

Copyright  
by  
Bilal Sadiq  
2010

The Dissertation Committee for Bilal Sadiq  
certifies that this is the approved version of the following dissertation:

**Optimality and Robustness  
in Opportunistic Scheduler Design for Wireless Networks**

Committee:

---

Gustavo de Veciana, Supervisor

---

Jeffrey G. Andrews

---

Aristotle Arapostathis

---

John J. Hasenbein

---

Sanjay Shakkottai

**Optimality and Robustness  
in Opportunistic Scheduler Design for Wireless Networks**

by

**Bilal Sadiq, B.S.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2010

To mamoo jee Ishtiaq and Farooq for science experiments, homework help,  
bicycle rides, and role models.

# Acknowledgments

This research was supported in part by NSF Awards CNS-0721532 and CNS-0917067, AFOSR Award FA9550-07-1-0428, and by Intel.

Foremost, I express my gratitude to my advisor, Prof. Gustavo de Veciana. His dedication to his students and their work has always set an example for me and motivated me to work harder. He teaches with immense clarity. And just as remarkably, he discerns my obscurest statements of new ideas and crisply explains them back to me.

I also thank the professors on my committee: Prof. Ari Arapostathis for numerous discussions and expository write-ups on stochastic control techniques used in this thesis; Prof. John Hasenbein for discussions on fluid models and job scheduling; Profs. Sanjay Shakkottai and Jeff Andrews for teaching me multiple courses and concepts which are extensively used throughout this thesis, as well as for many key discussions.

I also thank Profs. Mor Harchol-Balter and Adam Wierman for discussions related to SRPT service discipline and for sharing their insights. I thank Dr. Alexander Stolyar for discussions on throughput optimality under flow-level dynamics. I thank Shreeshankar Bodas for his student seminars on large deviations theory and bounds, from which I benefitted a lot. I thank Ashwin Sampath and Ritesh Madan for their extensive support during my internship at Qualcomm Flarion Technologies, as well as for sharing their knowledge and insights at numerous subsequent occasions.

I am also very thankful to my friends and colleagues Brian Smith, Alex Icaza, Suju Rajan, Seung Jun Baek, Shailesh Patil, and Hongseok Kim. Throughout my

time in grad school, I have greatly benefited from Brian's views and genius through not just technical discussions, but that of all kind.

Lastly and most uniquely, I am grateful to Melanie Gulick, who, like a saint, patiently guides, assists, and counsels all lost and distracted ECE graduate students.

To all of above I am truly grateful, and then there are those I need not speak of, for I am indebted to them so obviously, eternally.

# Optimality and Robustness in Opportunistic Scheduler Design for Wireless Networks

Publication No. \_\_\_\_\_

Bilal Sadiq, Ph.D.

The University of Texas at Austin, 2010

Supervisor: Gustavo de Veciana

We investigate in detail two multiuser opportunistic scheduling problems in centralized wireless systems: the scheduling of *delay-sensitive* flows with packet delay requirements of a few tens to few hundreds of milliseconds over the air interface, and the scheduling of *best-effort* flows with the objective of minimizing mean file transfer delay.

Schedulers for delay-sensitive flows are characterized by a fundamental trade-off between *maximizing total service rate by being opportunistic* and *balancing unequal queues (or delays) across users*. In choosing how to realize this tradeoff in schedulers, our key premise is that *robustness* should be a primary design objective alongside performance. Different performance objectives – mean packet delay, the tail of worst user’s queue distribution, or that of the overall queue distribution – result in remarkably different scheduling policies. Different design objectives and resulting schedulers are also not equally robust, which is important due to the uncertainty and variability in both the wireless environment and the traffic. The proposed class of schedulers offers low packet delays, less sensitivity to the scheduler parameters and channel characteristics, and a more graceful degradation of service in terms of the fraction of users meeting their delay requirements under transient overloads, when compared with other well-known schedulers.

Schedulers for best-effort flows are characterized by a fundamental tradeoff between *maximizing the total service rate* and *prioritizing flows with short residual sizes*. We characterize two regimes based on the “degree” of opportunistic gain present in the system. In the first regime – where the opportunistic capacity of the system increases sharply with the number of users – the use of residual flow-size information in scheduling will *not* result in a significant reduction in flow-level delays. Whereas, in the second regime – where the opportunistic capacity increases slowly with the number of users – using flow-size information alongside channel state information *may* result in a significant reduction. We then propose a class of schedulers which offers good performance in either regime, in terms of mean file transfer delays as well as probability of blocking for systems that enforce flow admission control.

This thesis provides a comprehensive theoretical study of these fundamental tradeoffs for opportunistic schedulers, as well as an exploration of some of the practical ramifications to engineering wireless systems.



# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xiv</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 QoS scheduling . . . . .	2
1.1.1 Related work . . . . .	3
1.1.2 Contributions . . . . .	6
1.2 Best-effort scheduling . . . . .	8
1.2.1 Related work . . . . .	9
1.2.2 Contributions . . . . .	11
<b>Part I QoS scheduling</b>	<b>13</b>
<b>Chapter 2. Mean delay optimal schedulers: a case for radial sum-rate monotonicity</b>	<b>14</b>
2.1 Overview, main contributions, and organization . . . . .	14
2.2 System model . . . . .	18
2.2.1 Optimality criterion . . . . .	19
2.2.2 Stabilizability and opportunistic capacity region . . . . .	20
2.3 Characterization of delay-optimal policy . . . . .	21
2.4 Radial sum-rate monotonicity: comparing the optimal policy with known heuristics . . . . .	25
2.4.1 The tradeoff under delay-optimal schedulers . . . . .	26
2.4.2 The tradeoffs under MaxWeight and Exp Rule . . . . .	28
2.5 Improved throughput-optimal policies . . . . .	30
2.5.1 Sufficient conditions for throughput-optimality . . . . .	30
2.5.2 The Log Rule . . . . .	33
2.6 Evaluating opportunistic scheduler design objectives – simulations . .	36
2.6.1 Simulation model and operational scenarios . . . . .	36

2.6.2	Discussion of results and insights . . . . .	39
2.7	Scheduling in Multichannel Systems . . . . .	41
2.7.1	Modifications to the system model . . . . .	42
2.7.2	Scheduling and resource allocation polices . . . . .	43
2.8	Conclusion . . . . .	45
 <b>Chapter 3. Large deviations sum-queue optimality of a radial sum-rate monotone opportunistic scheduler</b>		<b>47</b>
3.1	Overview, main contributions, and organization . . . . .	47
3.2	System model . . . . .	49
3.3	Throughput-optimal schedulers and radial sum-rate monotonicity . . . . .	53
3.4	The pseudo-Log scheduling rule . . . . .	57
3.5	Main result . . . . .	60
3.6	Fluid-scaled processes and large deviation principles . . . . .	61
3.7	Lower bound on overflow probability under any scheduling rule . . . . .	64
3.7.1	Proof of Theorem 3.1-(i) . . . . .	66
3.8	Upper bound on overflow probability under the p-Log rule . . . . .	68
3.9	Local fluid sample path . . . . .	69
3.9.1	Scheduling over time scales of LFSP . . . . .	72
3.10	Proof of Theorem 3.1-(iii): optimality of the p-Log rule . . . . .	80
3.11	Conclusion and extensions . . . . .	85
 <b>Part II Best-effort scheduling</b>		<b>94</b>
 <b>Chapter 4. Throughput optimality of <i>delay</i>-driven MaxWeight scheduler for wireless systems with <i>flow dynamics</i></b>		<b>95</b>
4.1	Overview and main contributions . . . . .	95
4.2	System model . . . . .	96
4.2.1	Delay-driven MaxWeight scheduler . . . . .	99
4.3	Main result . . . . .	99
4.3.0.1	Fluid limit of the <i>deterministic</i> initial state . . . . .	101
4.3.0.2	Deterministic fluid limit of the <i>random</i> state for $t \geq 0$ . . . . .	103
4.3.0.3	Dynamics and derivatives of fluid limit for $t \geq 0$ . . . . .	105
4.3.0.4	Using Proposition 4.1 to conclude the proof of Theorem 4.1 . . . . .	108
4.4	Conclusion and extensions . . . . .	109

<b>Chapter 5. Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics</b>	<b>111</b>
5.1 Overview and main contributions . . . . .	111
5.2 System model . . . . .	112
5.2.1 Wireless system with heterogeneous channels . . . . .	113
5.2.2 Equivalent system with i.i.d. channels but heterogeneous file sizes	115
5.3 Servers with state-dependent capacity regions and $M/GI/\mathcal{C}$ queue . .	116
5.4 Dynamic wireless system as an $M/GI/\mathcal{C}$ queue . . . . .	118
5.4.1 Opportunistic capacity region . . . . .	118
5.4.2 Time-scale separation argument and reduction to an $M/GI/\mathcal{C}$ queue . . . . .	121
5.5 Transient system . . . . .	123
5.5.1 Optimal scheduler for polymatroid capacity regions $\bar{\mathcal{C}}$ . . . . .	124
5.5.2 OPS scheduler for capacity regions $\bar{\mathcal{C}}$ and $\mathcal{C}$ . . . . .	126
5.5.3 Competitive ratio of OPS . . . . .	127
5.5.4 Discussion . . . . .	129
5.5.5 SRPT-OPS scheduler for capacity regions $\mathcal{C}$ and $\bar{\mathcal{C}}$ . . . . .	130
5.6 Dynamic system in steady state . . . . .	132
5.6.1 Simulation of a dynamic wireless system with time varying channels and the corresponding $M/GI/\bar{\mathcal{C}}$ queue . . . . .	136
5.7 Extensions and comments . . . . .	141
5.8 Conclusion . . . . .	142
<b>Conclusion</b>	<b>152</b>
<b>Appendices</b>	<b>156</b>
<b>Appendix A. Fluid-scale asymptotic optimality and RSM policies</b>	<b>157</b>
A.0.1 Deterministic Fluid Model . . . . .	158
A.0.2 Fluid-scale asymptotic optimality . . . . .	159
A.0.3 Relation between $\boldsymbol{\mu}^*$ and $\boldsymbol{\mu}^{F^*}$ . . . . .	159
A.0.4 Relation between RSM Policy for MDP and Greedy Policy for Fluid Model . . . . .	160
A.0.5 Main Result . . . . .	161

