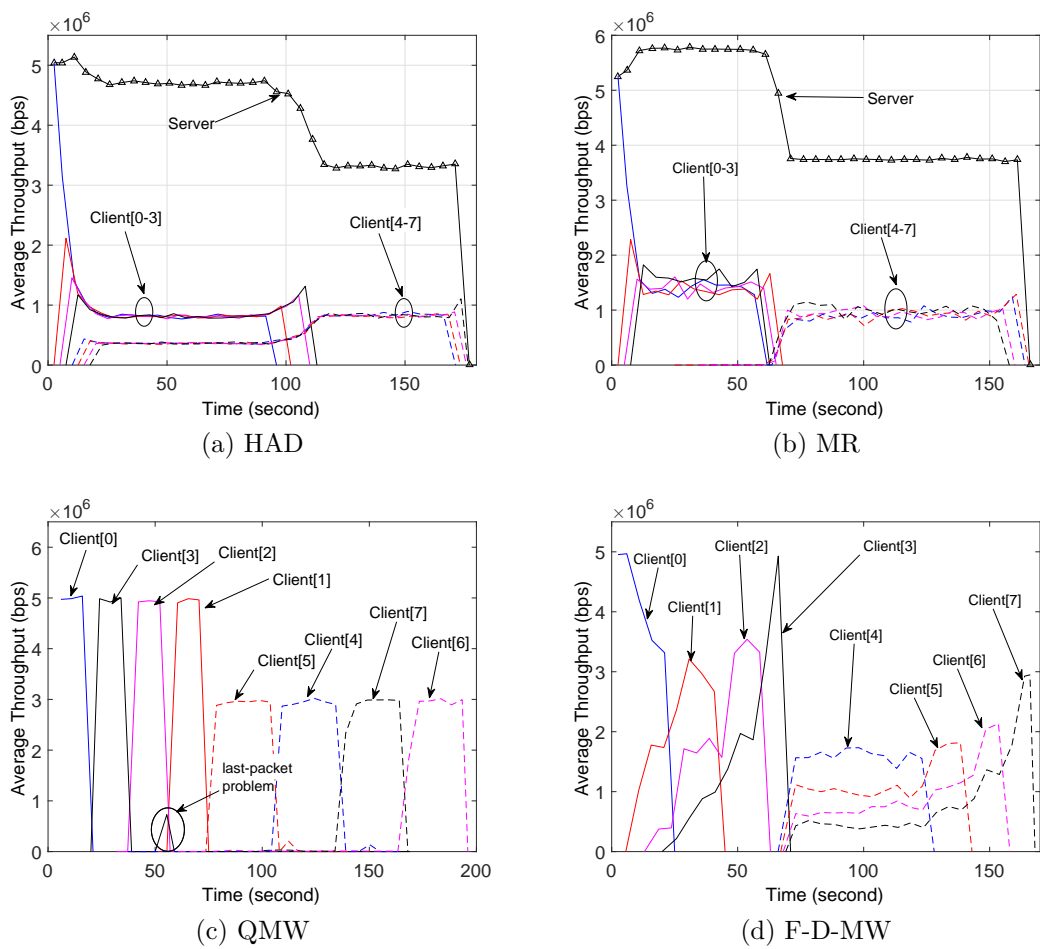


Figure 3.7: Throughput performance in the 8-flow heterogeneous network with channel variations.

RRT is smaller here according to the parameter settings, and the number of packets in one flow's queue can grow fast to earn enough priority such that the TCP connection of the other flows cannot be established. The performance of F-D-MW in Fig. 3.7(d) is similar to the one in Fig. 3.7(b), where the newer flows suffer from starvation till the old ones leave.

The fairness index in terms of the channel occupation time with an increasing number of flows in the homogeneous network is shown in Fig. 3.8(a). In this simulation, we used long-lived TCP flows in the homogeneous networks and counted the channel occupation of each flow over the period of 512 seconds. The Jain's index of all the flows in terms of channel occupation was investigated here. Let  $c_{ki}$  denote the channel occupation of  $Q_{ki}(t)$ . The fairness index of the homogeneous system  $\mathcal{J}_{homo}$  is calculated as

$$\mathcal{J}_{homo} = \frac{(\sum_{k=1}^K \sum_{i=1}^{N_k(t)} c_{ki})^2}{N(t) \sum (c_{ki})^2}.$$

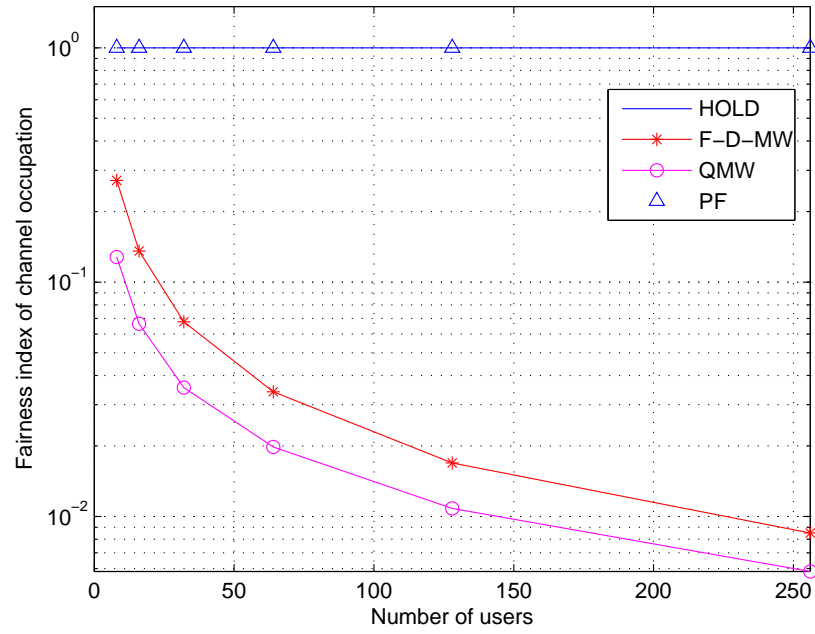
With the increase of the number of flows in the system, the fairness of HAD is not affected, and very close to 1 as the PF scheduling algorithm. However, for F-D-MW and QMW, since the channel time is shared by only one or a few number of flows in the system in a long period, with the increasing of flows in the system, the fairness index is monotonically decreasing.

In heterogeneous networks, we use weighted fairness index to measure the fairness of HAD. The weight of each flow  $w_{ki}$  is the maximum rate that can be achieved, and the channel occupation time is proportional to  $w_{ki}$ , so we use the normalized channel occupation time to measure the fairness in heterogeneous networks. The fairness index of the system  $\mathcal{J}_{hete}$  is calculated as

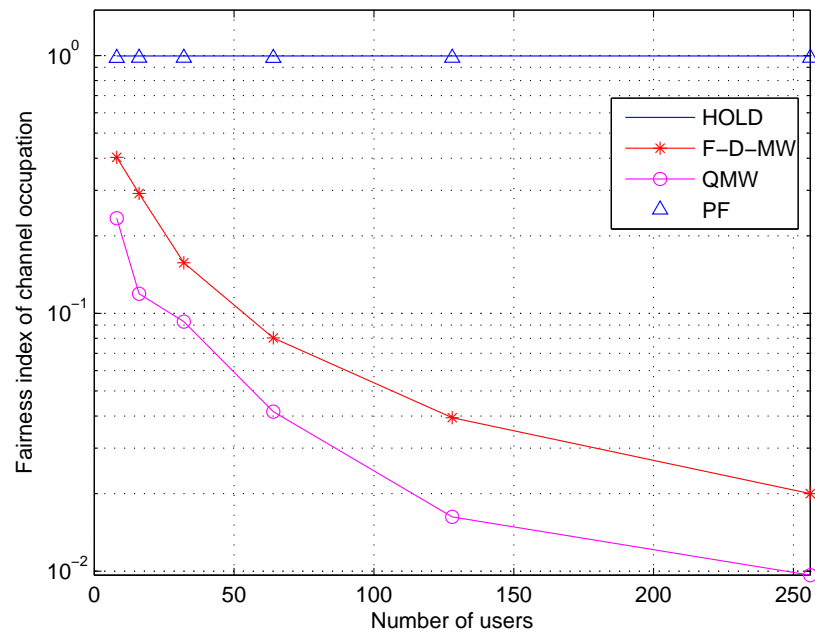
$$\mathcal{J}_{hete} = \frac{(\sum_{k=1}^K \sum_{i=1}^{N_k(t)} c_{ki} w_{ki}^{-1})^2}{N(t) \sum (c_{ki} w_{ki}^{-1})^2}.$$

With this definition, the weighted fairness of HAD is also verified to be as good as PF through simulation as shown in Fig. 3.8(b). Although QMW and F-D-MW are not designed to achieve weighted fairness, we put them together in Fig. 3.8(b) as a reference to the interested readers.

The corresponding system throughput of Fig. 3.8 is shown in Fig. 3.9, which indicates that the proposed HAD algorithm not only provides fair resource allocation as PF, but also maintains the throughput performance in very high level as that of MR and F-D-MW, especially when the number of flows in the system is sufficiently



(a) Homogeneous network



(b) Heterogeneous network

Figure 3.8: Jain's fairness index in terms of channel occupation.

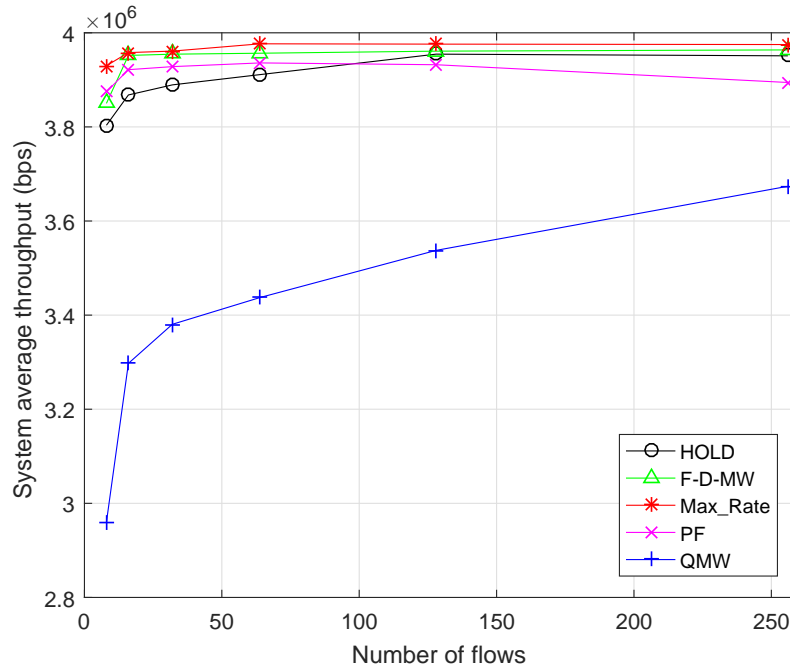


Figure 3.9: System throughput of different algorithms with increasing number of flows.

large. Different from those with HAD, MR and F-D-MW, with the increase of the number of flows, the throughput of PF decreases since its first priority is to guarantee fairness. On the contrary, the throughput of QMW is noticeably lower.

### 3.5.4 System Stability

The simulation in Fig. 3.10 investigates the throughput-optimality of different scheduling algorithms with flow-level dynamics. We used a two-state channel model, and considered heterogeneous network with 5 classes of flows with the traffic intensity 0.99. The number of flows in Fig. 3.10 is the number of simultaneous backlogged flows in the network in any given slot. Consider that each flow arrives in the system with a finite amount of data to transmit, and leaves the system once all the data are transmitted. When the system is stable, we have a finite number of flows in the system. When the number of flows grows into infinity when  $t \rightarrow \infty$ , the total amount of data in the system also grows into infinity and thus we have system instability [52]. In the simulation, the number of flows in the system can be stabilized by MR and F-D-MW which are proved to be throughput-optimal in the literature, while the pro-

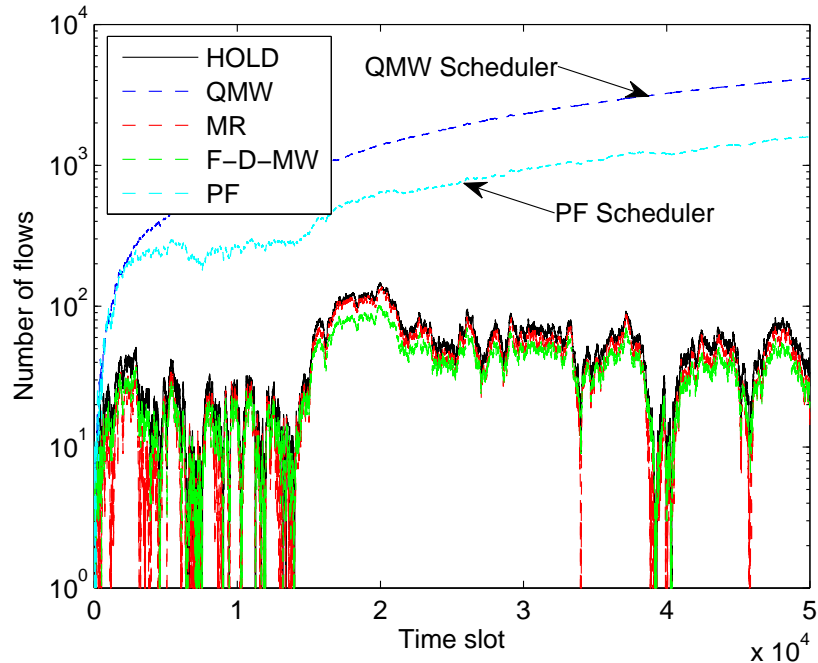


Figure 3.10: Throughput-optimality test.

posed HAD algorithms has almost the identical performance compared with MR and F-D-MW which confirms the throughput-optimality of HAD. In contrast, the system cannot be stabilized by QMW, which is shown not to be throughput-optimal with flow-level dynamics. Since in the current cellular networks, the most widely used scheduler is the PF algorithm, we also included PF in the simulation. The history update window was set to be 1000 slots as recommended in [22]. Fig. 3.10 shows that PF is not throughput-optimal because it is not able to stabilize the system when other provable throughput-optimal schedulers can. This result also validates the conclusion in [2, 58], which offered a number of examples to show the instability of PF, and the theoretical proof showing that the stability region is less than the capacity region, respectively.