<b>Appendix B. A note on delay-optimal opportunistic schedulers for multichannel systems</b>	<b>164</b>
B.1 System Model . . . . .	165
B.2 Alternative view of the system . . . . .	167
B.2.1 From “On-Off servers” to “polymatroid service regions” . . . . .	168
B.2.2 From “server allocation through bipartite matching” to “service allocation” . . . . .	169
B.2.3 Main Result . . . . .	170
B.2.3.1 Absence of tradeoff between <i>maximizing total service rate</i> and <i>balancing unequal queues</i> . . . . .	172
B.2.3.2 Restatement of the delay-optimal policy MTLB for divisible servers . . . . .	173
B.2.3.3 Nonlinearity of delay-optimal policy for <i>multiserver</i> . . . . .	174
B.2.3.4 Restatement of the delay-optimal policy for integral servers	174
B.2.3.5 “Server-side greedy” policy . . . . .	174
B.3 Extending the schedulers for on-off servers to multi-rate servers . . . . .	175
B.3.1 Service region associated with multi-rate servers . . . . .	175
B.3.2 Scheduling policies for systems with multi-rate servers . . . . .	175
<b>Bibliography</b>	<b>177</b>
<b>Vita</b>	<b>185</b>

## List of Tables

2.1	Mean data rate supported by wireless channel of each user . . . . .	37
2.2	Parameters used for each scheduling policy . . . . .	38
3.1	Intermediate steps towards proving Theorem 1-(iii). . . . .	70

# List of Figures

2.1	Partitions of queue state space under the optimal policy . . . . .	24
2.2	Curves along which the direction is held constant by the vector fields of MaxWeight and Exp rule. . . . .	29
2.3	Curves along which the direction is held constant by the vector field of Log-Rule . . . . .	33
2.4	Simulation-based performance comparisons for three opportunistic scheduling policies, Log Rule, Exp rule and Max Weight . . . . .	37
3.1	Capacity region depicting Minkowski addition, outer-normal vectors, and maximal vertices; the resulting switching curves under MaxWeight, Exp rule, and Log rule. . . . .	56
3.2	Switching curves under p-Log rule. . . . .	59
3.3	<i>Optimal</i> capacity vector $v^*(\lambda, \pi)$ . . . . .	65
3.4	Partitions and switching curves on the space of local fluid queue $\diamond q$ . . . . .	74
3.5	Illustration of original trajectory and its extension. . . . .	83
4.1	System model. . . . .	98
5.1	Capacity region $\mathcal{C}_2$ and the tightest polymatroid upper bound $\bar{\mathcal{C}}_2$ . . . . .	120
5.2	Competitive ratio of OPS . . . . .	129
5.3	Mean number of users in the system and mean sojourn time for unbounded capacity functions . . . . .	134
5.4	Mean number of users in the system and mean sojourn time for bounded capacity function . . . . .	134
5.5	Mean sojourn times under SRPT-OPS( $n^*$ ), OPS, and SRPT-HPR for capacity function $\mathbf{g}^{(3)}$ and user arrival rate $\lambda = 1.8$ user/sec. . . . .	135
5.6	Normalized capacity functions associated with various fading scenarios, and their approximate matches. . . . .	137
5.7	Mean number of users and mean sojourn time under various schedulers . . . . .	138
5.8	Mean sojourn time for <i>DSwTC-B</i> under OPS and SRPT-OPS( $n^*$ ). . . . .	141
5.9	Plot of $\left(\frac{\pi_k}{\theta_k}, k > 0\right)$ for various values of $a \in [0, 1)$ . . . . .	149
A.1	Relationships between various policies for the MDP and their counterparts for the deterministic fluid model. . . . .	158
A.2	Fluid model: (a) capacity region, (b) fluid trajectory under the <i>greedy</i> policy. . . . .	161

B.1	Illustration of $\mathcal{C}(\mathbf{r})$ for a $\bar{n} = 2$ user $\bar{m} = 6$ server system. . . . .	168
B.2	Illustration of $\mathcal{Q}(s(t))$ for various states $s$ and the corresponding least weakly submajorized element $\mathbf{q}^*(t + 1)$ . . . . .	172

# Chapter 1

## Introduction

Current and emerging broadband wireless systems, *e.g.*, LTE and WiMax, are evolving towards *purely scheduled* systems, in that all traffic including delay-sensitive services (*e.g.*, VoIP or SIP signaling, see [1]) needs to be scheduled. These systems also feature [2, Section 10.2]:

- fine granularity of resource allocation, *e.g.*, 180KHz Resource Block times 1ms Transmission Time Interval for LTE;
- fast MAC turnaround times, *e.g.*, 2ms for LTE;
- along with highly configurable schemes to obtain channel state information at the transmitter.

This allows for exploitation of time/frequency channel selectivity through opportunistic scheduling, and thus enables higher user throughputs. However, unlike what is typically the case in wired systems, more (opportunistic) capacity does not easily translate to better user-perceived performance [3].

More precisely, in the case of delay-sensitive flows, *e.g.*, voice and video packet streams, the scheduler has to carefully tradeoff *the maximization of total transmission rate* versus *the balancing of packet delays across users*. In other words, sometimes one may need to schedule users whose delays/queues are becoming large but whose current channel is not the most favorable. This tradeoff is rigorously investigated in the first half of this thesis.



Similarly, in the case of best-effort flows, *e.g.* file transfers and web browsing, the scheduler has to tradeoff *maximization of total transmission rate* versus *prioritizing flows with short residual size*, so as to reduce transfer delays or enhance web browsing interactivity. This tradeoff is investigated in the second half of this thesis.

Throughout the thesis, the regime of most interest is where the channel coherence time is long enough for permit channel tracking and opportunistic scheduling, but short enough such that the packet or file transfer delays (depending on the context) exceed a few multiple coherence times. The tradeoffs mentioned above, and therefore all results in this thesis, are also applicable to a scenario where the wireless channel for each user is constant over time but exhibits frequency selectivity.

Lastly, we would like to add that the above-mentioned tradeoffs are not specific to wireless systems. They appear in general queueing problems involving multiple *parallel unrelated processors* [4], where “unrelated” refers to the fact that the speed/rate of a processor can be different for different classes of jobs/users. An example would be a specialized machine or computer, or a shared wireless channel, where the latter has an added complexity if it is time varying. Therefore, some of the analytical results presented in this thesis have broader applicability to the control of parallel queues and servers.

## 1.1 QoS scheduling

In the first part of this thesis, we address the design of packet scheduling policies for a fixed number of delay-sensitive or QoS (quality of service) users sharing a wireless channel. Each user’s data arrives to a queue as a random stream where it awaits transmission. The wireless channel is time-varying in that the transmission rates supported for each user vary randomly over time. More generally, a scheduler can be permitted to allocate rates to the users from a polytope service region, where

the *service region depends on the channel state*. If the channel state is available, a policy can schedule users so as to exploit favorable channels, *e.g.*, schedule the user which currently has the highest rate – this is referred to as opportunistic scheduling [5–7].

Our objective is to evaluate the design of queue-and-channel-aware schedulers both from the point of view of performance and robustness. By robustness we informally mean a scheduler’s ability to perform well for the majority of users under unpredictable/changing conditions and even transient ‘overloads.’ If the system becomes temporarily overloaded, it is desirable for an opportunistic scheduler to gracefully degrade the service seen by the users. Though there has been a substantial amount of work on opportunistic schedulers, it is still unclear whether scheduler design should be guided by the objectives like minimizing mean packet delay and the asymptotic probability of sum-queue overflow, or instead by objectives like minimizing the asymptotic probability of max-queue overflow. In these considerations lies the motivation for the first of this thesis and our efforts to leverage analysis, where possible, and simulation to reach a better understanding of this problem.

### 1.1.1 Related work

To put our work into context, we begin by summarizing some of the key related work in this area. Among many others, [7] considers opportunistic scheduling in a setting where users’ queues are *infinitely backlogged*. They identify channel-aware opportunistic scheduling policies, which maximize the sum throughput under various types of fairness constraints. The missing element in this work is the impact of queueing dynamics. Recently, [8] showed that under a *constant* load, scheduling algorithms that are oblivious to queue state will incur an average delay that grows linearly in the number of users, whereas, channel-and-queue aware schedulers can achieve an average delay that is independent of the number of users. Even before

this, it was immediately recognized that, when queueing dynamics are introduced, opportunistic scheduling policies which are solely channel-aware may not be stable (*i.e.*, keep the users' queues bounded) unless the policy is chosen carefully, *e.g.*, using prior knowledge of mean arrival rates [9]. For this reason, a substantial focus was placed on designing schedulers that are both channel- and queue-aware and provably *throughput-optimal*, *i.e.*, ensure the queues' stability without any knowledge of arrival and channel statistics if indeed stability can be achieved under any policy. Except for some degenerate cases, such policies must tradeoff *maximizing current transmission rate* (*e.g.*, scheduling the queue with the best channel) versus *balancing unequal queues* (*e.g.*, scheduling the longest queue). Balancing queues, avoids empty queues, which enhances the ability to exploit high channel variations in the future. *We will refer to this tradeoff many times throughout the first half of this thesis.* Two representative classes of policies known to be throughput-optimal are MaxWeight [10] (also known as Modified Largest Weighted Work/Delay First) and Exp rule [11]. Yet, stability is a weak form of performance optimality.

Thus, it is of interest to study opportunistic policies that are *delay-optimal*, *e.g.*, polices that minimize the overall average delay (per data unit) seen by the users; or policies which minimize the probability that either the sum-queue or the largest queue overflows a large buffer. These polices are harder to characterize for servers with time-varying capacity, but some results are available that we briefly discuss next.

In [12] and [13] the Longest-Connected-Queue (LCQ) and Longest-Queue-Highest-Possible-Rate (LQHPR) policies are introduced. Strong results are shown for these policies; they stochastically minimize the max and sum queue process, and thus also the max and sum queue tails and mean delay. However, in addition to assuming certain symmetry conditions on arrival and channel statistics, [12] is limited to on-off channels where only a single user can be scheduled per time slot, and [13] assumes that the scheduler can allocate service rates from the current information theoretic

multiuser capacity region. In both cases, the above-mentioned tradeoff between queue balancing and throughput maximization is absent. Indeed in [12], all policies that pick a connected queue result in the same transmission rate, whereas, in case of [13], all policies that pick a service vector from the maximal points of the current capacity region, *i.e.* points on the max-sum-rate face, result in the same overall transmission rate. Thus one can achieve the *queue balancing* goal, without ever compromising *throughput*. Not surprisingly, in both cases the optimal policy turns out to be greedy, in that it allocates as much service rate as possible to the longest/longer queues.

A related server allocation problem is studied in [14]. The paper considers minimizing the mean delay in a two queue system where each queue has a dedicated server and a third server can be dynamically shared between them. As a result, the two queues can be allocated service rates from a polymatroid capacity region, thus the objective of queue balancing can again be achieved without compromising the total service rate. However, without the underlying symmetry assumptions of [12] and [13], only the existence of a monotone increasing switching curve on the queue state space is shown; note that the switching curve under LCQ and LQHPR policies lies along the line where both queues are equal. For a system with a general compact, convex, and coordinate convex capacity region and any finite number of queues, [15] gives a large deviations principle (LDP) for transient queue process under MaxWeight scheduler. This LDP can be used to compute, *e.g.*, the asymptotic probability of sum-queue or max-queue overflow, as well as the corresponding likely modes of overflow. Although the capacity region is not changing over time, the region is such that a scheduler must tradeoff maximizing total service rate with balancing unequal queues. Therefore this result is insightful in relating the modes overflow to the tradeoff made by the MaxWeight scheduler. A more recent work [16] gives a many sources large deviations result for the MaxWeight scheduler for a similar capacity region.

Finally, relaxing the symmetry assumptions of [12] and [13], the works in

[17–19] consider the asymptotic probability of max-queue overflow. The server capacity in [17], though time-varying, is identical for all users at any given time, thus the need to tradeoff queue-balancing versus service rate maximization is again absent. In fact, the sum-queue process in [17] is identical for all work conserving schedulers. However [18] and [19] consider a server with *asynchronously* time-varying capacity across users. [18] studies the asymptotic probability of max-queue overflow under MaxWeight scheduler and shows that as the exponent of queue length in the MaxWeight scheduler,  $\alpha$ , becomes large, the asymptotic probability of max-queue overflow under MaxWeight approaches the minimum achievable under any other scheduler. A stronger result is shown in [19], that is, the Exp rule scheduler in fact minimizes the steady state asymptotic probability of max-queue overflow. Indeed the models in [18] and [19] accurately capture a wireless channel shared by heterogenous users, and exhibit the tradeoff between queue-balancing and service rate maximization. Existence of this tradeoff also implies that, unlike the LCQ and LQHPR policies, the asymptotic optimality of Exp rule does not translate to minimizing the asymptotic probability of sum-queue overflow or the mean delay. In fact the policies that minimize mean delay and sum-queue overflow are very different and we believe are of practical interest.

In a related work on input-queued switches, [20] explains the conjecture that de-emphasizing queue-balancing improves mean delay of Maximum-Weight Matching algorithms.

### 1.1.2 Contributions

We begin Chapter 2 by characterizing mean-delay optimal opportunistic schedulers for heterogenous systems where the arrival and channel statistics are *known*. By considering a simple model falling in the classical Markov decision process framework, we show through numerical computation, that mean delay optimal policies exhibit a property we call *radial sum-rate monotonicity*. That is, when user queues grow

linearly (*i.e.* scaled up by a constant) the scheduler allocates service in a manner that *de-emphasizes the balancing of unequal queues* in favor of *maximizing current service rate (being opportunistic)*. This is in sharp contrast to previously proposed policies, *e.g.*, MaxWeight and Exp rules, which, nevertheless, have the advantage of being throughput-optimal.

We therefore propose a new class of policies, called the Log rule, that are radial sum-rate monotone (RSM) and provably throughput optimal. We also show in Appendix A that under sufficient load, RSM policies like the Log rule are *fluid-scale asymptotic optimal*, which roughly means that when starting with a large initial queue state, the queue trajectories under the mean delay optimal policy and under an RSM policy are identical on a scaled queue state-space (scaled by the norm of the initial state). Moreover, in Chapter 3 we use the approach of [19] to show that a candidate RSM policy in fact minimizes the asymptotic probability of *sum-queue overflow*. The “likely mode of overflow” under the Log rule is categorically different from that under the Exp rule.

So we have at our disposal several opportunistic scheduling policies which are *optimal* for different objectives. The question remains, in designing an opportunistic scheduler should one be guided by mean or asymptotic tail results, and should one focus on individual worst case or overall system criteria? In this regard, we use extensive simulation to attempt to gain further insight on the question, and evaluate the comparative effectiveness of various policies.

We also extend the proposed scheduling policies to multichannel systems supporting a large number of users, *e.g.*, OFDMA-based WANs such as LTE/WiMAX downlinks, where bandwidth and power resources can be shared by multiple users over a scheduling/transmission time interval. Recognizing practical limits on the spectral granularity of channel feedback in multichannel systems, we suggest a queue-aware

convex program formulation to realize opportunistic scheduling and resource allocation policies.

## 1.2 Best-effort scheduling

For best effort flows, the relevant performance metrics are defined over longer time scales, *i.e.*, the time scales of flow-level dynamics, *e.g.*, file transfer delays or web browsing interactivity. Unlike a system where there is a fixed number of users/flows and each flow generates a *stationary packet arrival process*, in this setting the arrivals correspond to new flows and users, *i.e.*, files to be transferred associated with different users, and thus the number of ongoing flows in the system is dynamic. Each flow can be viewed as having its own queue associated with the residual data that needs to be transmitted in order to successfully transfer a file or web page.

In the second part of this thesis, we consider a wireless downlink (or uplink) shared by a *dynamic* population of best-effort flows/users.<sup>1</sup> Flows of random size (bits) arrive at the base station at random times, and leave when their transfer is complete. As before, the transmission rate supported by the wireless channel (referred to as the channel state) of each user varies randomly over time and is independent of that of other users. The scheduling problem in this context is to select which flow to serve based on the current system state (*e.g.*, residual flow sizes and channel states of the contending flows), *with the objective of minimizing the file transfer delay (mean sojourn time of the flows)*. This scheduling problem involves a key tradeoff between *prioritizing flows with short residual sizes* and *maximizing the total service rate*; below we elaborate further on this.

One *extreme* of this problem is the case where the wireless channel is in fact a

---

<sup>1</sup>Each flow is associated with a unique user downloading (or uploading) a file from the base station; the terms flow, user, and file will sometimes be used interchangeably.

constant capacity server. This reduces the system to a work-conserving G/G/1 queue, and in that case the Shortest Remaining Processing Time (SRPT) scheduler [21] is known to be optimal in a very strong sense: SRPT not only minimizes the mean sojourn time but also the number of flows in the system at all times.

However, unlike a constant capacity server, the time-varying nature of wireless channels provides a scope for opportunistic scheduling [5–7]. Therefore, the server’s capacity is not only time-varying but also growing with the number of flows present in the system. Another *extreme* case of our problem is that where a *fixed* number of *infinitely backlogged* flows share the wireless downlink: opportunistic schedulers that maximize long-run average service rate/throughput under various fairness constraints are also well understood; see, *e.g.*, [7].

### 1.2.1 Related work

The work in [22] considers a dynamic heterogenous wireless system under a *specific* flow-size oblivious scheduler, namely, Proportional Fair (PF). Using a time scale separation argument, it is shown that such a system can be modeled as a multi-class Processor Sharing (PS) queue where the server capacity/speed increases with the total number of users in the queue. This model allows one to obtain explicit formulas for the mean sojourn time as well as the queue length distribution and the stability region of the system. The result is complemented by simulations which show that even when some of the relatively more restrictive assumptions (regarding certain *symmetry* of the normalized channel variations across various classes) do not hold, modeling the system as a multi-class PS is still quite accurate. In this thesis, we will use a similar model for the wireless channel as the above work. Note that [23] further shows that the sojourn times decrease if channels change quickly. Therefore, strictly speaking, using a time-scale separation argument to obtain a multi-class Processor Sharing model leads to optimistic performance estimates. Neither [23] nor [22] give



bounds on the sub-optimality of mean sojourn time under PS/PF. *Such bounds are developed in this thesis.*

In the literature, there have been various efforts to combine SRPT and opportunistic scheduling. However, few analytical results are available. Most work has focused on designing and investigating heuristics through a mix of simulation and analysis.

Both [24] and [25] consider a *transient* system, *i.e.*, the problem of transmitting a fixed set of finite sized files from the base station to their respective destination users, with the objective of minimizing the total expected sojourn time. [24] shows that the problem can be set up as a Markov decision process (MDP) with a finite state space and numerically solved. However, perhaps due to having a very large state-space for any useful system, it is difficult to characterize any structural properties of the optimal scheduler from the computed solution. [25] explores various ways of combining SRPT and opportunistic scheduling and provides bounds on mean sojourn time for the proposed heuristics. *In this thesis, we will show that the transient system studied in [24] and [25] can be modeled as a deterministic dynamic program, and explicitly give the optimal scheduler and sojourn times for certain insightful examples.*

Both [26] and [27] consider dynamic systems where new flows arrive randomly. [26] considers the case where *current* channel state information is not available to the scheduler. [27] motivates various heuristics that combine SRPT and opportunistic scheduling, and compares them via extensive simulation. From the presented results, one can make the qualitative observation that schedulers which perform well in various situations do *not* excessively compromise opportunistic gain to favor flows with short residual sizes. [27] also points out a necessary condition for the stabilizability of a dynamic system and suggests that this condition may be *sufficient*, but stops short of a formal proof.

Recently, [28] has formally characterized the stability region of such a system, and (implicitly) shown that, asymptotically in the number of flows, any throughput-optimal scheduler must fully exploit the opportunistic gain. It is also shown that a version of MaxWeight scheduler which prioritizes flows with *long* residual sizes can indeed be unstable.

[29] deals with designing practically viable throughput optimal schedulers and proposes a measurement-based MaxQuantile-like scheduler [30, 31]. The problem of scheduling a mix of best effort and QoS flows is also considered. Once again, the schedulers which seem to perform best in various simulation settings are those which do not, or only minimally, compromise the opportunistic gain.

### 1.2.2 Contributions

Recall that [28] showed that the queue-driven MaxWeight scheduler is *not* throughput optimal<sup>2</sup> in the presence of flow-level dynamics. In Chapter 4, we show that the version of MaxWeight which prioritizes flows that are experiencing long *delays* is nevertheless throughput optimal. Despite its throughput optimality, delay-driven MaxWeight may still be unsuitable for use in a real wireless system: the scheduler may excessively compromise opportunistic capacity until a significant fraction of the contending flows have a large and nearly equal sojourn time. Moreover, the relation between the *residual flow-size* and its *current delay* is unclear.

Therefore, in view of the above-mentioned tradeoff, next we investigate schedulers which explicitly use the residual file-size information and seek to minimize the mean file transfer delay. The main analytical results are given in Chapter 5, specifically Theorems 5.2 and 5.3. Theorem 5.2 characterizes the competitive ratio for flow-size-oblivious opportunistic schedulers, like Proportional Fair (PF) [6] or MaxQuan-

---

<sup>2</sup>Queue-driven Exponential rule and Log rule can similarly be shown to be not throughput optimal in the dynamic flow setting.

tile [30] [31], for a *transient* system, and shows that the presence of opportunistic gain mitigates their sub-optimality. Using this, we characterize two regimes based on the “degree” of opportunistic gain:

- A regime where the use of residual flow-size information in scheduling will *not* result in a significant reduction in flow’s delay.
- A regime where optimally using flow-size information alongside channel state information *may* result in a significant reduction in flow’s delay.