### 3.6 Conclusion

In this chapter, we have studied the compatibility between the TCP congestion control scheme and HAD scheduling algorithm. Since we observed that other throughput-optimal scheduling algorithms, e.g., QMW and F-D-MW, encounter the starvation problem when scheduling TCP regulated flows, we designed HAD scheduling algo-

rithm which is shown to be throughput-optimal and compatible with TCP flows through theoretical analysis.

To verify the theoretic results, comprehensive simulations have been conducted to compare the performances of HAD with QMW, F-D-MW and MR in both homogeneous and heterogeneous systems. Simulation results have validated the theoretical analysis, and demonstrated the superior performance of HAD when serving TCP flows compared to the existing solutions in terms of fairness. We also have completed the comparison between HAD and the widely adopted PF scheduler. The result of system stability have validated that HAD is throughput-optimal while the PF scheduler is not throughput-optimal.

The works in this and the previous chapter have shown that HAD is an efficient and practical scheduling algorithm. The conclusion of throughput-optimality implies the existence of the upper bound of the number of dynamic flows, which is, however, still not enough for the implementation of the scheduler. If the system is built too large, there would be a waste of resource and cost. If the system is built too small, there could be a performance degradation due to the insufficient utilization of system capacity. As a result, the analysis of the queueing behavior of the designed scheduler is required as the guideline of algorithm implementation, which motivates our work in the next chapter.

## Chapter 4

# Queuing Behavior Analysis of HAD Scheduling

### 4.1 Introduction

In Chapter 2, we have addressed and solved the problem of throughput-optimal scheduling algorithm. We have proposed the HAD scheduling algorithm which is able to stabilize the system with flow-level dynamics. In Chapter 3, we have further demonstrated that HAD can support TCP regulated flows. The works in the previous two chapters show that HAD is an efficient and practical throughput-optimal solution for systems with dynamic flows. From the theoretical analysis we know that the number of flows  $N(t)$  and the total backlog  $|Q(t)|$  in the system are bounded with HAD, but having the upper bound only is not sufficient for the implementation of algorithm, and it is important to investigate how much  $N(t)$  will be when they converge to the steady state, as well as the delay performance of HAD. In this section, we focus on the theoretical analysis of the queueing behaviour of HAD scheduling, including the expectation of  $N(t)$  and the delay performance.

As shown in Chapter 3, HAD is able to achieve a certain level of fairness between flows. Generally speaking, other throughput-optimal scheduling algorithms tend to let one or a few flows exclusively occupy the channel resource for a long time, while by adopting HAD, flows in the system are able to fairly share the transmission opportunities. Although TCP flows are considered [10], the fairness can be achieved when scheduling other non-TCP controlled flows. We here focus on the study of queueing behaviour of HAD by the approach of queueing theory.

## 4.2 Queuing Behavior Analysis

For simplicity, we investigate a homogeneous network with one class of flows, and assume that the arrival rate satisfies  $A(t) \leq 1$ . Let  $H_i^{sch}(t)$  denote the HOL access delay when  $Q_i(t)$  is scheduled. If we focus on homogeneous networks, it is shown that if all the flows have i.i.d. channel rate distribution, i.e.,  $\mathbb{E}[r_i(t)] = \mathbb{E}[r_j(t)]$  and  $r_i^{\max} = r_j^{\max}$ , we have  $\mathbb{E}[H_i^{sch}(t)] = \mathbb{E}[H_j^{sch}(t)]$  [10]. Considering that  $P\{r(t) = R^{\max}\} = 1$  if HAD is adopted when  $\rho$  is large enough, we have

$$\mathbb{E}[H_i^{sch}] = \mathbb{E}[N(t)]. \quad (4.1)$$

Assume that the length of the time slot is  $\delta$ , and hence the maximum transmission rate in one time slot is  $\delta R^{\max}$ . Let  $\mathbb{E}[B_i(t)] = \bar{B}$  denote the mean value of the initial queue length of the flows if  $B_i(t)$  is a random variable with a finite second order moment. Let  $T_i^{tx}(t)$  denote the number of time slots to finish the transmission for  $Q_i(t)$  which has a finite amount of data to transmit, and the expectation of  $T_i^{tx}(t)$  is:

$$\mathbb{E}[T_i^{tx}(t)] = \mathbb{E}[H_i^{\max}] \cdot \lceil \frac{\bar{B}}{\delta R^{\max}} \rceil = \mathbb{E}[N(t)] \cdot \lceil \frac{\bar{B}}{\delta R^{\max}} \rceil. \quad (4.2)$$

The average throughput of  $Q_i(t)$ , defined as  $\mathbb{E}\{W_i(t)\}$  can be calculated by

$$\mathbb{E}[W_i(t)] = \frac{R^{\max}}{\mathbb{E}[N(t)]}. \quad (4.3)$$

From the above analysis we can see that the average time that one dynamic flow stays in the system and the throughput of each flow are related to the parameters including the initial queue length, the maximum channel rate, the time slot duration and the average number of flows in the system. Next we will figure out how to calculate the average number of flows in the system in order to analyze the delay and throughput performance. In the following analysis, we assume that  $\delta = 1$  time unit, and thus  $\delta$  can be omitted in the equations.

We will adopt the same system model in Chapter 2 for the analysis except that we adopt constant dynamic flow size to simplify the approximation. We will skip the details of the system model introduction, and jump directly into the queuing behavior analysis.

### 4.2.1 State-dependent Markov Model

In a homogeneous wireless network working in slotted time with flow-level dynamics, if HAD scheduling algorithm is adopted, the number of flows in the system in one time slot can be described by a discrete time Markov chain as a queueing system, which is shown in Fig. 4.1. The state  $S_i$  represents that the number of flows in the system in a time slot is  $i$ , and  $S_i = S_{i-1} + 1$ . Suppose in time slot  $t$ , we have  $N(t) = S_i$ . From  $t$  to  $t + 1$ ,  $N(t)$  is possible to change from  $S_i$  to  $S_{i-1}$  or  $S_{i+1}$ , or stay in the state of  $S_i$ , depending on if there is a new flow's arrival or an old flow's departure. We define the transit probability as  $p_a = P\{N(t + 1) = N(t) + 1\}$ ,  $p_b = P\{N(t + 1) = N(t) - 1\}$ , and  $p_c = P\{N(t + 1) = N(t)\} = 1 - p_a - p_b$ .

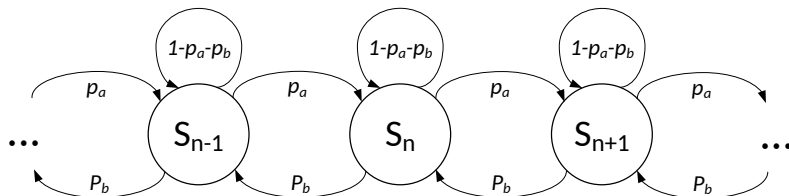


Figure 4.1: Markov chain of  $N(t)$ .

The scheduler behaves differently depending on whether the traffic intensity  $\rho$  is small or big. When  $\rho$  is small, the scheduler is able to stabilize the system without holding the condition in (2.3) to be always true. When  $\rho$  is close to 1, the condition in (2.3) needs to be satisfied to achieve system stability. Thus, the average scheduled transmission rate and the transit probability of the Markov process are dependent on the state of the system. We use  $R^{avg}(n)$  to denote the average scheduled transmission rate when the system is in state  $S_n$ , the state transition probability can be calculated as follows:

$$\begin{cases} p_1 = P\{one\ arrival\}, \\ p_{ai} = p_1(1 - \frac{1}{\bar{w}(i)}), \\ p_{bi} = (1 - p_1)\frac{1}{\bar{w}(i)}, \\ p_{ci} = (1 - p_{ai} - p_{bi}). \end{cases} \quad (4.4)$$

In (4.4),  $\bar{w}(i)$  is the state-dependent average work load of each flow in terms of the necessary time slots for the transmission of the whole flow, and can be calculated as  $\bar{w}(i) = \lceil \bar{B}/R^{avg}(i) \rceil$ . We consider how to obtain  $R^{avg}$  of a network in which all the nodes have a homogeneous channel profile. As mentioned, for each flow, the transmission rate can be chosen from the channel rate set  $\mathcal{R}$  which has finite supports,

i.e.,  $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$ , where  $m$  is a positive integer. The maximum channel rate in  $\mathcal{R}$  is denoted as  $R^{\max}$ . For simplicity, we consider  $m = 2$  and  $R_1 < R_2$ , and define  $R^{\text{slow}} = R_1$ . When  $m > 2$ , we use the following approximation in the analysis:

$$R^{\text{slow}} = \left\{ \frac{1}{m-1} \sum_i R_i \mid R_i \in \mathcal{R}, R_i \neq R^{\max} \right\}.$$

Let  $\mathcal{U}(t)$  denote the set of flows in which all the flows have the HOL access delay larger than that of the rest of the other flows in the system at time  $t$ . We use  $M(i)$  to denote the number of flows in  $\mathcal{U}(t)$ , i.e.,  $|\mathcal{U}(t)| = M(i)$  when the system is in state  $S_i$ . Since the flows with larger HOL access delay have the priority to be scheduled in HAD, the scheduler will choose a flow with  $R^{\text{slow}}$  at time slot  $t$  when all the  $M$  flows in  $\mathcal{U}(t)$  are in the channel state  $R^{\text{slow}}$ , where  $M(i)$  satisfies the following inequality considering (4.1):

$$R^{\text{slow}} \cdot i \geq R^{\max} \cdot (i - M(i)).$$

Further, we have  $M(i) \geq \lceil i(1 - \frac{R^{\text{slow}}}{R^{\max}}) \rceil$  in the time-slotted system, and thus we use

$$M(i) \approx i(1 - \frac{R^{\text{slow}}}{R^{\max}}) \quad (4.5)$$

in our analysis.

The number of flows in the system is changing along with the variance of the traffic intensity  $\rho$ . With the calculation of  $M(i)$ , the average channel rate of the system in state  $S_i$  can be calculated as

$$R^{\text{avg}} = (1 - P\{r_i(t) = R^{\max}\})^{M(i)} \cdot R^{\text{slow}} + [1 - (1 - P\{r_i(t) = R^{\max}\})^{M(i)}] \cdot R^{\max}. \quad (4.6)$$

The difference equations of (4.4) are shown as follows:

$$\begin{cases} P_i(p_{ai} + p_{bi}) = P_{i+1} \cdot p_{b(i+1)} + P_{i-1} \cdot p_{ai}, \\ P_0 p_a = P_1 p_{b1}. \end{cases} \quad (4.7)$$

The solution of (4.7) is

$$P_0 = \frac{1}{1 + \sum_{j \geq 1} \frac{\prod_{i=0}^{j-1} p_{ai}}{\prod_{i=0}^j p_{bi}}}. \quad (4.8)$$

and

$$P_n = \frac{\prod_{i=1}^{n-1} p_{ai}}{\prod_{i=1}^n p_{bi}} P_0 = \frac{p_1^{n-1} \prod_{i=1}^{n-1} \left(1 - \frac{1}{\bar{w}(i)}\right)}{(1-p_1)^n \prod_{i=1}^n \frac{1}{\bar{w}(i)}} P_0 = \frac{p_1^{n-1} \prod_{i=1}^{n-1} \frac{\bar{B} - R^{avg}(i)}{\bar{B}}}{(1-p_1)^n \prod_{i=1}^n \frac{R^{avg}(i)}{\bar{B}}} P_0, \quad (4.9)$$

The number of flows in the system can be calculated by  $\mathbb{E}[N(t)] = \sum_i i \cdot P_i$  with the definition of  $\bar{w}(i)$ , (4.8) and (4.9). However, this method is very complicated. We further developed an approximation method to analyze the system.

## 4.2.2 Approximation

First we analyze the scenario when  $\rho$  is small such that (2.3) is not necessary for system stability. In this case, the system can also be described as shown in Fig. 4.1, of which the transit probability and the balance equations can be found in (4.10).

$$\left\{ \begin{array}{l} \bar{w} = \lceil \bar{B}/R^{avg} \rceil, \\ p_1 = P\{\text{one arrival}\}, \\ p_a = p_1 \left(1 - \frac{1}{\bar{w}}\right), \\ p_b = (1-p_1) \frac{1}{\bar{w}}, \\ p_c = (1-p_a-p_b), \\ P_n(p_a + p_b) = P_{n-1}p_a + P_{n+1}p_b, \\ P_0p_a = P_1p_b. \end{array} \right. \quad (4.10)$$

In (4.10),  $\bar{w}$  and  $R^{avg}$  are defined the same as that in (4.4). Different from the analysis with the state-dependent Markov process, here we let  $M$  denote the average number of flows in  $\mathcal{U}(t)$ , i.e.,  $\mathbb{E}\{|\mathcal{U}(t)|\} = M$ , and thus (4.5) becomes

$$M \approx \lceil \mathbb{E}\{N(t)\} \left(1 - \frac{R^{slow}}{R^{\max}}\right) \rceil, \quad (4.11)$$

and (4.6) becomes

$$R^{avg} = (1 - P\{r_i(t) = R^{\max}\})^M \cdot R^{slow} + [1 - (1 - P\{r_i(t) = R^{\max}\})^M] \cdot R^{\max}. \quad (4.12)$$

Using the average channel rate to build the Markov chain as described in (4.10),

the solution can be found in (4.13).

$$\left\{ \begin{array}{l} P_0 = 1 - \frac{p_a}{p_b}, \\ P_n = \left(1 - \frac{p_a}{p_b}\right) \left(\frac{p_a}{p_b}\right)^n, \\ E[N] = \sum_{n=0}^{\infty} n P_n = \frac{p_a}{p_b - p_a} = p_1 \frac{1 - \frac{1}{\bar{w}}}{\frac{1}{\bar{w}} - p_1}, \\ E[H^{\max}] = E[N], \\ p_{ki} = \frac{R_k^{\max}}{\sum_k N_k R_k^{\max}}, \\ S_{ki} = \frac{(R_k^{\max})^2}{\sum_k N_k R_k^{\max}}, \\ E[T_i^{tx}] = E[N] \bar{w}. \end{array} \right. \quad (4.13)$$

We refer this analytical model as the M/M/1 approximation. Since (4.11) is involved in (4.13) to find  $\mathbb{E}\{N(t)\}$  which brings extra computational complexity, we consider to further simplify the analysis by proposing the M/M/1-M approximation with two iterations of the M/M/1 approximation. In M/M/1-M, we solve the M/M/1 model for two times. The result of the first time is used in the second time. We first solve (4.10) by setting  $\bar{w} = \lceil \bar{B}/R^{\max} \rceil$  to obtain an  $\mathbb{E}[N(t)]$  in the first iteration, and then plug it in (4.11) to obtain the M/M/1-M approximation results by solving (4.10) again. Although we have two iterations in the M/M/1-M approximation, the computational complexity is reduced since we can avoid the exponential (or logarithm) computation in the M/M/1 approximation. However, the accuracy of the M/M/1-M approximation may be compromised, which is explained as follows. Since  $N(t)$  is conservatively approximated with  $\bar{w} = \lceil \bar{B}/R^{\max} \rceil$  in the first iteration, the average transmission rate is likely to be reduced in the second iteration, and as a result,  $N(t)$  of the M/M/1-M approximation may be slightly larger than the actual situation.