More specifically, but still informally, if the opportunistic capacity of the wireless channel increases rapidly in the number of users, *e.g.*, as  $\log(n)$  or  $\log \log(n)$  where  $n$  is the number of users, then the mean sojourn time under a purely opportunistic scheduler like MaxQuantile or PF is only about 1–20% higher than the minimum possible. If, however, the opportunistic capacity increases more *slowly* (*e.g.*, as  $1 - a^n$  for  $a \in [0, 1)$ ), a significant reduction in mean sojourn time may be achievable if schedulers exploit the residual flow-size information. Using these insights, we propose a class of schedulers which is simple to implement and offers good performance irrespective of the operating regime – this is analyzed in Theorem 5.3.

Overall, this thesis presents some significant theoretical results and accompanying practical insights that can serve as the foundation towards building better opportunistic schedulers for next generation wireless systems.

# Part I

## QoS scheduling

## Chapter 2

# Mean delay optimal schedulers: a case for radial sum-rate monotonicity

### 2.1 Overview, main contributions, and organization

In this chapter, we consider the design of multiuser opportunistic packet schedulers for users sharing a time-varying wireless channel from performance and robustness points of view. For a simplified model falling in the classical Markov decision process framework, we numerically compute and characterize mean-delay-optimal scheduling policies. The computed policies exhibit *radial sum-rate monotonicity*: as users' queues grow linearly, the scheduler allocates service in a manner that de-emphasizes the *balancing of unequal queues* in favor of *maximizing current system throughput* (being opportunistic). This is in sharp contrast to previously proposed throughput-optimal policies, *e.g.*, Exp rule and MaxWeight (with *any* positive exponent of queue length). In order to meet performance and robustness objectives, we propose a new class of policies, called the Log rule, that are radial sum-rate monotone (RSM) and provably throughput optimal. We use extensive simulations to explore various possible design objectives for opportunistic schedulers. When users see heterogeneous channels, we find that emphasizing queue balancing, *e.g.* Exp rule and MaxWeight, may excessively compromise the overall delay. Finally, we suggest an implementation of the proposed policies for scheduling and resource allocation in OFDMA-based multichannel systems.

## Relation to other chapter

In the next chapter we will show that for a two-user system, an RSM policy minimizes the asymptotic probability of sum-queue overflow. As a corollary of the large deviation results, we show in Appendix A that under sufficient load, RSM policies are *fluid-scale asymptotic optimal* (FSAO). This roughly means that, when starting with a large initial queue state, the queue trajectories under the mean delay optimal policy and under an RSM policy are identical on a scaled queue state-space (scaled by the norm of the initial state).

In Appendix B, we show that even though mean-delay optimal schedulers for single-channel systems can be described as a *linear* program, such schedulers become a nonlinear convex program for multichannel systems. This provides analytical support for the proposed scheduling and resource allocation policies for multichannel system.

## Contributions

We begin by characterizing mean-delay optimal opportunistic schedulers for heterogenous systems where the arrival and channel statistics are known. We consider a simple model falling in the classical Markov decision process framework, where we can numerically compute the optimal scheduling policy. Our first contribution is showing through numerical computation that mean delay optimal policies exhibit *radial sum-rate monotonicity* (RSM), *i.e.*, when user queues grow linearly (*i.e.* scaled up by a constant) the scheduler allocates service in a manner that de-emphasizes the balancing of unequal queues in favor of maximizing current system throughput (being opportunistic). This is in sharp contrast to previously proposed policies, *e.g.*, MaxWeight and Exp rules, which, nevertheless, have the advantage of being throughput-optimal. Our second contribution is to propose a new class of policies, called the Log-rule, that are radial sum-rate monotone and provably throughput optimal. These policies are favorable both in terms of mean delay and robustness. Our

simulations for a realistic wireless channels confirm the superiority of the Log-rule which achieves a 20-75% reduction in the mean packet delay. The Log rule is proposed as a practical solution but is not provably mean-delay optimal. However, in the next chapter, we use the approach of [19] to show that the candidate RSM policy indeed minimizes the asymptotic probability of *sum-queue* overflow.

So we have at our disposal several opportunistic scheduling policies which are good for different objectives. The question remains, in designing an opportunistic scheduler should one be guided by mean or asymptotic tail results, and should one focus on individual worst case or overall system criteria? In this regard, we use extensive simulation to attempt to gain further insight on the question, and evaluate the comparative effectiveness of various policies.

We also extend the proposed scheduling policies to multichannel systems supporting a large number of users, *e.g.*, OFDMA-based WANs such as LTE/WiMAX networks, where bandwidth and power resources can be shared by multiple users over a scheduling/transmission time interval. Recognizing practical limits on the spectral granularity of channel feedback in multichannel systems, we suggest a queue-aware convex program formulation to realize opportunistic scheduling and resource allocation policies.

We make the following observations:

**Minimizing mean delay vs asymptotic tails** Based on simulations we observe that when users see heterogenous channels, policies such as Exp rule that aim to minimize the exponential decay rate of delay distribution tail of the *worst* user may excessively compromise average delay, in some cases penalizing the tail distributions of many of the users. Our simulations show that Log rule can achieve better mean delays (overall and on a per user basis) and comparable or better distribution tails for many, if not all, the users under reasonably high loads.

**Graceful degradation of service** Due to the uncertain and changing characteristics of wireless channels, precise resource allocation to meet quality of service requirements (QoS) for real-time or streaming flows is likely to be virtually impossible. As such, a desirable design objective is for a scheduler to gracefully degrade. If there is a change in the environment causing a temporary overload, then as many users as possible should meet their QoS requirements rather than all failing. Our simulation results show that Log-rule compares favorably in this regard. In a system with unpredictable heterogeneous channels, there will be a wider disparity in the performance users see under the Log rule, but a substantial number of users does very well. Hence depending on the QoS objective and specific character of the change in user's channels, one could end up with no users seeing acceptable performance under the Exp rule while, say, half the users meet their QoS requirement under Log rule. Finally we note that Log rule's underlying goal of minimizing mean packet delays might be a desirable objective from the point of view maximizing throughput seen by best effort traffic.

## Organization

Rest of the chapter is organized as follows.

*2.2 System model:* system model and definitions of “opportunistic capacity region” and “scheduling policy” are given. Optimality criterion is defined which is slightly more general than mean packet delay.

*2.3 Characterization of delay-optimal policy:* a time-scale separation argument is used to formulate the optimal policy as a numerically tractable Markov decision process.

*2.4 Comparing the optimal policy with known heuristics:* “Radial Sum-rate Monotonicity” is formally defined, and through numerical computations, the optimal policy is shown to be (weakly) RSM. By contrast, known heuristics (MaxWeight and Exp rule) are shown to differ.



2.5 *Improved throughput-optimal policies*: a new class of scheduling policies, called the Log rule, is proposed which is both RSM and throughput-optimal. A large-deviations optimality result for an RSM policy is also stated.

2.6 *Simulations – evaluating scheduler design objectives*: extensive simulation results are presented for an HDR-like downlink [32], to contrast various scheduling policies and underlying design objectives.

2.7 *Scheduling in multichannel systems*: implementation of the proposed schedulers for OFDMA-based multichannel systems is discussed.

## 2.2 System model

Consider the following continuous time model for scheduling  $\bar{n}$  users' traffic over a shared wireless channel. Each user  $n \in \mathcal{N} \equiv \{1, 2, \dots, \bar{n}\}$  is assigned a queue in which packets with independent and exponentially distributed sizes arrive as a Poisson stream with rate  $\lambda_n$  packets/sec. At any time  $t$ , define the (random) vector  $\mathbf{Q}(t) \equiv (Q_n(t), n \in \mathcal{N}) \in \mathbb{Z}_+^{\bar{n}}$ , where  $Q_n(t)$  denotes the number of packets in the  $n^{\text{th}}$  queue at time  $t$ . The state of the users' wireless channels at time  $t$  is modeled by random  $M(t)$  which can take values in the finite set  $\mathcal{M} \equiv \{1, 2, \dots, \bar{m}\}$ . We assume that for all  $t \neq t'$ , the channel states  $M(t)$  and  $M(t')$  are independent and have the same distribution as a random variable  $M$ . Associated with each channel state  $m \in \mathcal{M}$  is a vector  $\mathbf{r}(m) = (r_n(m), n \in \mathcal{N})$ , where  $r_n(m)$  has the following interpretation: when the channel is in state  $m$  and dedicated to the  $n^{\text{th}}$  user, then  $r_n(m)$  is the instantaneous service rate in packets/sec available to the  $n^{\text{th}}$  user. We allow the channel to be split among users at any time instant according to a stochastic vector  $\boldsymbol{\sigma}(t) = (\sigma_n(t), n \in \mathcal{N})$  – recall that a stochastic vector has non-negative components that sum up to 1 – in which case the service rate available to the  $n^{\text{th}}$  user at time  $t$  will be  $\sigma_n(t)r_n(M(t))$ . We follow the convention that capital letters, *e.g.*,  $\mathbf{Q}(\cdot)$  and  $M(\cdot)$ , denote a random variable, whereas small letters, *e.g.*,  $\mathbf{q}(\cdot)$  and

$m(\cdot)$ , denote a particular realization. Moreover, we will make the natural distinction between “increasing” and “strictly increasing.”

The problem of scheduling users for service is then to choose a vector  $\boldsymbol{\sigma}(t)$  for each time instant  $t$ , such that a given optimality criterion is met. A scheduling policy is said to be static state-feedback if it chooses the vector  $\boldsymbol{\sigma}(t)$  according to a *fixed* rule based solely on the *current* system state  $(\mathbf{q}(t), m(t))$ . More precisely, a static state-feedback scheduling policy is defined as a function  $\mathbf{f}$  which takes the system state  $(\mathbf{q}(t), m(t))$  at any time  $t$  into a stochastic vector  $\boldsymbol{\sigma}(t)$ :

$$\boldsymbol{\sigma}(t) = \mathbf{f}(\mathbf{q}(t), m(t)) . \quad (2.1)$$

Let  $\mathcal{F}$  denote the set of static state-feedback policies. Given the optimality criterion described next, Poisson arrivals, exponentially distributed packet sizes, and i.i.d. channel state vectors, there is no loss of generality in restricting our attention to the policies in  $\mathcal{F}$ .