Next we consider the scenario when  $\rho$  is close to one. In this case, since (2.3) becomes necessary for stability, we can use an M/D/1 Markov model to calculate, where

$$\bar{w} = \lceil \bar{B}/R^{\max} \rceil.$$

The M/D/1 Markov model is more complex than the M/M/1-M model, because to define the system state, we need to not only record the number of flows in the system, but also the number of packets in each flow, which make the whole Markov

chain a multiple dimension one. Considering that when the traffic intensity of the system is close to one, the system instability is more likely to happen, and for a throughput-optimal scheduler, as long as it is possible, it will schedule flows that are in the maximum possible channel condition so that the multiuser diversity gain can be fully explored. This behavior makes the data departure rate similar to that in an M/D/1 queueing system. To simplify the problem, we directly apply the queueing analysis results of the M/D/1 model (i.e., (4.1) in [15]) to obtain an approximation solution. The average number of flows in the system can be approximated by

$$E[N] = \frac{\left(\frac{2p_a}{p_b} - \left(\frac{p_a}{p_b}\right)^2\right)}{2\left(1 - \frac{p_a}{p_b}\right)}. \quad (4.14)$$

In this case, we refer the analytical model as M/D/1 approximation. By substituting the results in (4.13) and (4.14) into (4.2) and (4.3), we are able to obtain the desired performance results.

The difference between M/M/1-M approximation and M/D/1 approximation is that in the M/M/1-M approximation, the probability that each queue is served in the maximum channel rate is not 1. This implies that we can treat the result of the M/M/1-M approximation for a relatively small  $\rho$ , while taking the result of M/D/1 approximation when the traffic intensity is close to one which is the boundary of the network capacity. Our analysis results can also be treated as a reference of the other throughput-optimal scheduling algorithms, considering that HAD performs similarly to the other throughput-optimal algorithms.

### 4.3 Performance Evaluation

To verify the analysis of HAD in terms of  $N(t)$  and  $T_i^{tx}(t)$  in Sec. 4.2, we compared the analytical and simulation results, which are shown in Figs. 4.2-4.3. In our work, we adopt M/M/1-M and M/D/1 approximations to exam the accuracy of the analysis. In this simulation, we have one class of flows in the system and further increased the traffic burst size to 70 units.

In Fig. 4.2, the M/M/1-M and M/D/1 approximation result for  $N(t)$  is shown as the red curve marked with upward-pointing triangles and stars, respectively. When  $\rho$  is small, the M/M/1-M approximation is very close to  $N(t)$  of HAD, while the

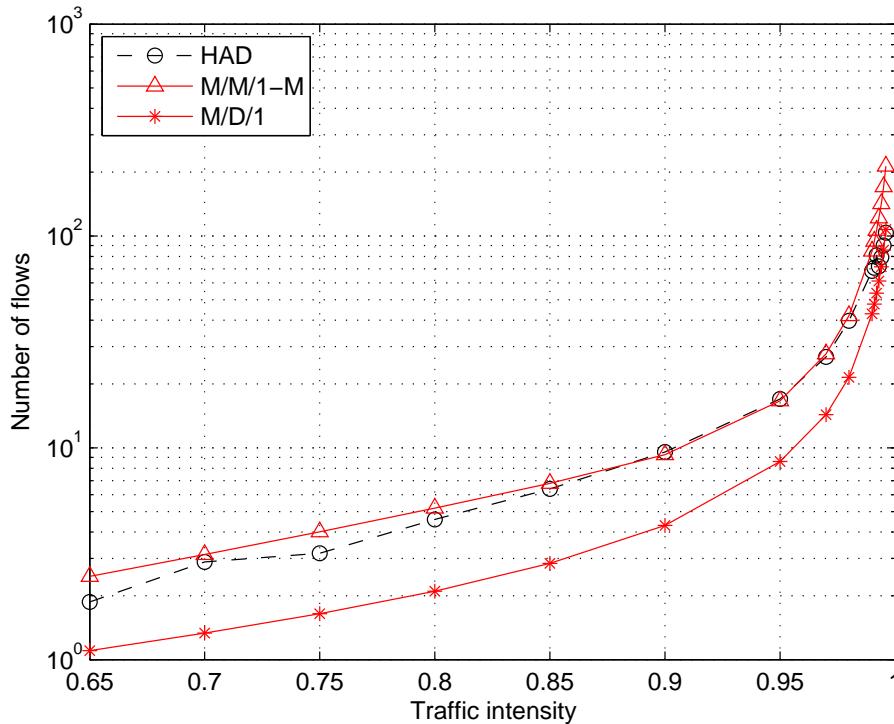


Figure 4.2: Number of flows in the system with varying  $\rho$ .

M/D/1 approximation is shown as the lower bound. When  $\rho$  is close to 1, the M/D/1 approximation can better describe the system behaviour.

The delay performance is shown in Fig. 4.3. Before  $\rho \leq 0.97$ , the M/M/1-M approximation result for  $T_i^{tx}(t)$  can accurately describe the flow sojourn time in the system, which is getting increasingly closer to the M/D/1 approximation when  $\rho$  increases from 0.97 to 1. This result verifies our previous analysis. The error bar in the figure represents the confidence interval of 95% of the delay of the HAD scheduler. We also included the simulation results of the flow delay of F-D-MW, which lies in between the M/M/1-M and M/D/1 approximation results for most of the situations except when  $\rho$  is extremely large. In this figure, the first-bit delay is the waiting time between the moments of the entrance a flow in the system and the first schedule of this flow, which is referred as the start-up latency. From the simulation we can also observe that HAD has a much shorter start-up latency which is the key to support real-time data, while the flow delay of HAD is only marginally larger than that of F-D-MW.

From all the above simulation results, we can observe that, the HAD scheduling

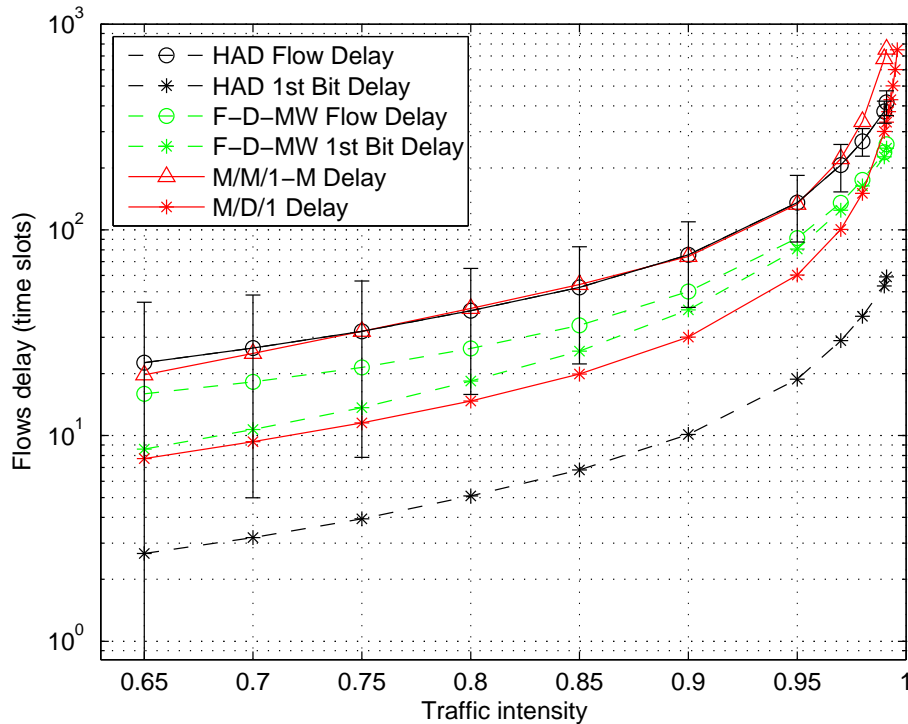


Figure 4.3: Average queue delay with varying  $\rho$ .

algorithm is not only able to maintain the stability and throughput-optimality with flow-level dynamics in a heterogeneous system, but also provides better fairness among flows, which is a desirable feature. With HAD, a new flow in the system does not have to wait for a long time before the first transmission, while with the F-D-MW scheduling, the first few packets in a new flow have to wait for a long time to be transmitted, which may result in a large start-up latency. The fact that HAD does not require any prior knowledge of the arrival process and the channel rate distribution makes it easier to implement.

## 4.4 Conclusion

In this chapter, we have investigated the analytic model for the proposed HAD scheduling algorithm in the system with dynamic flows. Based on the previously obtained results, we have proposed a Markov chain model to describe the evolution of the system with flow-level dynamics. The analytic model has then been solved by finding out the transit probability and the probability of each state. Since the direct

solution is too complex to verify, we have further studied how to simplify the solution. Two approximation approaches have been proposed corresponding to different stage of the traffic intensity. We have verified the analysis result by conducting performance evaluation in MATLAB. The simulation results match the analytic result very well.

## Chapter 5

# Throughput-optimal Scheduling for Hybrid Systems

### 5.1 Introduction

A scheduling algorithm is throughput-optimal if it can always achieve the network queue stability given any traffic arrival rate vector that lies strictly within the capacity region. As introduced in Chapter 2, the pioneer works of Tassiulas and Ephremides [49–51] proposed the QMW scheduler, and proved that it is a throughput-optimal strategy. The details of the scheduling rule of QMW can be found in Chapters 2 and 3. It became a popular research topic since the scheduling strategy of QMW is simple while it can achieve throughput-optimality. The following works extended throughput to other metrics such as the delay performance [39], energy consumption [36, 41], and fairness [17, 40], etc.

Although QMW presents desirable throughput performance, one necessary condition is that the network consists of only a fixed number of *persistent flows* which are long-lived and have continuous data injection. For machine-type applications in some real world networks, such as the sensor networks or vehicular networks, persistent flows may exist. However, *dynamic flows* are commonly observed for human-to-human communication applications. Dynamic flows have finite amount of service requests upon arrival in the network, and leave the system once the demanded services are fulfilled. Since flows arrive and leave the system over time, the number of flows in the system may change from one slot to the next. The examples of applications with dynamic flows include the email/text messages delivered from one people

to another, web browsing, etc. In networks with dynamic flows, QMW is no longer throughput-optimal [52].

There have been a few solutions to design scheduling algorithms for the systems with dynamic flows [32, 46], and the majority of the existing works considered the networks that exclusively include the flows of only one type, i.e., either persistent or dynamic flows, rather than both of them. However, the coexistence of the persistent and dynamic flows cannot be ignored in practice, e.g., in 5G cellular systems, both machine-to-machine and human-to-human applications share the same spectrum, and thus it is important to design the corresponding throughput-optimal scheduling algorithms. The approach of separating the two types of flows and scheduling them independently is not the best choice, because separating the resources for two types of flows will result in a lower multiplexing gain and lower efficiency.

All above mentioned issues motivate us to investigate the scheduling algorithm for the hybrid systems with both persistent and dynamic flows. The contributions of this chapter are three-fold. First, we propose a scheduling algorithm for the networks with both persistent flows and dynamic flows. Second, we extend the algorithm to a general form to represent a class of algorithms so that the algorithm design can be more flexible and adaptive to different QoS requirements. Third, through the performance evaluation we reveal that the offline MaxRate scheduling algorithm (MR), throughput-optimal for dynamic flows, is not throughput-optimal when dynamic flows coexist with persistent flows.

## 5.2 Related Works

The application of throughput-optimal QMW has been found in a wide range of research areas such as scheduling design with secretary guarantee [59], smart grid [63] and wireless sensor networks [7]. Besides QMW, queue length based throughput-optimal scheduling has other variations [21, 25, 44, 45, 48]. But it has been revealed that all these schedulers are not throughput-optimal if the system consists of dynamic flows [52, 53]. The authors of [52] also proposed a scheduling algorithm to stabilize the systems of dynamic flows, which is an off-line scheduler requiring the knowledge of the channel profile. Subsequent works have developed various of scheduling algorithms that were proved to be throughput-optimal for dynamic flows [9, 33]. For example, the MaxRate scheduling algorithm (MR) was studied in [33], which always selects the flows in the system when they are associated with their maximum possible

transmission rates. MR is an off-line scheduling algorithm because it needs to know the channel profile in order to know when the maximum transmission rate is reached. An on-line alternative was proposed in the same paper with the introduction of a learning period to know what could be the best channel condition, and the length of the learning window is a sensitive parameter. MR has been considered as the benchmark of the throughput-optimal scheduling algorithms for the systems with the dynamic flows. However, its performance in the hybrid systems with the coexistence of persistent and dynamic flows is unknown. Other works focused on the distributed implementation of throughput-optimal schedulers [27, 29, 60].

Along with the queue length based scheduling algorithms, F-D-MW [3, 18, 35] has been shown to be throughput-optimal for the persistent flows as well. F-D-MW gives the priority to the flows whose waiting time of the head-of-line packets are the longest in the system. F-D-MW has been investigated widely in the subsequent works. [38] studied the network utility maximization with F-D-MW in wireless systems. [23] developed the delay based back-pressure throughput-optimal scheduler for multihop wireless networks. Considering flow-level dynamics, the work in [46] showed that F-D-MW is also throughput-optimal by applying it to the systems of dynamic flows. [28] revealed that F-D-MW can be applied to the hybrid system with the presence of both persistent and dynamic flows. There are two problems remaining to be explored. First, the channel rate variation is not considered. Second, the delay performance of F-D-MW is not desirable. By adopting F-D-MW, the new flows in the system may suffer a long start-up latency after arriving in the system, which is not compatible with the existing congestion control protocols [12].

Different from the existing works, in this chapter, we investigate the scheduling algorithm that jointly considers the queue length and the HOL access delay (which is a different delay measurement compared with the files delay in F-D-MW). We give the analysis of the throughput-optimality of the proposed algorithm with channel variations in the hybrid networks that have both persistent and dynamic flows. We also show that the HOL access delay based scheduling algorithm can be generalized for more flexibility in scheduling algorithm design.

### 5.3 System Model

We consider the systems with two types of flows, persistent flows and dynamic flows, which share the channel resource. Let  $Q^{(p)}$  and  $Q^{(d)}$  denote the set of persistent

flows and dynamic flows, respectively. In each set, there can be multiple classes of flows, which makes the system a heterogeneous one. Within each class, the flows have i.i.d. traffic arrival characteristic and the same channel profile, i.e., with the same channel rate distributions. Denote by  $M$  and  $K$  the number of classes of persistent and dynamic flows, respectively.

### 5.3.1 Arrival Model

Each persistent flow in the system is long-lived and has continuous traffic arrival, so it has infinite amount of data to transmit when  $t \rightarrow \infty$ . Each dynamic flow has finite amount of data to transmit upon its arrival in the system, and leaves the system once its buffer is empty. Over the time, the average number of class- $k$  dynamic flows arriving the system is denoted by  $\lambda_k^{(d)}$ . We focus on the flows that are backlogged in the system which have buffered packets to transmit, and let  $Q_{mj}^{(p)}$  and  $Q_{ki}^{(d)}(t)$  denote the  $j$ -th persistent flow of class- $m$ , and the  $i$ -th dynamic flow of class- $k$  which exists in the system at time slot  $t$ . All the persistent flows arrive during (before) time slot  $t = 0$  and never leave the system. For dynamic flows, since we have the departure of the old flows and the arrival of the new flows, the  $i$ -th flow of class- $k$  may change from one time slot to the other. With each class of the existing dynamic flows, the flow index  $i$  is determined according to its arrival time.

The amount of the remaining bits, i.e., the residual bits, of  $Q_{xy}^{(\cdot)}(t)$  waiting for transmission at the beginning of time slot  $t$  is denoted by  $|Q_{xy}^{(\cdot)}(t)|$ . The number of persistent flows of class- $m$  at the beginning of time slot  $t$  is  $N_m^{(p)}$ , and the number of class- $k$  dynamic flows that are currently backlogged in the system is  $N_k^{(d)}(t)$ . The total numbers of persistent flows and dynamic flows at the beginning of time slot  $t$  are  $N^{(p)} = \sum_{m=1}^M N_m^{(p)}$ , and  $N^{(d)}(t) = \sum_{k=1}^K N_k^{(d)}(t)$ , respectively.

Let the size of one individual packet in persistent flows be  $B_0^{(p)}$ . For a class- $m$  persistent flow  $Q_{mi}^{(p)}$ , the number of arrival packets in one slot is denoted as  $A_{mi}^{(p)}(t)$  with mean  $\lambda_{mi}^{(p)} = \mathbb{E}[A_{mi}^{(p)}(t)]$ , and denote the new data for the individual persistent flow by  $\alpha_{mi}^{(p)}(t) = A_{mi}^{(p)}(t) \cdot B_0^{(p)}$ . The class- $m$  persistent flows have the mean number of arrival packets  $\lambda_m^{(p)} = \sum \lambda_{mi}^{(p)}$ . Let  $\lambda^{(p)} = \sum \lambda_m^{(p)}$  denote the average number of arrival packets from all of the persistent flows.

For dynamic flows, let  $A_k^{(d)}(t) \in \{0\} \cup \mathbb{Z}_+$  denote the number of class- $k$  dynamic flows arriving during time slot  $t$ , which is a random variable.  $A_k^{(d)}(\cdot)$  is i.i.d. with the mean  $\lambda_k^{(d)} = \mathbb{E}[A_k^{(d)}(1)]$ . We suppose that the scheduling decision is made at the

beginning of every time slot, so all the flows that arrive after the beginning of slot  $t$  can only be scheduled at the beginning of slot  $t + 1$ . The initial flow size of  $Q_{ki}^{(d)}(t)$  is denoted as  $B_{ki}^{(d)}(t)$ . In class- $k$ , we assume that  $B_{ki}^{(d)}(t)$  is the i.i.d. copy of some integer random variable  $B_k^{(d)}$  and has a finite mean  $\mathbb{E}[B_k^{(d)}]$ . The second moments of  $A_k^{(\cdot)}(\cdot)$  and  $B_k^{(d)}(\cdot)$  are both finite.

We define  $|Q_m^{(\cdot)}(t)| = \sum_i |Q_{mi}^{(\cdot)}(t)|$  as the class- $m$  backlog. With  $|Q^{(\cdot)}(t)| = \sum_m |Q_m^{(\cdot)}(t)|$ , we have  $|Q(t)| = |Q^{(p)}(t)| + |Q^{(d)}(t)|$  as the system backlog.

### 5.3.2 Channel Model

Let  $r_{mi}^{(\cdot)}(t)$  denote the transmission rate of the wireless channel at time  $t$  between  $Q_{mi}^{(\cdot)}(t)$  and the BS. The unit of the channel rate is bit/slot. The BS can transmit at most  $r_{mi}^{(\cdot)}(t)$  bits at time  $t$  for  $Q_{mi}^{(\cdot)}(t)$ . The rate  $r_{mi}^{(\cdot)}(t)$  may vary over time as a result of fading. For class- $m$  flows, we assume that  $r_{mi}^{(\cdot)}(t)$  are i.i.d. copies of positive integer random variable  $R_m^{(\cdot)}$  with finite supports, i.e.,  $R_m^{(\cdot)} \in \mathcal{R}_m^{(\cdot)} = \{R_{m1}^{(\cdot)}, R_{m2}^{(\cdot)}, \dots\}$ . Let  $|\mathcal{R}_m^{(\cdot)}|$  denote the number of channel rate options in  $\mathcal{R}_m^{(\cdot)}$ . Different classes may have heterogeneous channel condition distributions. The maximum possible transmission rate of the class- $k$  flows is defined as  $R_m^{(\cdot)\max} = \sup\{r : \mathbb{P}\{R_m^{(\cdot)} = r\} > 0\}$ , and the maximum possible transmission rate of the system is defined as  $R^{\max} = \max_m R_m^{(\cdot)\max}$ .

### 5.3.3 System Capacity Region and Throughput-optimality

The capacity region of the systems with a fixed number of persistent flows can be found in [37], which is different from the one with dynamic flows only [52], and thus before the scheduling algorithm is investigated, the capacity region of hybrid systems with both persistent and dynamic flows should be addressed. For the simplicity of the presentation, we make the assumption that there is only one class of flows, i.e.,  $M = 1$ , and  $m$  can be omitted in the notation  $Q_{mj}^{(p)}$  and  $R_{mj}^{(p)}$ , etc., when necessary. The capacity region can be defined in terms of the traffic intensity  $\rho$ , which measures the average occupancy of the shared channel resource. In other words,  $\rho$  is the average number of time slots that are required to transmit the arrival traffic in one time slot when the most efficient transmission strategy is adopted. The sufficient condition for stability to be achievable is  $\rho < 1$  [52]. Thus, if the average amount of arrival traffic in one time slot can be transmitted in less than one time slot by the maximum possible transmission rate, there exists at least one scheduling algorithm to achieve system stability. If  $\rho > 1$ , on average more than one slot is required to transmit the amount

of arrival data in one slot, and the residual data will accumulate into infinity over time which results in instability. From this perspective, the system capacity region is defined as  $\rho < 1$ , and any arrival rate vector in the capacity region should be stably transmitted by throughput-optimal algorithms.

For the hybrid system,  $\rho = \rho^{(p)} + \rho^{(d)}$ , where  $\rho^{(p)}$  and  $\rho^{(d)}$  are the traffic intensities corresponding to the persistent flows and dynamic flows, respectively. We discuss  $\rho^{(p)}$  first. Because  $N^{(p)}(t)$  is a finite constant, the probability that  $P\{r(t) < R^{(p)\max}\}$  cannot be ignored, especially when  $N^{(p)}(t)$  is small. Suppose  $R_1^{(p)} > R_2^{(p)} > \dots > R_{|\mathcal{R}^{(p)}|}^{(p)}$ . For the persistent flows, the probability that at least one flow is in the channel state of  $R_i^{(p)}$  is defined as

$$P[R_i^{(p)}] = 1 - P[r_i^{(p)}(t) \neq R_i^{(p)} | \forall Q_i^{(p)} \in Q^{(p)}].$$

Recall that  $\lambda^{(p)}$  is the total number of arrival packet in one time slot, and  $B_0^{(p)}$  is the packet size. Let  $R_0^{(p)} \neq 0$  with  $P[R_0^{(p)}] = 0$ , the traffic intensity of the persistent flows can be calculated as follows:

$$\rho^{(p)} = \sum_{i=1}^{|\mathcal{R}^{(p)}|} \frac{\max\{0, \lambda^{(p)} B_0^{(p)} - \sum_{j=0}^{i-1} R_j^{(p)} \cdot P[R_j^{(p)}]\}}{R_i^{(p)}}. \quad (5.1)$$

For the dynamic flows, let  $\gamma_k$  represent the expected number of time slots that are required for the service of a class- $k$  flow if served with  $R_k^{(d)\max}$ , and we have  $\gamma_k = \mathbb{E}\left[B_k^{(d)} / R_k^{(d)\max}\right]$ . Let  $\rho_k^{(d)} = \lambda_k^{(d)} \gamma_k$  denote the traffic intensity of class- $k$  dynamic flows, and  $\rho^{(d)} = \sum_{k=1}^K \rho_k^{(d)}$  denote the traffic intensity of the dynamic flows.

For any arrival process that lies in the capacity region, if the system is strongly stable, i.e.,  $\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}|Q(t)| < \infty$ , then the corresponding scheduling algorithm is throughput-optimal.

## 5.4 QHAD Scheduling Algorithm

In this section, we propose a Queue-length associated HOL Access Delay scheduling algorithm (QHAD) for the hybrid systems with persistent and dynamic flows.

### 5.4.1 QHAD

Since delay is involved in the control function of the proposed scheduling algorithm, we give the definition of the HOL access delay  $H_{ki}(t)$  in Definition 2 which we will use in our scheduling algorithm.

**Definition 2** (The HOL Access Delay). *Let  $I_{ki}^{(\cdot)H}(t)$  denote the head bit in  $Q_{ki}^{(\cdot)}(t)$  which will be the first bit in the queue to be transmitted. The HOL access delay of  $Q_{ki}^{(\cdot)}(t)$  is defined as  $H_{ki}^{(\cdot)}(t) = t - t_0$ , where  $t$  is the current time, and  $t_0$  is the time at which  $I_{ki}^{(\cdot)H}(t)$  becomes the first bit in  $Q_{ki}^{(\cdot)}(t)$ .*

For dynamic flows, the HOL access delay can be calculated by

$$H_{ki}^{(\cdot)}(t+1) = \left( H_{ki}^{(\cdot)}(t) + 1 \right) \left( 1 - \mathbb{1}_{\{Q_{ki}^{(\cdot)}(t)\}}(t) \right), \quad (5.2)$$

where  $\mathbb{1}_{\{Q_{ki}^{(\cdot)}(t)\}}(t)$  is the indicator function such that  $\mathbb{1}_{\{Q_{ki}^{(\cdot)}(t)\}}(t) = 1$  only when  $Q_{ki}^{(\cdot)}(t)$  is scheduled at time slot  $t$ . For each persistent flow, when the queue length is not zero, the HOL access delay is also calculated by (5.2), while when the queue length is zero, which means the buffer is empty, the HOL access delay is always zero.

With the above definition, the scheduling rule of QHAD can be found in Algorithm 4.

**Algorithm 4.** *The Queue-length associated HOL Access Delay scheduling algorithm (QHAD) seeks the user  $\{k, i\}$  to transmit that satisfies the following condition at the beginning of time slot  $t$ :*

$$\begin{aligned} \{k, i\}^* (Q_{ki}^{(\cdot)}(t), H_{ki}^{(\cdot)}(t), r_{ki}^{(\cdot)}(t)) \in \\ \arg \max_{k,i} \left( |Q_{ki}^{(\cdot)}(t)| + H_{ki}^{(\cdot)}(t) \right) \cdot r_{ki}^{(\cdot)}(t), \end{aligned} \quad (5.3)$$

*with uniform tie-breaking if there are more than one user satisfying the condition. The scheduling decision is made in every time slot independently.*

### 5.4.2 Stability with QHAD

The system stability with QHAD can be concluded by several steps. For simplicity of presentation, in this subsection we assume that there is one class of flow in each type of flows, i.e.,  $M = 1$  and  $K = 1$ , and thus  $k$  in the notation  $Q_{ki}^{(\cdot)}(t)$  is omitted.