### 2.2.1 Optimality criterion

Consider a system initiated at  $t = 0$  in state  $\mathbf{Q}(0) = \mathbf{q}(0)$  which evolves under scheduling policy  $\mathbf{f}$ . The expected long-run average queue for the  $n^{\text{th}}$  user is given by,

$$\bar{q}_n(\mathbf{f}) \equiv \limsup_{t \rightarrow \infty} \mathbb{E}_{\mathbf{q}(0)}^{\mathbf{f}} \left[ \frac{1}{t} \int_0^t Q_n(\tau) d\tau \right] , \quad (2.2)$$

where  $\mathbb{E}_{\mathbf{q}}^{\mathbf{f}}$  denotes expectation under  $\mathbf{f}$  conditional on  $\mathbf{Q}(0) = \mathbf{q}$ . We define the delay-optimal scheduling policy  $\mathbf{f}^*$  as the one which minimizes the total weighted average queue length, if it exists, for a given weight vector  $\mathbf{w} = (w_n, n \in \mathcal{N}) > \mathbf{0}$ , *i.e.*,

$$\mathbf{f}^* \in \arg \min_{\mathbf{f} \in \mathcal{F}} \sum_{n \in \mathcal{N}} w_n \bar{q}_n(\mathbf{f}) . \quad (2.3)$$

It follows from Little’s Law that if the process  $(\mathbf{Q}(t), t \geq 0)$  is stationary, this optimality criterion minimizes the overall (weighted) average packet delay seen by the  $n$ -users.

### 2.2.2 Stabilizability and opportunistic capacity region

For  $n \in \mathcal{N}$ , let  $\mathbf{e}_n \in \mathbb{R}_+^{\bar{n}}$  denote the  $n^{\text{th}}$  standard basis vector in  $\mathbb{R}_+^{\bar{n}}$ . For each  $m \in \mathcal{M}$ , let  $\mathcal{C}_m \in \mathbb{R}_+^{\bar{n}}$  denote the convex hull of origin and following  $\bar{n}$  points:

$$r_1(m)\mathbf{e}_1, r_2(m)\mathbf{e}_2, \dots, r_{\bar{n}}(m)\mathbf{e}_{\bar{n}}.$$

That is,  $\mathcal{C}_m$  is the set of service rates that can be jointly offered to the  $\bar{n}$  users, conditional on the channel being in state  $m$ . Define the opportunistic capacity region,  $\mathcal{C}$ , of a channel as the set of long-run *average* service rates that can be jointly offered to the  $\bar{n}$  users under all possible scheduling policies. The opportunistic capacity region associated with the distribution of  $M$  is given by the weighted Minkowski sum of regions  $\mathcal{C}_m$ , *i.e.*,

$$\begin{aligned} \mathcal{C} &\equiv \mathbb{P}(M = 1)\mathcal{C}_1 \oplus \dots \oplus \mathbb{P}(M = \bar{m})\mathcal{C}_{\bar{m}}, \\ &= \left\{ \sum_{m \in \mathcal{M}} \mathbb{P}(M = m)\mathbf{u}(m) : \mathbf{u}(m) \in \mathcal{C}_m \right\}. \end{aligned} \quad (2.4)$$

The capacity region  $\mathcal{C}$  is a compact, convex, coordinate-convex polyhedron in  $\mathbb{R}_+^{\bar{n}}$  whose exact shape depends on the distribution of  $M$  [7]. Let  $\mathcal{C}^v \equiv \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(\bar{l})}\}$  denote the set of maximal vertices of  $\mathcal{C}$ , where  $\bar{l}$  denotes the number of vertices. For any  $\mathcal{N}' \subseteq \mathcal{N}$ , define  $\mathcal{C}(\mathcal{N}') \equiv \{\mathbf{u} \in \mathcal{C} : u_n = 0, \forall n \notin \mathcal{N}'\}$ , where  $\mathcal{C}(\mathcal{N}')$  is the channel capacity region when the channel is shared only amongst the users in  $\mathcal{N}'$ .

The following (restatement of) Lemma 2.1 from [22] will be used in the subsequent sections.

**Lemma 2.1** *Assume all queues are infinitely backlogged. For any weight vector  $\boldsymbol{\alpha} = (\alpha_n, n \in \mathcal{N}) \geq \mathbf{0}$ , let  $\boldsymbol{\beta}(\boldsymbol{\alpha}) \in \mathcal{C}$  denote the vector of average service rates seen by the queues under the policy which serves user  $n^*$  at time  $t$  if,*

$$n^* \in \arg \max_{n \in \mathcal{N}} \{ \alpha_n r_n(m(t)) \}, \quad (2.5)$$

augmented with a tie breaking rule; then,

$$\boldsymbol{\beta}(\boldsymbol{\alpha}) \in \arg \max_{\mathbf{u} \in \mathcal{C}} \langle \boldsymbol{\alpha}, \mathbf{u} \rangle . \quad (2.6)$$

In words,  $\boldsymbol{\alpha}$  is an outer normal vector to the capacity region  $\mathcal{C}$  at point  $\boldsymbol{\beta}(\boldsymbol{\alpha})$  on the boundary.

As shown in [10], the system of  $\bar{n}$  queues is stabilizable if and only if there exists a vector  $\mathbf{u} = (u_n, n \in \mathcal{N}) \in \mathcal{C}$  such that for all  $n \in \mathcal{N}$ ,

$$\lambda_n < u_n . \quad (2.7)$$

We assume that the system under consideration is stabilizable which implies that the weighted sum defined in (2.3) is bounded under at least one stationary policy.

### 2.3 Characterization of delay-optimal policy

Consider the process  $(\mathbf{Q}(t), t \geq 0)$  initiated in state  $\mathbf{Q}(0) = \mathbf{q}(0)$  and evolving under a policy  $\mathbf{f}$ . Then conditional on the process being in state  $\mathbf{q}$ , the  $n^{\text{th}}$  queue is offered an average service rate of  $\mu_n(\mathbf{q})$  given by,

$$\mu_n(\mathbf{q}) = \mathbb{E} [ r_n(M) f_n(\mathbf{q}, M) ] , \quad (2.8)$$

where the expectation is with respect to  $M$ . By definition of  $\mathcal{C}$  in (2.4), the average service rate vector  $\boldsymbol{\mu}(\mathbf{q}) = (\mu_n(\mathbf{q}), n \in \mathcal{N})$  lies<sup>1</sup> in  $\mathcal{C}$ . We assume that over an epoch, each queue  $n \in \mathcal{N}$  is served constantly at rate  $\mu_n(\mathbf{q})$ , thus the set of  $n$  queues see state-dependent service rates chosen from  $\mathcal{C}$ . A rigorous justification of this can be found in [33] and relies on packet or file dynamics that are slow relative to channel variations, where the latter can be averaged. A similar assumption is made in [22] to obtain processor-sharing queueing model for a slotted time system where a packet (or file) typically takes many slots to process while the channel can change from slot to slot.

Note that strictly speaking, it is shown in [23] that analysis under the assumption of infinitely fast channel variations leads to optimistic flow-level performance estimates.

Under these assumptions, the scheduling problem of finding the right function  $\mathbf{f}(\mathbf{q}, \cdot)$  for each  $\mathbf{q}$  such that the total (weighted) average queue length is minimized (see (2.3)), is one of finding the right service rate vector  $\boldsymbol{\mu}(\mathbf{q}) \in \mathcal{C}$  for each  $\mathbf{q}$ . Using this, we re-define a scheduling policy as a function  $\boldsymbol{\mu} : \mathbb{Z}_+^{\bar{n}} \rightarrow \mathcal{C}$  that takes a queue state vector in  $\mathbb{Z}_+^{\bar{n}}$  to a service rate vector in  $\mathcal{C}$ , where  $\boldsymbol{\mu}$  relates to  $\mathbf{f}$  through (2.8).

Under a fixed policy  $\boldsymbol{\mu}$ , the process  $(\mathbf{Q}(t), t \geq 0)$  forms a time-homogeneous Markov chain on  $\mathbb{Z}_+^{\bar{n}}$  with state-dependent transition rates. For convenience, we shall uniformize  $\mathbf{Q}(t)$ . For any  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ , let  $\mathbf{A}_n \mathbf{q} \equiv \mathbf{q} + \mathbf{e}_n$  and  $\mathbf{D}_n \mathbf{q} \equiv (\mathbf{q} - \mathbf{e}_n)^+$ , where  $\mathbf{q}^+ \equiv (\mathbf{y} : y_n = \max\{0, q_n\})$ . Let  $\gamma \equiv |\boldsymbol{\lambda}| + \max_{\mathbf{u} \in \mathcal{C}} |\mathbf{u}|$ , where  $|\cdot|$  denotes  $L_1$  norm. Let  $\tau_k$  denote the (random) time of the  $k^{\text{th}}$  transition of  $\mathbf{Q}(t)$  and  $\tau_0 = 0$ . Also, let  $\mathbf{Q}_k = \lim_{t \downarrow \tau_k} \mathbf{Q}(t)$ . Then under policy  $\boldsymbol{\mu}$ , the process  $\mathbf{Q}(t)$  can be viewed as having a state-independent event rate of  $\gamma$  (*i.e.*  $(\tau_{k+1} - \tau_k) \sim \exp(\gamma)$ ) and transition probabilities given by, for all  $n \in \mathcal{N}$ ,

$$\begin{aligned} P(\mathbf{Q}_{k+1} = \mathbf{A}_n \mathbf{q} | \mathbf{Q}_k = \mathbf{q}) &= \frac{\lambda_n}{\gamma}, \\ P(\mathbf{Q}_{k+1} = \mathbf{D}_n \mathbf{q} | \mathbf{Q}_k = \mathbf{q}) &= \frac{\mu_n(\mathbf{q})}{\gamma}, \\ P(\mathbf{Q}_{k+1} = \mathbf{q} | \mathbf{Q}_k = \mathbf{q}) &= 1 - \frac{|\boldsymbol{\lambda}| + |\boldsymbol{\mu}(\mathbf{q})|}{\gamma}. \end{aligned} \tag{2.9}$$

Define the cost under policy  $\boldsymbol{\mu}$  over  $[0, \tau_k)$  when starting in state  $\mathbf{q}$  as

$$\mathbb{E}_{\mathbf{q}}^{\boldsymbol{\mu}} \left[ \int_0^{\tau_k} \langle \mathbf{w}, \mathbf{Q}(t) \rangle dt \right],$$

which, ignoring a constant multiplier  $\gamma^{-1}$ , can be shown to be equal to,

$$V_k^{\boldsymbol{\mu}}(\mathbf{q}) \equiv \mathbb{E}_{\mathbf{q}}^{\boldsymbol{\mu}} \left[ \sum_{l=0}^{k-1} \langle \mathbf{w}, \mathbf{Q}_l \rangle \right]. \tag{2.10}$$

---

<sup>1</sup>The map  $\mathbf{f}(\mathbf{q}, \cdot) \rightarrow \boldsymbol{\mu}(\mathbf{q}) \in \mathcal{C}$  given by (2.8) is surjective.