The results can be easily extended for multi-class scenarios, which are omitted due to space limit. First, we study some observations of an unstable system.

**Lemma 2.** *With QHAD, when  $t \rightarrow \infty$ , if  $\exists i \in N^{(p)}(t)$  such that  $|Q_i^{(p)}(t)| \rightarrow \infty$ , we have  $\forall j \in N^{(p)}(t)$ ,  $|Q_j^{(p)}(t)| \rightarrow \infty$ , and  $N^{(d)}(t) \rightarrow \infty$ .*

*Proof.* We first show that  $\forall j \in N^{(p)}(t)$  we have  $|Q_j^{(p)}(t)| \rightarrow \infty$ . Without loss of generality, suppose  $|Q_1^{(p)}(t)| \rightarrow \infty$  at time  $t$ , and let  $\gamma_1^{(p)}(t)$  be the number of time slots required to transmit the buffered data in  $Q_1^{(p)}(t)$ . Obviously  $\gamma_1^{(p)}(t) \rightarrow \infty$ . Define  $\mathbf{I}$  as the set of all the flows with infinite queue length. Since  $H_i^{(p)}(t)$  represents the waiting time since the last transmission of  $Q_i^{(p)}(t)$ ,  $H_i^{(p)}(t)$  is finite if  $|Q_i^{(p)}(t)|$  is finite at time  $t$  considering that persistent flows have continuous traffic arrival. According to QHAD,  $Q_i^{(p)}(t)$  from  $\mathbf{I}$  will get scheduled for  $t \rightarrow \infty$  (considering  $\gamma_1^{(p)}(t) \rightarrow \infty$ ) before any scheduled transmission of the flows that do not belong to  $\mathbf{I}$ , no matter whether  $N^{(d)}(t) \rightarrow \infty$  or not. During  $[t, t + \gamma_1^{(p)}(t))$ , since the persistent flows that do not belong to  $\mathbf{I}$  have continuous traffic arrival, their queue lengths will also increase into  $\infty$ .

Next we explain  $N^{(d)}(t) \rightarrow \infty$ . Firstly, we demonstrate  $\exists j \in N^{(d)}(t) : H_j^{(d)}(t) \rightarrow \infty$ . Assume  $\max H_j^{(d)}(t) < \infty$ . Without loss of generality, suppose  $|Q_1^{(p)}(t)| \rightarrow \infty$ , and thus we have  $\forall j \in N^{(d)}(t)$ ,  $(H_j^{(d)}(t) + Q_j^{(d)}(t))r_j^{(d)}(t) < |Q_1^{(p)}(t)|r_1^{(p)}(t)$  in  $[t, t + \gamma_1^{(p)}(t))$ . As a result, the number of time slots between the previous and next transmission of  $Q_j^{(d)}(t)$  is infinite since QHAD prioritize  $Q_1^{(p)}(t)$ . This leads to contradiction to the assumption that  $\max H_j^{(d)}(t) < \infty$ . Thus we have  $\exists j \in N^{(d)}(t) : H_j^{(d)}(t) \rightarrow \infty$ . Based on this result, next we prove  $\forall j \in N^{(d)}(t) : H_j^{(d)}(t) \rightarrow \infty$ .

Assume that there are only a finite number of dynamic flows in the system that have infinite  $H_j^{(d)}(t)$ , among which  $H_1^{(d)}(t) \rightarrow \infty$ , and thus for  $Q_1^{(d)}(t)$ , the number of time slots between its two continues transmissions is infinite. With the above assumption, there are dynamic flows with finite HOL access delay in the system, and without loss of generality we assume  $H_2^{(d)}(t) < \infty$ , which means the number of time slots between two continues transmissions of  $Q_2^{(d)}(t)$  is finite. However, according to the scheduling rule of QHAD, since we always have  $(H_1^{(d)}(t) + \epsilon)r_1^{(d)}(t) > (H_2^{(d)}(t) + |Q_2^{(d)}(t)|)r_2^{(d)}(t)$ , the number of time slots for  $Q_1^{(d)}(t)$  to wait between two transmissions is always less than that for  $Q_2^{(d)}(t)$ , which leads to contradiction, and thus our assumption that only a finite number of dynamic flows in the system that have infinite  $H_j^{(d)}(t)$  is not true. This concludes the result of  $N^{(d)}(t) \rightarrow \infty$ .  $\square$

Lemma 2 infers that if one persistent flow is not stabilized, all the other flows

in the system, either persistent or dynamic, cannot be stabilized. Next we show the other observation about instability.

**Lemma 3.** *With QHAD, when  $t \rightarrow \infty$ , if  $N^{(d)}(t) \rightarrow \infty$ , we have  $\forall j \in N^{(p)}(t)$ ,  $|Q_j^{(p)}(t)| \rightarrow \infty$ .*

*Proof.* For simplicity we assume that  $M = 1$  and  $K = 1$ , and thus  $k$  in the notation  $Q_{ki}^{(\cdot)}(t)$  is omitted. When  $N^{(d)}(t) \rightarrow \infty$ , the dynamic flows are not stabilized and there are infinite number of dynamic flows with  $H_i^{(d)}(t) \rightarrow \infty$ . let  $\gamma^{(d)}(t)$  be the number of time slots required to transmit the buffered dynamic flows, and obviously  $\gamma^{(d)}(t) \rightarrow \infty$ .

Suppose for any persistent flow  $Q_j^{(p)}(t)$ , we have  $|Q_j^{(p)}(t)| < \infty$ , which means the persistent flows are stabilized. Thus  $\forall j \in N^{(p)}$ ,  $H_j^{(p)}(t) < \infty$ , and for the dynamic flows with  $H_i^{(d)}(t) \rightarrow \infty$ ,  $(H_i^{(d)}(t) + \epsilon_Q)r_i^{(d)}(t) \gg (|Q_j^{(p)}(t)| + H_j^{(p)}(t))r_i^{(p)}(t)$ . According to the scheduling rule of QHAD, the dynamic flows will dominate the transmission and no persistent flows would be scheduled during  $[t, t + \gamma^{(d)}(t))$ , and accordingly  $H_j^{(p)}(t) \rightarrow \infty$ . This leads to contradiction with  $H_j^{(p)}(t) < \infty$ , which means that the assumption that the persistent flows are stabilized is not true, and thus  $\exists i \in N^{(p)}(t)$  such that  $|Q_i^{(p)}(t)| \rightarrow \infty$ . From Lemma 2 we have  $\forall j \in N^{(p)}(t)$ ,  $|Q_j^{(p)}(t)| \rightarrow \infty$ .  $\square$

Lemma 3 essentially infers that if the dynamic flows are not stabilized, the persistent flows are not stabilized either. Without prove, we give the lemma below since it is the contraposition of Lemma 3.

**Lemma 4.** *With QHAD, when  $t \rightarrow \infty$ , if  $\forall j \in N^{(p)}(t)$ ,  $|Q_j^{(p)}(t)| < \infty$ , we have  $N^{(d)}(t) < \infty$ .*

Next we study the scheduling of the persistent flows with QHAD in the system with the definition of the following Lyapunov function:

$$L(t) = \sum_i |Q_i^{(p)}(t)|^2 + c \cdot H_i^{(p)}(t), \quad (5.4)$$

where  $c$  is a constant. Let  $\Delta L$  denote the Lyapunov drift, we have

$$\begin{aligned}
\Delta L &= L(t+1) - L(t) \\
&= \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t+1)|^2 + c \sum_{i=1}^{N^{(p)}} H_i^{(p)}(t+1) \\
&\quad - \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)|^2 - c \sum_{i=1}^{N^{(p)}} H_i^{(p)}(t).
\end{aligned} \tag{5.5}$$

Given the queue evolution as

$$|Q_i^{(p)}(t+1)| = |Q_i^{(p)}(t)| + \alpha_i^{(p)}(t) - r_i^{(p)}(t),$$

(5.5) becomes

$$\begin{aligned}
\Delta L &= \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)|^2 + \alpha_i^{(p)}(t) - r_i^{(p)}(t)^2 - \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)|^2 \\
&\quad + c \sum_{i=1}^{N^{(p)}} H_i^{(p)}(t+1) - c \sum_{i=1}^{N^{(p)}} H_i^{(p)}(t).
\end{aligned} \tag{5.6}$$

Let  $S^{QHAD}(t)$  be the set of scheduled flows at time  $t$  by QHAD (if we schedule only one flow every time slot, we have only one flow in  $S^{QHAD}(t)$ ). Since

$$\begin{aligned}
&\sum_{i=1}^{N^{(p)}} \left( |Q_i^{(p)}(t)| + \alpha_i^{(p)}(t) - r_i^{(p)}(t) \right)^2 \\
&= \sum_{i=1}^{N^{(p)}} \left[ |Q_i^{(p)}(t)|^2 + (\alpha_i^{(p)}(t) - r_i^{(p)}(t))^2 \right] \\
&\quad + \sum_{i=1}^{N^{(p)}} 2|Q_i^{(p)}(t)|(\alpha_i^{(p)}(t) - r_i^{(p)}(t)) \\
&\leq \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)|^2 + \sum_{i=1}^{N^{(p)}} \left[ (\alpha_i^{(p)}(t))^2 + (r_i^{(p)}(t))^2 \right] \\
&\quad + 2 \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)|\alpha_i^{(p)}(t) - 2 \sum_{i \in S^{QHAD}(t)} |Q_i^{(p)}(t)|r_i^{(p)}(t),
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=1}^{N^{(p)}} H_i^{(p)}(t+1) &\leq \sum_{i=1}^{N^{(p)}} [H_i^{(p)}(t) + 1] - \sum_{i \in SQHAD(t)} H_i^{(p)}(t) \\
&= \sum_{i=1}^{N^{(p)}} H_i^{(p)}(t) + N^{(p)}(t) - \sum_{i \in SQHAD(t)} H_i^{(p)}(t),
\end{aligned} \tag{5.7}$$

from (5.6), we have

$$\begin{aligned}
\Delta L &\leq \sum_{i=1}^{N^{(p)}} \left[ (\alpha_i^{(p)}(t))^2 + (r_i^{(p)}(t))^2 \right] + 2 \sum_{i=1}^{N^{(p)}} |Q_i^{(p)}(t)| \alpha_i^{(p)}(t) \\
&\quad - \sum_{i \in SQHAD(t)} \left( 2|Q_i^{(p)}(t)| r_i^{(p)}(t) + c H_i^{(p)}(t) \right) + c \cdot N^{(p)}(t).
\end{aligned} \tag{5.8}$$

To find the upper bound of the right hand side (RHS) of (5.8), we next look for a lower bound of  $\sum Q_i^{(p)}(t) r_i^{(p)}(t)$  by studying the relationship between QHAD and MR.

Denote by  $S^{MR}$  the set of selected flows with the MR scheduler. Since each flow has  $H_i^{(p)}(t) \geq 0$ , and considering (5.3), we have

$$\begin{aligned}
&\sum_{i \in SQHAD(t)} \left( Q_i^{(p)}(t) + H_i^{(p)}(t) \right) r_i^{(p)}(t) \\
&\geq \sum_{j \in S^{MR}(t)} \left( Q_j^{(p)}(t) + H_j^{(p)}(t) \right) r_j^{(p)}(t) \\
&\geq \sum_{j \in S^{MR}(t)} Q_j^{(p)}(t) r_j^{(p)}(t),
\end{aligned}$$

which indicates

$$\sum_{i \in SQHAD(t)} Q_i^{(p)}(t) r_i^{(p)}(t) \geq \sum_{j \in S^{MR}(t)} Q_j^{(p)}(t) r_j^{(p)}(t) - \sum_{i \in SQHAD(t)} H_i^{(p)}(t) r_i^{(p)}(t),$$

and thus (5.8) becomes

$$\begin{aligned}
\Delta L &\leq \sum_{i=1}^{N^{(p)}} \left[ (\alpha_i^{(p)}(t))^2 + (r_i^{(p)}(t))^2 \right] + 2 \sum_{i=1}^{N^{(p)}} Q_i^{(p)}(t) \alpha_i^{(p)}(t) \\
&\quad - 2 \sum_{j \in S^{MR}(t)} Q_j^{(p)}(t) r_j^{(p)}(t) + 2 \sum_{i \in S^{QHAD}(t)} H_i^{(p)}(t) r_i^{(p)}(t) \\
&\quad - c \sum_{i \in S^{QHAD}(t)} H_i^{(p)}(t) + c \cdot N^{(p)}(t). \tag{5.9}
\end{aligned}$$

Since MR selects the flow with its best channel condition, for the traffic arrival rates that lie within the capacity region, we have  $\lambda^{(p)} B_o^{(p)} + \epsilon \leq \mathbb{E} \left[ \sum_{j \in S^{MR}(t)} r_j^{(p)}(t) \right]$ . By taking expectation on both sides of (5.9), we have

$$\mathbb{E}[\Delta L] \leq C_1 - \epsilon_1 \sum_{i=1}^{N^{(p)}} \mathbb{E} \left[ |Q_i^{(p)}(t)| \right] + \sum_{i \in S^{QHAD}(t)} \mathbb{E} \left[ C_2 H_i^{(p)}(t) \right], \tag{5.10}$$

where we have the constant

$$C_1 = \mathbb{E} \left[ \sum_{i=1}^{N^{(p)}} \left( (\alpha_i^{(p)}(t))^2 + (r_i^{(p)}(t))^2 \right) + c \cdot N^{(p)}(t) \right],$$

and  $C_2 = 2r_i^{(p)}(t) - c$ . By choosing any  $c$  that is sufficiently large to make  $\mathbb{E} \left[ C_2 H_i^{(p)}(t) \right] \leq 0$ , (5.10) becomes

$$\mathbb{E}[\Delta L] \leq C_1 - \epsilon_1 \sum_{i=1}^{N^{(p)}} \mathbb{E} \left[ |Q_i^{(p)}(t)| \right], \tag{5.11}$$

which indicates that the persistent flows can be stabilized by QHAD [37], i.e., in any time slot,  $\forall j \in N^{(p)}(t)$ ,  $|Q_j^{(p)}(t)| < \infty$ . With this result and Lemma 4, we come to the conclusion that the system can be stabilized by QHAD.

## 5.5 HADGe Scheduling Algorithm

In QHAD, the scheduler uses a linear combination of the queue length and the HOL access delay of each flow as the control objective function. To provide better flexibility and adaptiveness of the scheduling algorithms so that it is possible for the network to make a tradeoff between different performance metrics, it is important to generalize

the form of the control objective function. Since the queue length based scheduling algorithms has been extensively investigated in the literature for the systems with persistent flows, we hereby only study the generalization of the HOL access delay based scheduling algorithms (HAD) [9] for the systems with dynamic flows. With the system model and the definition of HOL access delay, we adopt the following generalized HOL access delay based scheduling algorithm.

**Algorithm 5** (The HADGe Scheduling Algorithm). *The Generalized HOL Access Delay based scheduling algorithm (HADGe) seeks the user  $\{k, i\}$  ( $1 \leq k \leq K, 1 \leq i \leq N_k(t)$ ) to transmit that satisfies the following condition at the beginning of time slot  $t$ :*

$$\{k, i\}^*(H_{ki}(t), r_{ki}(t)) \in \arg \max_{Q_{ki}(t) \in \mathcal{N}(t)} f(H_{ki}(t)) \cdot r_{ki}(t), \quad (5.12)$$

*with uniform tie-breaking if there are more than one user satisfying the condition. The scheduling decision is made in every time slot independently.*

$\mathcal{N}(t)$  is the set of flows in the system at time  $t$ . The function  $f(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is a concave and non-decreasing one with  $f(0) = 0$  and  $\lim_{x \rightarrow \infty} f(x) = \infty$ . For any  $\alpha \geq 1$  and  $x > 0$ , there exists a number  $c(\alpha) > 0$  such that  $f(\alpha x) \leq c(\alpha) \cdot f(x)$ . Given an  $f(\cdot)$ , we assume that  $\exists x > 0$  such that

$$f'(x)/f(x) \geq R_k^{\max}/\mathbb{E}[r_{ki}(t)] - 1.$$

Since the algorithm we investigate in this works is delay-based, we decide to use the Lyapunov function in (5.13).

$$L(Q_i(t), H_i(t)) = \sum_i |Q_i(t)| \cdot f(H_i(t)). \quad (5.13)$$

In the derivative, we need to deal with the indicator function  $\mathbf{1}(t)$ . Depending on the scheduling rule, if one flow is scheduled in time slot  $t$ , the corresponding indicator function equals 1 at the same time, otherwise it all equals 0.

With the HOL access delay based scheduling algorithm, the evolution of the HOL access delay  $H_i(t)$  is shown as

$$H_i(t+1) = (H_i(t) + 1)(1 - \mathbf{1}_{\{Q_i(t) \text{ is scheduled at time } t\}}). \quad (5.14)$$

To deal with the indicator in (5.14), we can convert it into the form of

$$\mathbb{1}_{\{Q_i(t) \text{ is scheduled at time } t\}} = (Q_i(t) - Q_i(t+1))/r_i(t), \quad (5.15)$$

and (5.14) becomes

$$H_i(t+1) = (H_i(t) + 1)(1 - (Q_i(t) - Q_i(t+1))/r_i(t)). \quad (5.16)$$

Next we discuss the system stability with HADGe. The Lyapunov function considered is (5.13). The Lyapunov function of HADGe for time slot  $t+1$  is

$$L(Q_i(t+1), H_i(t+1)) = \sum_{i \in \mathcal{N}(t+1)} |Q_i(t+1)| \cdot f(H_i(t+1)). \quad (5.17)$$

We can divide  $Q_i(t+1)$  into two parts: the old flows that already exist before time slot  $t$ , and the new flows that just arrive at time slot  $t$ , and hence (5.17) becomes

$$\begin{aligned} L(Q_i(t+1), H_i(t+1)) &= \\ &= \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f(H_i(t+1)) + f(1) \cdot \sum_{j=1}^{A(t)} B_j(t), \end{aligned} \quad (5.18)$$

where  $A(t)$  represents the number of new flows at the system in time slot  $t$ , and  $B_j(t)$  is the initial queue length of the  $j$ -th new flow of  $A(t)$ . The first term on the RHS is for the old flows that come before  $t$ , while the second term on the RHS is for the new flows. Next we focus on the first term. Considering (5.2) and (5.16), the first term of the RHS becomes

$$\begin{aligned} & \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f(H_i(t+1)) \\ &= \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f\left((H_i(t) + 1) \left(1 - \mathbb{1}_{\{Q_i(t)\}}(t)\right)\right) \\ &= \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f\left(\left(H_i(t) + 1\right) \left(1 - \frac{Q_i(t) - Q_i(t+1)}{r_i(t)}\right)\right). \end{aligned}$$

Since  $f(\cdot)$  is a non-decreasing function, and for dynamic flows we always have  $Q_i(t) -$

$Q_i(t+1) \geq 0$  no matter whether  $Q_i(t)$  is scheduled or not, the above equation becomes

$$\begin{aligned}
& \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f(H_i(t+1)) \\
&= \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f\left(\frac{1}{r_i(t)} (H_i(t)+1) (r_i(t) - Q_i(t) + Q_i(t+1))\right) \\
&= \sum_{i \in \mathcal{N}(t)} (|Q_i(t)| - r_i(t))^+ \cdot f\left(\frac{1}{r_i(t)} (H_i(t)+1) (r_i(t) - Q_i(t) + Q_i(t+1))\right) \\
&\leq \sum_{i \in \mathcal{N}(t)} (|Q_i(t)| - r_i(t))^+ \cdot f\left(\frac{1}{r_i(t)} (H_i(t)+1) r_i(t)\right) \\
&= \sum_{i \in \mathcal{N}(t)} (|Q_i(t)| - r_i(t))^+ \cdot f(H_i(t)+1).
\end{aligned}$$

If  $|Q_i(t)| - r_i(t) < 0$ , we can safely remove the corresponding item from the summation over  $\mathcal{N}(t)$  since  $(|Q_i(t)| - r_i(t))^+ = 0$ , and only consider  $Q_j(t)$  such that  $|Q_i(t)| \geq r_i(t)$ . Suppose  $Q_i(t) \in \tilde{\mathcal{N}}(t)$  if  $|Q_i(t)| > r_i(t)$ , where  $i = 1, 2, \dots, \tilde{K}$ , and let  $x_i$  be some value between  $H(t)$  and  $H(t+1)$ , with the Mean Value Theorem, we have

$$\begin{aligned}
& \sum_{i \in \mathcal{N}(t)} |Q_i(t+1)| \cdot f(H_i(t+1)) \\
&\leq \sum_{i \in \tilde{\mathcal{N}}(t)} (|Q_i(t)| - r_i(t)) (f(H_i(t)) + 1) \\
&= \sum_{i \in \tilde{\mathcal{N}}(t)} (|Q_i(t)| - r_i(t)) (f(H_i(t)) + f'(x_i)) \\
&\leq \sum_{i \in \tilde{\mathcal{N}}(t)} |Q_i(t)| [f(H_i(t)) + f'(x_i)] - \\
&\quad r_i(t) [f(H_i(t)) + f'(x_i)] \\
&\leq \sum_{i \in \tilde{\mathcal{N}}(t)} |Q_i(t)| [f(H_i(t)) + f'(H_i(t))] - \\
&\quad r_i(t) [f(H_i(t)) + f'(H_i(t))].
\end{aligned} \tag{5.19}$$

Based on (5.17), the corresponding drift is defined as

$$\Delta L(t) = \mathbb{E}[L(Q_i(t+1), H_i(t+1)) - L(Q_i(t), H_i(t)) | Q_i(t), H_i(t)]. \tag{5.20}$$

Combining (5.18) and (5.19), we have

$$\begin{aligned} \Delta L(t) &\leq \mathbb{E}\left[\sum_{i \in \mathcal{N}(t)} |Q_i(t)| f'(H_i(t))\right] \\ &\quad - \mathbb{E}\left[\sum_{i \in \mathcal{N}(t)} r_i^{\max} f(H_i(t))\right] + f(1)\rho. \end{aligned} \quad (5.21)$$

Notice that when we changed  $i \in \tilde{\mathcal{N}}(t)$  in (5.19) into  $\mathcal{N}(t)$  in (5.20), the result here is still true because for all flows in  $\tilde{\mathcal{N}}(t)$  but not in  $\mathcal{N}(t)$ , the corresponding queue lengths and HOL access delays in the next time slot will be all equal to zeros, which do not affect the result.

Next we provide an upper bound of the first term on the right hand side of (5.21) by considering the expectation conditioned on  $H^{\max}(t) = \max_{i \in \mathcal{N}} H_i(t)$ . Given  $Q_i(t) - r_i(t) \leq B_0$ , with Lemma 1 in [28], we have

$$\begin{aligned} &\mathbb{E}\left[\sum_{i \in \mathcal{N}(t)} |Q_i(t)| f'(H_i(t)) | H_i^{\max}(t)\right] \\ &\leq \mathbb{E}\left[\sum_{\tau=t-H^{\max}(t)+1}^{t-1} f'(t-\tau) \sum_i (Q_i(t-\tau) - r_i(t-\tau)) | H^{\max}(t)\right] + \lambda^{\max} B_0 f'(1) \\ &\leq \alpha \sum_{\tau=t-H^{\max}(t)+1}^{t-1} f'(t-\tau) + \lambda^{\max} B_0 f'(1) \\ &= \alpha \mathbb{E}\left[\sum_{\tau=1}^{H^{\max}(t)-1} f'(t-\tau)\right] + \lambda^{\max} B_0 f'(1) \\ &\leq \alpha \mathbb{E}[f(H^{\max}(t))] + (\alpha + \lambda^{\max} B_0) f'(1) \end{aligned} \quad (5.22)$$

Define  $B_1 = (\alpha + \lambda^{\max} B_0) f'(1) + f(1)\alpha$ , considering that only the flow with the highest delay given certain channel rate in each time slot can be scheduled in HADGe, we have

$$\Delta L(t) \leq \alpha \cdot \mathbb{E}[f(H^{\max}(t))] + B_1 - r^{\max} \mathbb{E}[f(H_i^{\max}(t))] \quad (5.23)$$

by combining (5.22) and (5.21). Since we consider that the traffic arrival rate is within the system capacity region, from (5.23) we can conclude

$$\Delta L(t) \leq -\epsilon \mathbb{E}[f(H^{\max}(t))] + B_1, \quad (5.24)$$

which indicates that the system can be stabilized by the proposed scheduling algorithm with any traffic arrival rate in the system capacity.

The design of HADGe offers a direction to generalize QHAD for hybrid systems. HADGe is throughput-optimal in flow-level dynamic systems allowing certain level of channel variations. Notice that the function  $g(x) = x$  belongs to the set of concave functions. (5.7), one of the key steps in the proof of QHAD's throughput-optimality regarding the HOL access delay, still holds if we replace  $g(H_i^{(\cdot)}(t)) = H_i^{(\cdot)}(t)$  by another concave function  $f(H_i^{(\cdot)}(t))$ . Lemmas 2-4 are also true regarding the instability observations of  $f(H_i^{(\cdot)}(t))$  instead of  $H_i^{(\cdot)}(t)$ . Thus we believe that a promising approach to generalize QHAD is to replace  $H_i^{(\cdot)}(t)$  in (5.3) by  $f(H_i^{(\cdot)}(t))$ . We leave this topic to one of our future works.

## 5.6 Performance and Discussion

In this section, we evaluate the performance of HADGe and QHAD, along with the other throughput-optimal scheduling algorithms, including the Queue-length based MaxWeight (QMW) [50], the Flow-Delay based MaxWeight (F-D-MW) [46], the Max Rate (MR) scheduling algorithms [33].

### 5.6.1 QHAD

The throughput-optimality of QHAD has been validated by simulations in Matlab. We use a two-state channel model for both persistent and dynamic flows and the channel rate set is  $\{10Mbps, 8Mbps\}$ . The arrival traffic rate of the system is calculated according to the traffic intensity in each simulation. With  $\rho^{(p)} = \{0.3, 0.5, 0.8\}$  and the x-axis as the system running time up to  $5 \times 10^4$  time slot which is long enough to reflect if a scheduler is throughput-optimal or not, Figs. 5.1-5.3 show the system backlog, and Figs. 5.4-5.6 show the total number of flows in the system, in which there are both the persistent and dynamic flows. The traffic intensity for all the figures are 0.99, and the number of persistent flow is set to be one. The performances of QMW and MR are shown in the simulation results for comparison.

In Fig. 5.1 with  $\rho^{(p)} = 0.3$ , HADGe can suppress the growing of the system backlog for all the time. The curve for HADGe-P shows the amount of the data in the persistent flow, which is also bounded over time. MR performs almost identical with HADGe, which can maintain the queue stability when  $\rho^{(p)} = 0.3$ . QMW fails to

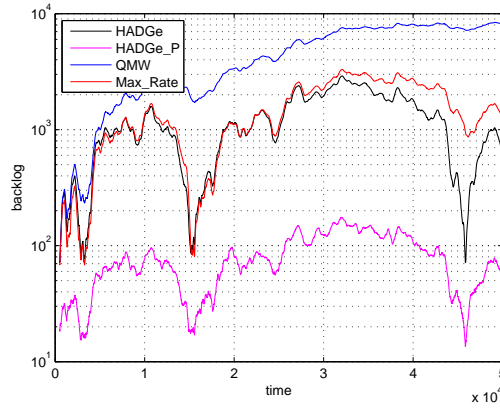


Figure 5.1: System backlog, persistent flow workload 30%.