Likewise, the average cost under policy  $\boldsymbol{\mu}$ , when starting in state  $\mathbf{q}$ , is given by,

$$J_{\boldsymbol{\mu}}(\mathbf{q}) = \limsup_{k \rightarrow \infty} \frac{1}{k} V_k^{\boldsymbol{\mu}}(\mathbf{q}) . \quad (2.11)$$

The optimality criterion given in (2.3) seeks to minimize this average cost. The problem of finding the minimum average cost and an optimal policy fits the classical dynamic programming framework (*e.g.* see [34]). Thus the minimum average cost over all policies (denoted by  $J^*$ ) is well defined, independent of the starting state, and together with a relative cost function  $h : \mathbb{Z}_+^{\bar{n}} \rightarrow [0, \infty)$ , which is unique up to an additive constant, satisfies Bellman's equation, *i.e.*, for all  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ ,

$$\begin{aligned} J^* &= \min_{\boldsymbol{\mu}} \left\{ \langle \mathbf{w}, \mathbf{q} \rangle + \mathbb{E}^{\boldsymbol{\mu}} \left[ h(\mathbf{Q}_{k+1}) - h(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q} \right] \right\} , \\ &= \min_{\mathbf{u} \in \mathcal{C}} \left\{ \langle \mathbf{w}, \mathbf{q} \rangle + \sum_{n \in \mathcal{N}} \frac{\lambda_n}{\gamma} (h(\mathbf{A}_n \mathbf{q}) - h(\mathbf{q})) + \right. \\ &\quad \left. \frac{u_n}{\gamma} (h(\mathbf{D}_n \mathbf{q}) - h(\mathbf{q})) \right\} . \end{aligned} \quad (2.12)$$

Moreover, let  $\Delta h(\mathbf{q}) \equiv (h(\mathbf{q}) - h(\mathbf{D}_n \mathbf{q}), n \in \mathcal{N})$ , and define  $\boldsymbol{\mu}^*$  as a policy that achieves the minimum in (2.12) for every  $\mathbf{q}$ , *i.e.*,

$$\boldsymbol{\mu}^*(\mathbf{q}) \in \arg \max_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{u}, \Delta h(\mathbf{q}) \rangle, \quad (2.13)$$

then  $\boldsymbol{\mu}^*$  is an optimal policy achieving the minimum average cost  $J^*$ . The following lemma characterizes optimal scheduling decisions in *time* (see (2.1)), and follows from Lemma 2.1 by interpreting  $\Delta h(\mathbf{q})$  and  $\boldsymbol{\mu}^*(\mathbf{q})$  in (2.13) as  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  respectively.

**Lemma 2.2** *The following policy achieves the minimum average cost (see (2.3) and (2.11)): at any time when the system is in state  $(\mathbf{Q}, M) = (\mathbf{q}, m)$ , choose a stochastic vector  $\boldsymbol{\sigma}$  that satisfies*

$$\sigma_{n^*} = 1 \text{ for some } n^* \in \arg \max_{n \in \mathcal{N}} \{ \Delta_n h(\mathbf{q}) r_n(m) \}, \quad (2.14)$$

where  $h$  is a relative cost function satisfying Bellman's equation (2.12).

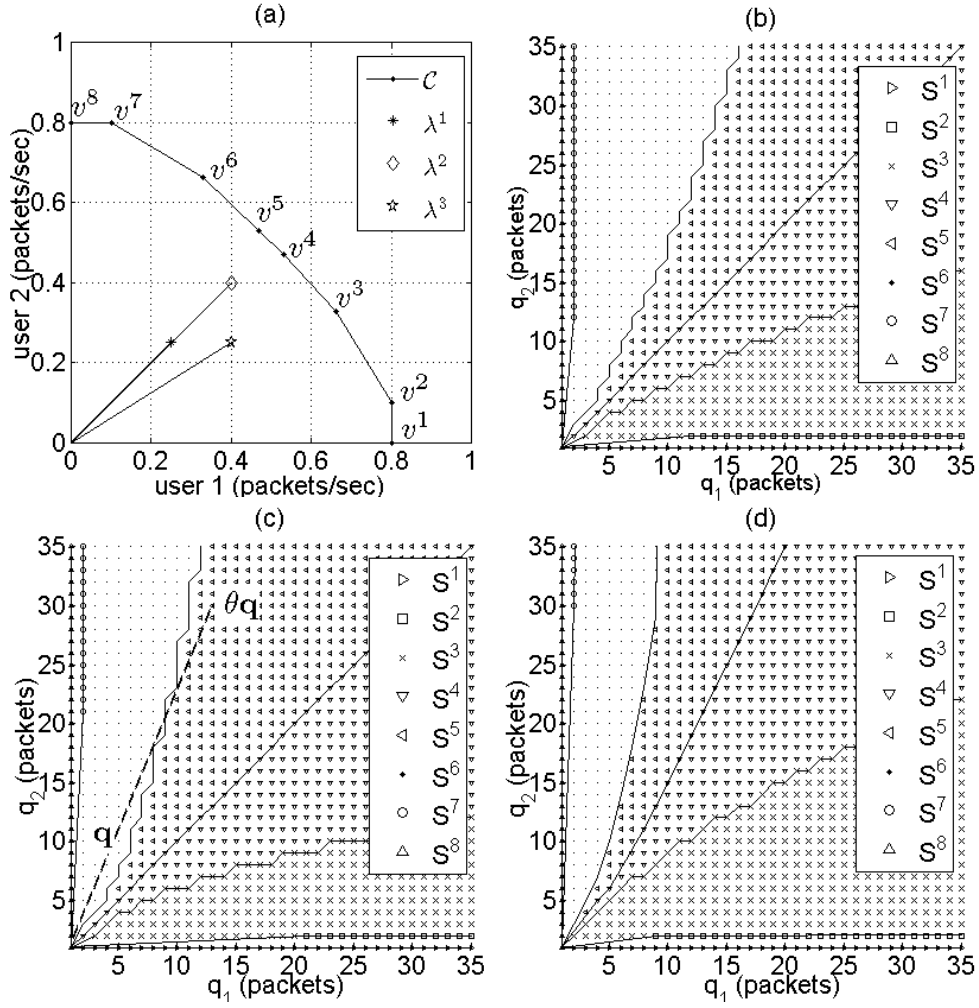


Figure 2.1: Partitions under the optimal policy: (a) 2-user capacity region. Remaining three figures show partitions corresponding to (b) arrival vector  $\lambda^1 = (0.25, 0.25)$  (c) arrival vector  $\lambda^2 = (0.4, 0.4)$  (d) arrival vector  $\lambda^3 = (0.4, 0.25)$ .

Lemma 2.2 and (2.13) relate the tradeoff mentioned in Chapter 1 to the geometry of vector field  $\Delta h$  associated with the relative cost function. We explore this tradeoff in the next section where we use relative value iteration (see, *e.g.*, [35]) to numerically compute  $h$  and  $J^*$ .

## 2.4 Radial sum-rate monotonicity: comparing the optimal policy with known heuristics

In this section, we investigate how delay optimal schedulers, as well as throughput optimal policies such as MaxWeight and Exp rule, tradeoff current service rate versus balancing unequal queues. Specifically, we consider how the service rate vector chosen by each policy changes as the queues grow proportionally from a state  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$  to a state  $\theta\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$  for  $\theta > 1$ . Note that  $\mathbf{q}$  and  $\theta\mathbf{q}$  lie on a *radial* line in  $\mathbb{R}_+^{\bar{n}}$  that passes through origin. For any  $\mathbf{q}$ , let  $\mathcal{N}_{\mathbf{q}} \equiv \{n \in \mathcal{N} : q_n \neq 0\}$ , *i.e.*, the set of non-empty queues. We begin by defining an interesting property which we refer to as *radial sum-rate monotonicity*, as well as a weaker version of this property.

**Definition 2.1** *Given a weight vector  $\mathbf{w} > \mathbf{0}$ , we say a scheduling policy  $\boldsymbol{\mu}$  is radial sum-rate monotone with respect to vector  $\mathbf{w}$  if it satisfies two conditions. For any  $\mathbf{q}$  and scalar  $\theta$  such that  $\theta\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ :*

- (a) *the total weighted service rate,  $\langle \mathbf{w}, \boldsymbol{\mu}(\theta\mathbf{q}) \rangle$ , is an increasing function of  $\theta$ ,*
- (b)  $\lim_{\theta \rightarrow \infty} \langle \mathbf{w}, \boldsymbol{\mu}(\theta\mathbf{q}) \rangle = \max_{\mathbf{u}} (\langle \mathbf{w}, \mathbf{u} \rangle : \mathbf{u} \in \mathcal{C}(\mathcal{N}_{\mathbf{q}}))$ .

*Moreover, we say that  $\boldsymbol{\mu}$  is weakly RSM if it satisfies (a).*

Hence, as the queue grows proportionally, a radial sum-rate monotone (RSM) policy allocates service rates in a manner that *de-emphasizes queue-balancing* in favor of *increasing the total weighted service rate* (with respect to weight vector  $\mathbf{w}$ ). Another useful and natural property, called *transition monotonicity* [36], describes the behavior of a policy along an *axial* line, *i.e.*, as a queue grows from state  $\mathbf{q}$  to a state  $\mathbf{q} + \theta\mathbf{e}_n$  for any integer  $\theta > 0$  and  $n \in \mathcal{N}$ .

**Definition 2.2** *A scheduling policy  $\boldsymbol{\mu}$  is transition monotone if for all  $n \in \mathcal{N}$  and  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ , we have<sup>2</sup>  $\mu_n(\mathbf{q}) \leq \mu_n(\mathbf{q} + \mathbf{e}_n)$ , and  $\lim_{\theta \rightarrow \infty} \mu_n(\mathbf{q} + \theta\mathbf{e}_n) = \max_{\mathbf{u} \in \mathcal{C}} u_n$ .*



Hence, as a single queue grows while others remain unchanged, a transition monotone policy allocates more service to the growing queue, and in the limit, only the longest queue is scheduled (whenever it sees a non-zero channel). As a result, asymptotically along an axial line, the total weighted service rate decreases.