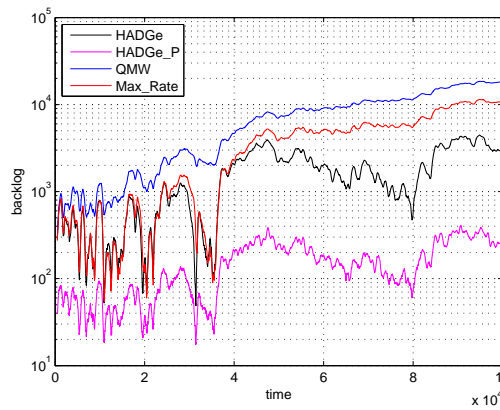


Figure 5.2: System backlog, persistent flow workload 50%.

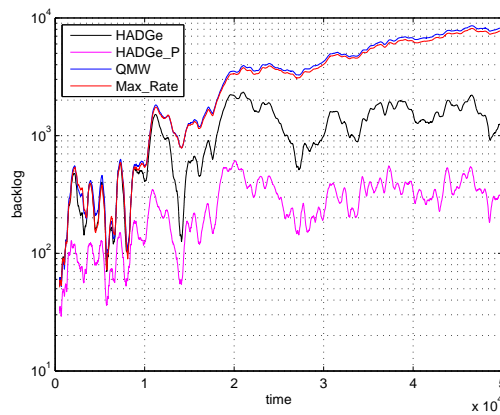


Figure 5.3: System backlog, persistent flow workload 80%.

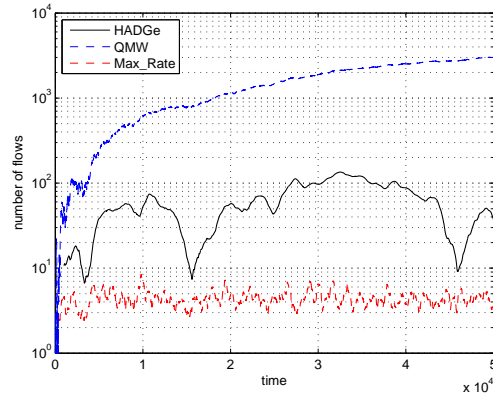


Figure 5.4: Number of flows, persistent flow workload 30%.

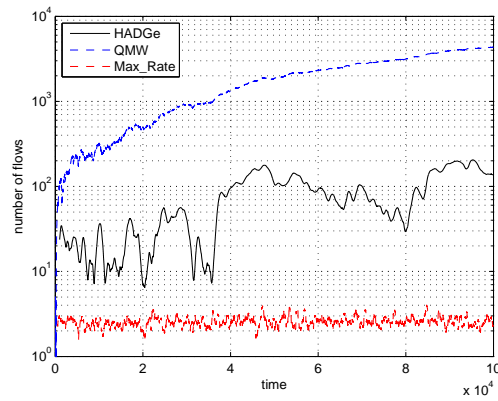


Figure 5.5: Number of flows, persistent flow workload 50%.

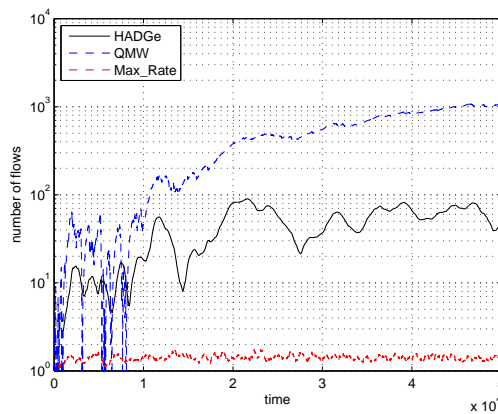


Figure 5.6: Number of flows, persistent flow workload 80%.

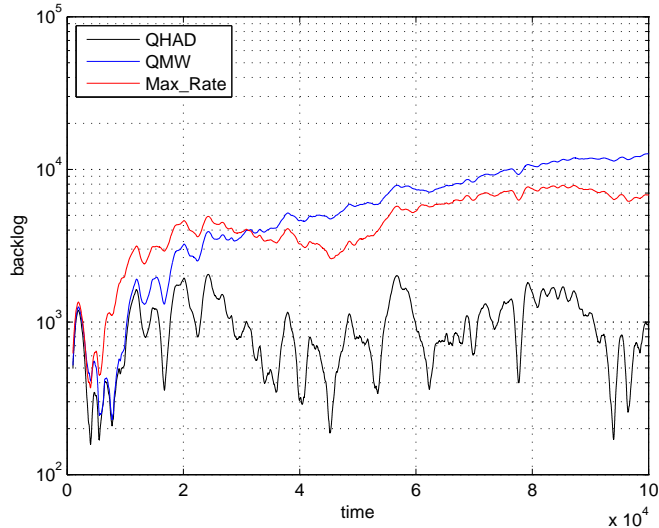


Figure 5.7: System backlog, persistent flow workload 95%.

achieve the queue stability due to the dynamic flows.

Increasing  $\rho^{(p)}$  to 0.5, the result is shown in Fig. 5.2. As expected, HADGe always keeps the total amount of system backlog bounded over time demonstrating throughput-optimality. The backlog of the persistent flow is also stabilized by HADGe. On contrast, the total backlog with QMW cannot be stabilized. One interesting observation about MR in this simulation is that MR shows instability since the backlog of MR has the trend of growing into infinity over time just like that of QMW. The reason is that the dynamic flows in MR share too much of the channel time, and thus the persistent flow has insufficient share of the channel time to transmit the arrival traffic, i.e., for MR the channel allocation is less than  $\rho^{(p)}$ ; meanwhile, the redundant resources for  $Q^{(d)}$  keeps the number of dynamics flows in a very low level, and thus the total number of flows, including both the persistent and dynamic flows, is also limited to very low. As a result, The probability of always scheduling a flow in the maximum possible transmission rate reduces dramatically, and hence the channel is not efficiently utilized without fully exploring the multiuser diversity gain.

The above analysis for the instability of MR can be verified through Figs. 5.4-5.5. In Fig. 5.4 with  $\rho^{(p)} = 0.3$ , MR has the least  $N(t)$ . QHAD has much higher  $N(t)$  than MR, but it is well bounded.  $N(t)$  of QMW cannot be stabilized. In Fig. 5.5 with  $\rho^{(p)} = 0.5$ , since the workload from the dynamic flows is reduced,  $N(t)$  of MR is further reduced. However, a fewer number of flows means less chance to transmit in  $R^{\max}$ . As a result, MR shows instability in Fig. 5.2.  $N(t)$  of QHAD is maintained

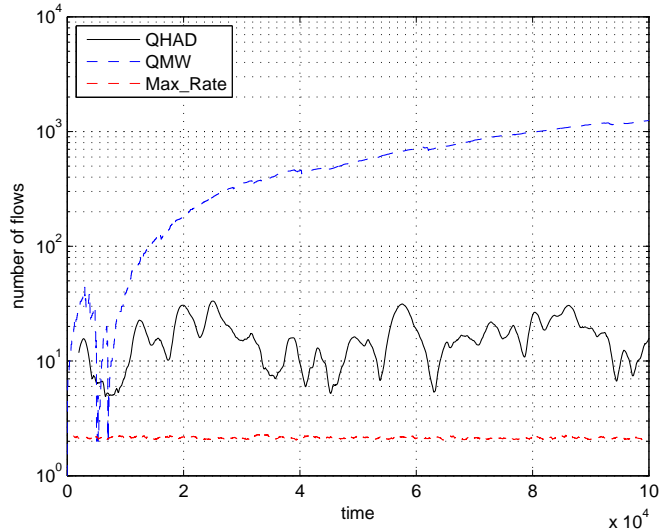


Figure 5.8: Number of flows, persistent flow workload 95%.

on the same level as that in Fig. 5.4, which gives the scheduler plenty of possibility to choose the flows in  $R^{\max}$  channel state.

Increasing  $\rho^{(p)}$  to 0.8, the system backlog is shown in Fig. 5.3. We can observe the same trend comparing with that in Fig. 5.3: only QHAD is able to stabilize the system's backlog, while QMW and MR is not throughput-optimal. The number of flows with  $\rho^{(p)} = 0.8$  is shown in Fig. 5.6. We can observe again that  $N(t)$  of MR is too small to use the multiuser diversity gain, and  $N(t)$  of QMW cannot be stabilized. Only  $N(t)$  of QHAD is stabilized in a proper level so that the multiuser diversity gain is fully used, and the throughput-optimality is achieved.

Increasing the number of persistent flows in the system to two and setting  $\rho^{(p)} = 0.95$ ,  $\rho = 0.99$ , the system backlog and the number of flows are shown in Fig. 5.7 and Fig. 5.8, respectively. The simulation results further validate our conclusion that QHAD is able to stabilize the hybrid system with both persistent and dynamic flows. From all the performance evaluation, we make the following conclusions for the schedulers in the hybrid systems. First, QHAD is verified to be throughput-optimal. Second, when the system has only dynamic flows, MR is throughput-optimal. Third, when the system has both persistent and dynamic flows, MR is not throughput-optimal.

### 5.6.2 HADGe

The throughput-optimality of HADGe in the systems with dynamic flows is validated in Figs. 5.9-5.11, with three concave functions of the HOL access delay, i.e.,  $f_1(H_i(t)) = H_i(t)^{1/2}$ ,  $f_2(H_i(t)) = \log(H_i(t) + 1)$ ,  $f_3(H_i(t)) = \log \log(H_i(t) + e)$ . The x-axis is the system running time, and we use the number of flows in the y-axis as the metric of throughput-optimality. The arrival traffic intensity is 0.99, which is very close to 1 and can easily result in instability if the adopted scheduler is not throughput-optimal. The performances of MR, F-D-MW and QMW are also presented.

In Fig. 5.9, HADGe in the form of  $f_1$  performs almost identical to MR and F-D-MW which are provably throughput-optimal in the literature. Thus we verified that HADGe with the concave function  $f_1$  is throughput-optimal. The same conclusion can be drawn from Fig. 5.10 where  $f_2$  is adopted in HADGe. In Fig. 5.11, we observed that there is a noticeable gap between MR and HADGe with  $f_3$ . But we also had the observation that the average  $N(t)$  of HADGe is bounded and never increases into infinity over time, which fits the definition of throughput-optimality. In order to double verify this, we showed the performance of F-D-MW in the form of  $f_3$  as well, which has been proved to be throughput-optimal in [28, 46]. There is only a marginal gap between HADGe in  $f_3$  and F-D-MW in  $f_3$ , and their trajectory is also identical, which validates our result. Since QMW is not throughput-optimal for the system with dynamic flows,  $N(t)$  of QMW has the trend of growing into infinity in all the three figures here.

## 5.7 Conclusion

In this chapter, we have investigated the scheduling algorithm design for the coexistence of the persistent and dynamic flows. We proposed QHAD scheduling algorithm which jointly considers the queue length and the HOL access delay of each individual flow, and showed that QHAD is throughput-optimal for the hybrid systems with the presence of both persistent and dynamic flows. Furthermore, we studied the generalization of the HAD scheduling to HADGe scheduling in order to provide potential flexibilities in scheduling algorithm designs. We have shown that this generalized form of scheduling algorithm is throughput-optimal.

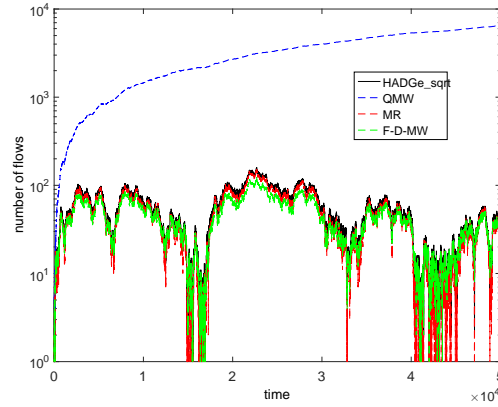


Figure 5.9: Number of flows for  $f_1(H_i(t)) = H_i(t)^{1/2}$ .

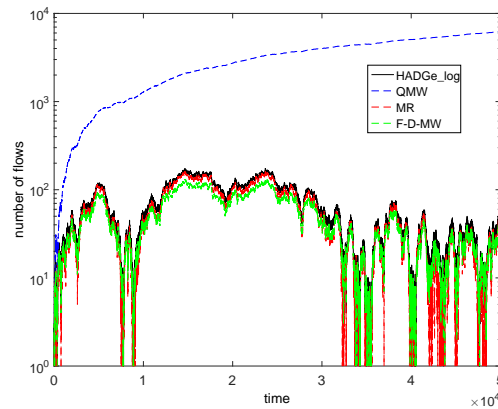


Figure 5.10: Number of flows for  $f_2(H_i(t)) = \log(H_i(t) + 1)$ .

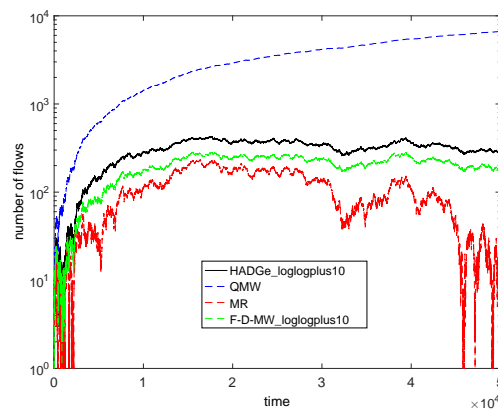


Figure 5.11: Number of flows for  $f_3(H_i(t)) = \log \log(H_i(t) + e)$ .

## Chapter 6

# Conclusions and Future Research Issues

### 6.1 Conclusions

In this dissertation, we have discussed various aspects of scheduling and resource allocation in multiuser wireless systems.

1. We have discussed the condition for the queue stability of flow-level dynamic wireless networks, and proposed HAD scheduler using HOL access delay as the weight of channel rate to stabilize dynamic flows. Its throughput-optimality has been proved by analysis and validated in simulation in heterogeneous wireless networks.
2. The incompatibility between the existing throughput-optimal solutions and the TCP protocol has been revealed, and the reason of unfairness problem has been explained. We have proposed to adopt HAD scheduler for TCP regulated flows, and demonstrated that HAD can support TCP flows better than the existing solutions under various channel conditions.
3. The analytic model to analyze the queueing behavior of the HAD scheduler has been proposed. Since the direct solution is complicated, we have further investigated how to simplify the analysis by an approximation approach. The accuracy of the analytic model has been verified by the simulation.
4. The QHAD scheduling algorithm for systems with both persistent and dynamic flows has been proposed and its throughput-optimality has been investigated.