**Remark 2.1** *Radial sum-rate and transition monotonicities both describe the above-mentioned tradeoff as  $\mathbf{q}$  is taken to  $\infty$ , however along different paths. Indeed, a policy can be both radial sum-rate and transition monotone, in which case, the total weighted service rate will increase along radial lines, however (asymptotically) decrease along axial lines. Moreover, it is simple to show that a throughput-optimal policy must be transition monotone except possibly on a compact subset of queue state space; see, e.g., [37].*

#### 2.4.1 The tradeoff under delay-optimal schedulers

Since the capacity region  $\mathcal{C}$  is a polyhedron, instead of searching over the entire region for the maximum in (2.13), it suffices to only consider the vertices of  $\mathcal{C}$ ,

$$\boldsymbol{\mu}^*(\mathbf{q}) \in \arg \max_{\mathbf{u} \in \mathcal{C}^v} \langle \mathbf{u}, \Delta h(\mathbf{q}) \rangle . \quad (2.15)$$

Hence, the optimal policy partitions the state space  $\mathbb{Z}_+^{\bar{n}}$  into at most  $\bar{l}$  non-empty decision regions  $\mathcal{S}^1, \mathcal{S}^2, \dots, \mathcal{S}^{\bar{l}}$ , each of which is associated with a distinct vertex, *i.e.*,

$$\mathcal{S}^l \equiv \{\mathbf{q} : \boldsymbol{\mu}^*(\mathbf{q}) = \mathbf{v}^{(l)}\}.$$

In each region  $\mathcal{S}^l$ , the scheduler tries to *push* the queue process  $\mathbf{Q}_k$  along vector  $\boldsymbol{\lambda} - \mathbf{v}^{(l)}$ . Fig. 2.1 shows the optimal policy's partitioning of  $\mathbb{Z}_+^{\bar{n}}$  for a two user system with weight vector  $\mathbf{w} = (1, 1)$  under three different arrival vectors. The first plot shows the hypothetical 2-user capacity region and arrival vectors considered. The second

---

<sup>2</sup>By coordinate-convexity of  $\mathcal{C}$ , it follows that for all  $n' \neq n$ , we have  $\mu_{n'}(\mathbf{q}) \geq \mu_{n'}(\mathbf{A}_n \mathbf{q})$ .

plot depicts the partition for  $\lambda = (0.25, 0.25)$  packets/sec. The third plot exhibits a more pronounced radial sum-rate monotonicity when arrival rate is increased to  $\lambda = (0.4, 0.4)$  packets/sec, and the last plot is intended to exhibit the warping effect on the partition resulting from asymmetric arrival rates  $\lambda = (0.4, 0.25)$  packets/sec. The boundaries between decision region are referred to as the *switching curves*.

For the optimal policy to be RSM we must have that as  $\theta \rightarrow \infty$  such that  $\theta \mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$  and  $\mathbf{q} > 0$ , the ratio  $\frac{\Delta_{n'} h(\theta \mathbf{q})}{\Delta_n h(\theta \mathbf{q})}$  monotonically converges to  $\frac{w_{n'}}{w_n}$ , and therefore,

$$\lim_{\theta \rightarrow \infty} \Delta h(\theta \mathbf{q}) \propto (w_n \mathbf{1}_{\{q_n > 0\}}, n \in \mathcal{N}) . \quad (2.16)$$

Whereas, for the optimal policy to be *weakly* RSM, we only need that as  $\theta$  increases, the ratio  $\frac{\Delta_{n'} h(\theta \mathbf{q})}{\Delta_n h(\theta \mathbf{q})}$  monotonically gets closer to  $\frac{w_{n'}}{w_n}$  (but is not required to converge to  $\frac{w_{n'}}{w_n}$ ).

By computing the relative cost function  $h$  and the optimal scheduling policy for various arrival rate vectors and capacity regions, we observe that the optimal policies exhibit weak radial sum-rate monotonicity. An intuitive explanation of why the delay-optimal policies exhibit weak radial sum-rate monotonicity can be based on the following two observations:

1. the cost incurred per unit time in state  $\theta \mathbf{q}$  for  $\theta > 1$  is more than the cost incurred per unit time in state  $\mathbf{q}$ ;
2. the state  $\theta \mathbf{q}$  is *farther* from any axis than the state  $\mathbf{q}$ .

Both observations suggest that in state  $\theta \mathbf{q}$ , an optimal policy would indeed de-emphasize queue balancing in favor of increasing the current total weighted service rate.

To verify if the optimal policy is RSM, we need the asymptotics of the relative cost function  $h(\theta \mathbf{q})$  for large  $\theta$  (see (2.16)), which we cannot *compute* through

relative value iteration. One can however use deterministic *fluid models* to obtain<sup>3</sup>  $\lim_{\theta \rightarrow \infty} \frac{h(\theta \mathbf{q})}{\theta^2}$ , see [38, Theorem 10.0.5]. Such limits can be used to determine the asymptotic slope of the switching curves on the state space of the *fluid-scaled* queue process. For details, see Appendix A, where we solve the fluid models of some non-trivial systems and show that, in general, the optimal policy may not be RSM. However, for a symmetric system subject to sufficient load, RSM policies are *fluid-scale asymptotic optimal* and therefore, RSM policies and optimal policies have similar switching curves on the state space of the fluid-scaled queue process.

### Weighted max-rate horn

Consider the decision regions  $\mathcal{S}^4$  and  $\mathcal{S}^5$ , *i.e.*, regions corresponding to those vertices of  $\mathcal{C}$  which have the largest projection along vector  $\mathbf{w}$ . Under the optimal policy, union of these decision regions is shaped like a French horn (referred to as weighted max-rate horn). As we shall see next, under the Exp rule (with appropriately chosen constants), the union of the same partitions is shaped like a cylinder with gradually increasing diameter, whereas, under MaxWeight, all partitions are simply cones.

#### 2.4.2 The tradeoffs under MaxWeight and Exp Rule

The MaxWeight and the Exp rule policies can also be expressed in a similar form as (2.13). These policies replace  $\Delta h$  with a suitable vector field on  $\mathbb{Z}_+^{\bar{n}}$  such that the system is stable for any stabilizable  $\lambda$ . Hence the tradeoff under each policy can be investigated by considering how the vector fields change direction as queues grow proportionally.

MaxWeight policies [10] can be defined as follows: when the system is in state

---

<sup>3</sup>Function  $h$  exhibits quadratic growth [38, Theorem 9.0.5], therefore the limit is meaningful.

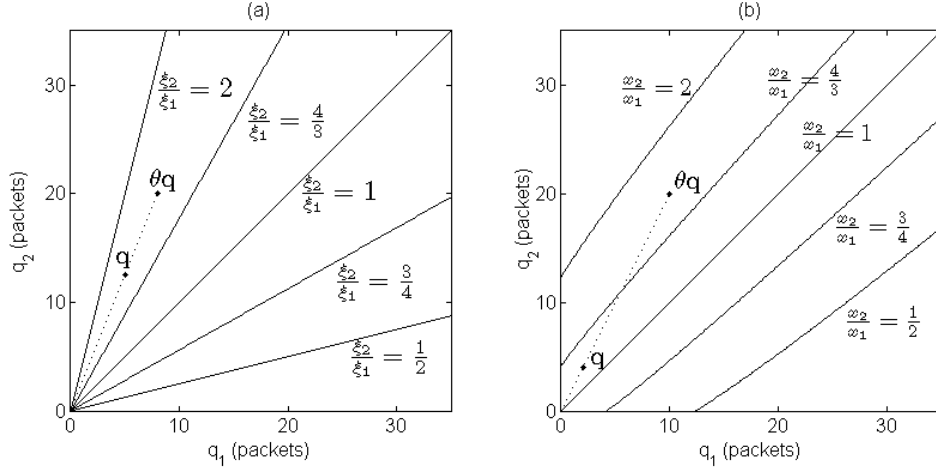


Figure 2.2: Curves along which the direction is held constant by the vector field (a)  $\xi(\cdot)$  with  $b_n = 1 \forall n \in \mathcal{N}$ ,  $\alpha = 0.5$ , (b)  $\mathbf{x}(\cdot)$  with  $a_n = 0.1$ ,  $b_n = 1 \forall n \in \mathcal{N}$ ,  $c = 1$ ,  $\eta = 0.5$

$(\mathbf{Q}, M) = (\mathbf{q}, m)$ , choose a stochastic vector  $\sigma$  that satisfies

$$\sigma_{n^*} = 1 \text{ for some } n^* \in \arg \max_{n \in \mathcal{N}} \{ \xi_n(\mathbf{q}) r_n(m) \},$$

where  $\xi_n(\mathbf{q})$  is the  $n^{\text{th}}$  component of  $\xi(\mathbf{q}) \equiv (b_n q_n^\alpha, n \in \mathcal{N})$ , for any fixed positive  $b_n$ 's and  $\alpha$ . Equivalently, when the queue state is  $\mathbf{q}$ , the policy uses a service rate vector  $\mu^W(\mathbf{q})$  given by,

$$\mu^W(\mathbf{q}) \in \arg \max_{\mathbf{u} \in \mathcal{C}^v} \langle \mathbf{u}, \xi(\mathbf{q}) \rangle. \quad (2.17)$$

Similarly, the Exp rule [11] is given by,

$$\mu^X(\mathbf{q}) \in \arg \max_{\mathbf{u} \in \mathcal{C}^v} \langle \mathbf{u}, \mathbf{x}(\mathbf{q}) \rangle, \quad (2.18)$$

where,

$$\mathbf{x}(\mathbf{q}) \equiv \left( b_n \exp \left( \frac{a_n q_n}{c + (\bar{n}^{-1} \sum_{j \in \mathcal{N}} a_j q_j)^\eta} \right), n \in \mathcal{N} \right),$$

for any fixed positive  $a_n$ 's,  $b_n$ 's,  $c$ , and  $0 < \eta < 1$ .

While both the MaxWeight and the Exp rule are transition monotone, neither is radially sum-rate monotone. For  $n = 2$  and extending the domain of  $\xi$  and  $\mathbf{x}$  to

$\mathbb{R}_+^{\bar{n}}$ , Fig. 2.2 shows the curves in  $\mathbb{R}_+^2$  along which the vector fields  $\boldsymbol{\xi}$  and  $\boldsymbol{x}$  hold their direction, *i.e.*,

$$\left\{ \mathbf{q} : \frac{\xi_2(\mathbf{q})}{\xi_1(\mathbf{q})} = \text{“constant”} \right\} \quad \text{and} \quad \left\{ \mathbf{q} : \frac{x_2(\mathbf{q})}{x_1(\mathbf{q})} = \text{“constant”} \right\}$$

for various values of “constant”. Curves like these form the boundaries of the decision regions, *i.e.*, the switching curves. The vector field  $\boldsymbol{\xi}$  is homogeneous, hence the service rate allocation under MaxWeight is invariant as the queues grow from state  $\mathbf{q}$  to state  $\theta\mathbf{q}$ . By contrast, in the case of the Exp rule (with  $\mathbf{b}$  set to  $\mathbf{w}$ ), the total weighted service rate  $\langle \mathbf{w}, \boldsymbol{\mu}^X(\theta\mathbf{q}) \rangle$  decreases with  $\theta$  and the emphasis shifts to queue-balancing, so much so that as  $\theta \rightarrow \infty$ , only the longest weighted queue(s) receives service.

## 2.5 Improved throughput-optimal policies

We begin this section with a sufficiency theorem regarding throughput-optimal policies.