The proposed algorithm has been extended into a more general form to provide more flexibility in scheduling algorithm design.

## 6.2 Future Research Issues

There are many open issues for further research in the topics we discussed in this dissertation.

1. For the scheduling algorithm design for systems with flow-level dynamics, we list the possible further research issues as follows. First, how the current algorithm proposed in Chapter 2 performs in multihop wireless networks should be further investigated. Currently we consider the scheduling problem in the network with one base station which connects to and schedules all the users in the system. A multihop network will not only introduce the interference between adjacent hops, but also increase the packet delivery delay. Thus the design of a throughput-optimal scheduling algorithm for multihop wireless networks with satisfactory delay performance needs to be studied in the future. Second, the current scheduling algorithm is performed in a central scheduler. In practice, a central node may not be always available, and thus how the scheduling algorithm can be performed in a distributive manner is another future topic.
2. For the work on the compatibility between throughput-optimal scheduling algorithms and the TCP protocol, we take the problem of how to guarantee QoS for different services and applications as one future work. In Chapter 3 we classify the flows into different categories based on their physical channel condition. In reality, the classification may also consider the specific QoS requirements of different applications, e.g., IEEE 802.11e introduces multiple traffic classes in the protocol, with each class having its dedicated queue. These classes are defined based on interactivity level of the application, such as VoIP or background traffic. Each class has a weight assigned to it which provides them different quality of services. How to design the scheduling algorithm which is able to provide the differentiated services based on the application QoS requirements should be further discussed.
3. For the work on the queueing behavior analysis, one further research issue is a thorough investigation on the threshold of the traffic intensity when the per-

formance of HAD deviates the M/M/1-D model and converges to the M/D/1 model. Second is the analysis model for the other throughput-optimal scheduling algorithms. The current analysis model is specifically proposed for HAD, which is not accurate enough for the other algorithms. It is also interesting to study a universal analytic model that can describe the queueing behavior of a number of throughput-optimal schedulers, as well as the performance trade-off between them. Third is the mean individual queue length analysis and how to ensure the packet loss ratio in a finite buffer implementation.

4. In Chapter 5, the generalization of the scheduling algorithm can be more universal. Current generalization is for systems with dynamic flows only. How to further generalize throughput-optimal scheduling algorithms in wireless systems with the coexistence of persistent and dynamic flows is worth further attention. Given that the current generalization only brings concave control functions into consideration, the performance of schedulers with concave control functions is another important topic which will make the generalization more complete.

# Bibliography

- [1] *3GPP Technical Specification 36.300, V13.2.0*, January 2016. release 13.
- [2] Matthew Andrews. Instability of the proportional fair scheduling algorithm for HDR. *IEEE Trans. Wireless Commun.*, 3(5):1422–1426, 2004.
- [3] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Rajiv Vijayakumar, and Phil Whiting. Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, 18(2):191–217, 2004.
- [4] Lin Cai, Xuemin Shen, and Jon W. Mark. *Multimedia Services in Wireless Internet: Modeling and Analysis*. John Wiley & Sons, Inc., 2009.
- [5] Lin Cai, Xuemin Shen, Jon W Mark, and Jianping Pan. Performance modeling and analysis of window-controlled multimedia flows in wireless/wired networks. *IEEE transactions on wireless communications*, 6(4):1356–1365, 2007.
- [6] Lin Cai, Xuemin Shen, Jianping Pan, and Jon W Mark. Performance analysis of TCP-friendly AIMD algorithms for multimedia applications. *IEEE Trans. Multimedia*, 7(2):339–355, 2005.
- [7] Jiming Chen, Weiqiang Xu, Shibo He, Youxian Sun, P. Thulasiraman, and Xuemin Shen. Utility-based asynchronous flow control algorithm for wireless sensor networks. *IEEE J. Select. Areas Commun.*, 28(7):1116–1126, 2010.
- [8] Jiming Chen, Weiqiang Xu, Shibo He, Youxian Sun, P. Thulasiraman, and Xuemin Shen. Utility-based asynchronous flow control algorithm for wireless sensor networks. *IEEE J. Select. Areas Commun.*, 28(7):1116–1126, 2010.

- [9] Yi Chen, Xuan Wang, and Lin Cai. Hol delay based scheduling in wireless networks with flow-level dynamics. In *IEEE GLOBECOM*, pages 4898–4903, 2014.
- [10] Yi Chen, Xuan Wang, and Lin Cai. On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks. Technical report, Dept. Elect. Comput. Eng., Univ. Victoria, 2015.
- [11] Yi Chen, Xuan Wang, and Lin Cai. Head-of-line access delay based scheduling algorithm for flow-level dynamics. *IEEE Trans. Veh. Technol.*, *accepted*, 2016.
- [12] Yi Chen, Xuan Wang, and Lin Cai. On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks. *IEEE Trans. Wireless Commun.*, *accepted*, 2016.
- [13] Yi Chen, Xuan Wang, and Lin Cai. Scheduling design for coexistence of persistent and dynamic flows. *IEEE Trans. Netw.*, *to be submitted*, 2016.
- [14] Cisco. Data center bridging. Technical report.
- [15] R. B. Cooper. *Introduction to Queuing Theory, 2nd Edition*. New York: Elsevier North Holland, 1981.
- [16] Robert Elliott. A measure of fairness of service for scheduling algorithms in multiuser systems. In *Proc. IEEE CCECE*, volume 3, pages 1583–1588, 2002.
- [17] Atilla Eryilmaz and R Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *IEEE INFOCOM, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1794–1803, 2005.
- [18] Atilla Eryilmaz, R Srikant, and James R Perkins. Stable scheduling policies for fading wireless channels. *IEEE/ACM Trans. Netw.*, 13(2):411–424, 2005.
- [19] Luigi A Grieco and Saverio Mascolo. Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. *ACM SIGCOMM Comput. Commun. Review*, 34(2):25–38, 2004.
- [20] Jeroen Idserda, Geert Heijenk, and Pieter-Tjerk de Boer. TCP/IP modelling in OMNeT++. 2004.

- [21] Krishna Jagannathan, Mihalis Markakis, Eytan Modiano, and John N Tsitsiklis. Throughput optimal scheduling in the presence of heavy-tailed traffic. In *Allerton'10*, pages 953–960.
- [22] A Jalali, R Padovani, and R Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *Proc. IEEE 51st Semiannu. Veh. Technol. Conf.*, volume 3, pages 1854–1858, Spring, May 2000.
- [23] Bo Ji, Changhee Joo, and Ness B Shroff. Delay-based back-pressure scheduling in multihop wireless networks. *IEEE/ACM Trans. Netw.*, 21(5):1539–1552, 2013.
- [24] Bo Ji, Changhee Joo, and Ness B Shroff. Exploring the inefficiency and instability of back-pressure algorithms. In *IEEE INFOCOM*, pages 1528–1536, 2013.
- [25] Changhee Joo. On the performance of back-pressure scheduling schemes with logarithmic weight. *IEEE Trans. Wireless Commun.*, 10(11):3632–3637, 2011.
- [26] Rafael Laufer, Theodoros Salonidis, Henrik Lundgren, and Pascal Le Guyadec. Xpress: A cross-layer backpressure architecture for wireless multi-hop networks. In *Proc. ACM MobiCom*, pages 49–60. ACM, 2011.
- [27] Bin Li and Atilla Eryilmaz. Optimal distributed scheduling under time-varying conditions: A fast-csma algorithm with applications. *IEEE Trans. Wireless Commun.*, 12(7):3278–3288, 2013.
- [28] Bin Li, Atilla Eryilmaz, and R Srikant. On the universality of age-based scheduling in wireless networks. In *IEEE INFOCOM*, 2015.
- [29] Qiao Li and Rohit Negi. Distributed throughput-optimal scheduling in ad hoc wireless networks. In *IEEE ICC*, pages 1–5, 2011.
- [30] Xiaojun Lin and Ness B Shroff. Joint rate control and scheduling in multihop wireless networks. In *Proc. IEEE CDC*, volume 2, pages 1484–1489, 2004.
- [31] Jia Liu, Atilla Eryilmaz, Ness B Shroff, and Elizabeth S Bentley. Heavy-ball: A new approach to tame delay and convergence in wireless network optimization. In *Proc. IEEE INFOCOM*, 2016.

- [32] Shihuan Liu, Lei Ying, and R Srikant. Scheduling in multichannel wireless networks with flow-level dynamics. In *ACM SIGMETRICS Performance Evaluation Review*, volume 38, pages 191–202, 2010.
- [33] Shihuan Liu, Lei Ying, and R Srikant. Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics. *IEEE/ACM Trans. Netw.*, 19(4):1057–1070, 2011.
- [34] Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP selective acknowledgment options. RFC 2018, Apr. 1996.
- [35] Nick McKeown, Adisak Mekkittikul, Venkat Anantharam, and Jean Walrand. Achieving 100% throughput in an input-queued switch. *IEEE Trans. Commun.*, 47(8):1260–1267, 1999.
- [36] Michael J Neely. Optimal energy and delay tradeoffs for multiuser wireless downlinks. *IEEE Trans. Inf. Theory*, 53(9):3095–3113, 2007.
- [37] Michael J Neely. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Commun. Netw.*, 3(1):1–211, 2010.
- [38] Michael J Neely. Delay-based network utility maximization. *IEEE/ACM Trans. Netw.*, 21(1):41–54, 2013.
- [39] Michael J Neely and Eytan Modiano. Capacity and delay tradeoffs for ad hoc mobile networks. *IEEE Trans. Inf. Theory*, 51(6):1917–1937, 2005.
- [40] Michael J Neely, Eytan Modiano, and Chih-Ping Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Trans. Netw.*, 16(2):396–409, 2008.
- [41] Michael J Neely, Eytan Modiano, and Charles E Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE J. Sel. Areas Commun.*, 23(1):89–103, 2005.
- [42] Jitendra Padhye, Victor Firoiu, Donald F Towsley, and James F Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Trans. Netw.*, 8(2):133–145, 2000.

- [43] QualComm. Qualcomm aims at peer-to-peer with flashlinq. Technical report, 2011.
- [44] Bilal Sadiq, Seung Jun Baek, and Gustavo De Veciana. Delay-optimal opportunistic scheduling and approximations: The log rule. *IEEE/ACM Trans. Netw.*, 19(2):405–418, 2011.
- [45] Bilal Sadiq and Gustavo De Veciana. Optimality and large deviations of queues under the pseudo-log rule opportunistic scheduling. In *Allerton'08*, pages 776–783.
- [46] Bilal Sadiq and Gustavo De Veciana. Throughput optimality of delay-driven maxweight scheduler for a wireless system with flow dynamics. In *Allerton'09*, pages 1097–1102.
- [47] Hulya Seferoglu and Eytan Modiano. TCP-aware backpressure routing and scheduling. In *Proc. IEEE ITA*, pages 1–9, 2014.
- [48] Sanjay Shakkottai and Alexander L Stolyar. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *Translations of the American Mathematical Society-Series 2*, 207:185–202, 2002.
- [49] Leandros Tassiulas. Scheduling and performance limits of networks with constantly changing topology. *IEEE Trans. Inf. Theory*, 43(3):1067–1073, 1997.
- [50] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Autom. Control*, 37(12):1936–1948, 1992.
- [51] Leandros Tassiulas and Anthony Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Inf. Theory*, 39(2):466–478, 1993.
- [52] Peter van de Ven, Sem Borst, and Seva Shneer. Instability of maxweight scheduling algorithms. In *IEEE INFOCOM*, pages 1701–1709, 2009.
- [53] Peter Van de Ven, Sem Borst, and Lei Ying. Spatial inefficiency of maxweight scheduling. In *IEEE WiOpt*, pages 62–69, 2011.

- [54] Peter M van de Ven, Sem C Borst, and Lei Ying. Inefficiency of maxweight scheduling in spatial wireless networks. *Elsevier Computer Communications*, 36(12):1350–1359, 2013.
- [55] Lijun Wang, Lin Cai, Xinzhi Liu, Xuemin Sherman Shen, and Junshan Zhang. Stability analysis of multiple-bottleneck networks. *Computer Networks*, 53(3):338–352, 2009.
- [56] Xin Wang and Zhaoquan Li. Joint optimization of TCP congestion control and distributed CSMA scheduling. In *Proc. IEEE GLOBECOM*, pages 5729–5733, 2012.
- [57] Xuan Wang and Lin Cai. Limiting properties of overloaded multiuser wireless systems with throughput-optimal scheduling. *IEEE Trans. Commun.*, 62(10):3517–3527, 2014.
- [58] Xuan Wang and Lin Cai. Stability region of opportunistic scheduling in wireless networks. *IEEE Trans. Veh. Technol.*, 63(8):4017–4027, 2014.
- [59] Xuan Wang, Yi Chen, Lin Cai, and Jianping Pan. Scheduling in a secure wireless network. In *IEEE INFOCOM*, 2014.
- [60] Shuang Xia and Pu Wang. Distributed throughput optimal scheduling in the presence of heavy-tailed traffic. In *IEEE ICC*, pages 3490–3496, 2015.
- [61] Siyuan Xiang and Lin Cai. Transmission control for compressive sensing video over wireless channel. *IEEE Trans. Wireless Commun.*, 12(3):1429–1437, 2013.
- [62] Yingqun Yu and Georgios B Giannakis. Joint congestion control and OFDMA scheduling for hybrid wireline-wireless networks. In *Proc. IEEE INFOCOM*, pages 973–981, 2007.
- [63] Lei Zheng and Lin Cai. A distributed demand response control strategy using lyapunov optimization. *IEEE Trans. Smart Grid*, 5(4):2075–2083, July 2014.