### 2.5.1 Sufficient conditions for throughput-optimality

**Theorem 2.1** *Let  $\mathbf{g} : \mathbb{R}_+^{\bar{n}} \rightarrow \mathbb{R}_+^{\bar{n}}$  be a gradient field (i.e.  $\mathbf{g} = \nabla G$  for some  $G : \mathbb{R}_+^{\bar{n}} \rightarrow \mathbb{R}$ ).*

*Moreover, suppose  $\mathbf{g}$  is differentiable on  $\mathbb{R}_+^{\bar{n}}$  and for all  $n \in \mathcal{N}$ , and satisfies,*

$$\lim_{\mathbf{y} \rightarrow \infty: y_n=0} \frac{g_n(\mathbf{y})}{|\mathbf{g}(\mathbf{y})|} = 0, \quad (2.19)$$

$$\lim_{\mathbf{y} \rightarrow \infty} \frac{\partial g_n(\mathbf{y}) / \partial y_n}{|\mathbf{g}(\mathbf{y})|} = 0, \quad (2.20)$$

*and for some  $\epsilon > 0$ ,  $|\mathbf{g}(\mathbf{y})| > \epsilon$  for all  $\mathbf{y}$  outside a compact subset of  $\mathbb{R}_+^{\bar{n}}$ , then any policy  $\hat{\boldsymbol{\mu}}$  satisfying the following is throughput-optimal: for all  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ ,*

$$\hat{\boldsymbol{\mu}}(\mathbf{q}) \in \arg \max_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{u}, \mathbf{g}(\mathbf{q}) \rangle. \quad (2.21)$$

**Remark 2.2** *The condition that  $\mathbf{g}$  be a gradient field and  $|\mathbf{g}(\mathbf{y})| > \epsilon$  outside a compact set, is used to establish the existence of a potential (Lyapunov) function  $G$  such*

that  $\nabla G = \mathbf{g}$ . Condition (2.19) is needed to ensure that when queue state vector  $\mathbf{q}$  is large, the policy given by  $\mathbf{g}$  is work-conserving, i.e., it does not allocate any service rate to an empty queue at the cost of non-empty queues. Condition (2.20) is used to ensure that the Hessian of  $G$  can be dominated by its gradient in the Taylor expansion (see proof). See Appendix 2.5.1 for proof. See [37] for a similar result with a slightly different system model. [37] improves upon the above result by not requiring  $\mathbf{g}$  to be a gradient field.

**Examples** Examples of functions that satisfy the conditions of Theorem 2.1 are  $\xi(\cdot)$  with its domain extended to  $\mathbb{R}_+^{\bar{n}}$ ,  $\mathbf{g}(\mathbf{y}) = (\exp(y_n^\alpha), n \in \mathcal{N})$  for  $\alpha \in (0, 1)$ , and  $\mathbf{g}(\mathbf{y}) = (\log(1 + \log(1 + y_n)), n \in \mathcal{N})$ , indicating that a throughput-optimal policy can exhibit anywhere from sub-logarithmical to almost exponential sensitivity to changes in queue lengths. Another interesting example is,

$$\mathbf{g}(\mathbf{y}) = (y_n^{\alpha_1}(y_n + c)^{\alpha_2}, n \in \mathcal{N}),$$

for  $\alpha_1 \geq \alpha_2 > 0$  and  $c \geq 0$ , which behave as MaxWeight with exponent  $\alpha_1$  near the origin in  $\mathbb{Z}_+^{\bar{n}}$  and as MaxWeight with exponent  $\alpha_2$  radially far from the origin.

### Proof of Theorem 2.1

We will use Foster's criterion to show that  $(\mathbf{Q}_k, k \geq 0)$  is positive recurrent for any stabilizable  $\lambda$  (see (2.7)). Specifically, take a (Lyapunov) function  $G : \mathbb{R}_+^{\bar{n}} \rightarrow \mathbb{R}_+$  such that  $\nabla G = \mathbf{g}$  and  $G(\mathbf{0}) = 0$ . Then,

$$\begin{aligned} & \mathbb{E}^{\hat{\mu}}[G(\mathbf{Q}_{k+1}) - G(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q}] \\ &= \gamma^{-1} \sum_{n \in \mathcal{N}} \lambda_n (G(\mathbf{A}_n \mathbf{q}) - G(\mathbf{q})) + \gamma^{-1} \sum_{n \in \mathcal{N}_q} \hat{\mu}_n(\mathbf{q}) (G(\mathbf{D}_n \mathbf{q}) - G(\mathbf{q})), \end{aligned} \tag{2.22}$$

where, as before,  $\mathcal{N}_{\mathbf{q}} = \{n : q_n \neq 0\}$ . Since  $\mathbf{g}$  is differentiable, let  $\dot{g}_n(\cdot) = \partial g_n(\cdot)/\partial x_n$  for all  $n \in \mathcal{N}$ . Then  $G$  has the following Taylor expansion for any  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ ,

$$\begin{aligned} G(\mathbf{A}_n \mathbf{q}) - G(\mathbf{q}) &= g_n(\mathbf{q}) + \frac{1}{2} \dot{g}_n(\mathbf{q} + \alpha_n \mathbf{e}_n) \quad \forall n \in \mathcal{N}, \\ G(\mathbf{D}_n \mathbf{q}) - G(\mathbf{q}) &= -g_n(\mathbf{q}) + \frac{1}{2} \dot{g}_n(\mathbf{q} - \beta_n \mathbf{e}_n) \quad \forall n \in \mathcal{N}_{\mathbf{q}}, \end{aligned}$$

for some  $\alpha_n \in [0, 1]$  and  $\beta_n \in [0, 1]$  that depend on  $\mathbf{q}$ . One can rewrite (2.22) as follows,

$$\begin{aligned} &\mathbb{E}^{\hat{\mu}}[G(\mathbf{Q}_{k+1}) - G(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q}] \\ &= \gamma^{-1} \left( \sum_{n \in \mathcal{N}} \lambda_n g_n(\mathbf{q}) + \frac{1}{2} \sum_{n \in \mathcal{N}} \lambda_n \dot{g}_n(\mathbf{q} + \alpha_n \mathbf{e}_n) - \right. \\ &\quad \left. \sum_{n \in \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) g_n(\mathbf{q}) + \frac{1}{2} \sum_{n \in \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) \dot{g}_n(\mathbf{q} - \beta_n \mathbf{e}_n) \right). \end{aligned}$$

Adding and subtracting  $\gamma^{-1} \sum_{n \in \mathcal{N} \setminus \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) g_n(\mathbf{q})$  from the left side of above yields,

$$\begin{aligned} &\mathbb{E}^{\hat{\mu}}[G(\mathbf{Q}_{k+1}) - G(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q}] \\ &= \gamma^{-1} \langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\mathbf{q}), \mathbf{g}(\mathbf{q}) \rangle + \gamma^{-1} \sum_{n \in \mathcal{N} \setminus \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) g_n(\mathbf{q}) + \\ &\quad \frac{\gamma^{-1}}{2} \sum_{n \in \mathcal{N}} \lambda_n \dot{g}_n(\mathbf{q} + \alpha_n \mathbf{e}_n) + \frac{\gamma^{-1}}{2} \sum_{n \in \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) \dot{g}_n(\mathbf{q} - \beta_n \mathbf{e}_n) \\ &\leq \gamma^{-1} \langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\mathbf{q}), \mathbf{g}(\mathbf{q}) \rangle + \gamma^{-1} \sum_{n \in \mathcal{N} \setminus \mathcal{N}_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) g_n(\mathbf{q}) + \\ &\quad \max \left\{ \max_n \{ \dot{g}_n(\mathbf{q} + \alpha_n \mathbf{e}_n) : n \in \mathcal{N} \}, \right. \\ &\quad \left. \max_n \{ \dot{g}_n(\mathbf{q} - \beta_n \mathbf{e}_n) : n \in \mathcal{N}_{\mathbf{q}} \} \right\}. \end{aligned} \tag{2.23}$$

Let  $\mathbf{u} \in \mathcal{C}$  be a service rate vector that satisfies  $\boldsymbol{\lambda} < \mathbf{u}$ . Let  $\epsilon_1 > 0$  be given by  $\epsilon_1 = \gamma^{-1} \min_{n \in \mathcal{N}} \{(u_n - \lambda_n)\}$ . Moreover, by Lemma 2.1, for all  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ ,

$$\langle \boldsymbol{\lambda} - \hat{\boldsymbol{\mu}}(\mathbf{q}), \mathbf{g}(\mathbf{q}) \rangle \leq \langle \boldsymbol{\lambda} - \mathbf{u}, \mathbf{g}(\mathbf{q}) \rangle \leq -\epsilon_1 \gamma |\mathbf{g}(\mathbf{q})|$$

Substituting in (2.23),

$$\mathbb{E}^{\hat{\mu}}[G(\mathbf{Q}_{k+1}) - G(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q}]$$

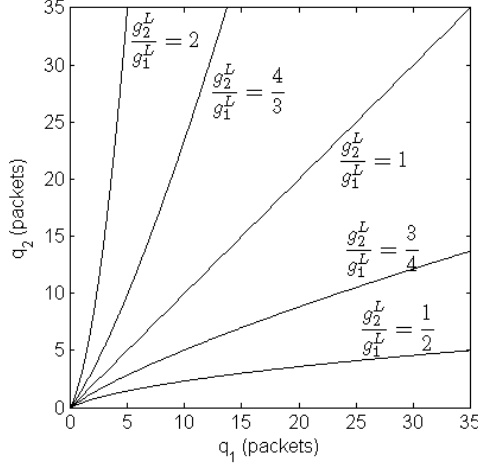


Figure 2.3: Curves along which the direction is held constant by  $g^L(\cdot)$  with  $a_n = 1$ ,  $b_n = 1 \forall n \in \mathcal{N}$ ,  $c = 1$ .

$$\begin{aligned}
&\leq -\epsilon_1 |\mathbf{g}(\mathbf{q})| + \gamma^{-1} \sum_{n \in \mathcal{N} \setminus I_{\mathbf{q}}} \hat{\mu}_n(\mathbf{q}) g_n(\mathbf{q}) + \\
&\quad \max \left\{ \max_n \{ \dot{g}_n(\mathbf{q} + \alpha_n \mathbf{e}_n) : n \in \mathcal{N} \}, \right. \\
&\quad \left. \max_n \{ \dot{g}_n(\mathbf{q} - \beta_n \mathbf{e}_n) : n \in \mathcal{N}_{\mathbf{q}} \} \right\}. \tag{2.24}
\end{aligned}$$

By using (2.19), when  $\mathbf{q}$  is suitably large,  $g_n(\mathbf{q})$  for each  $n \in \mathcal{N} \setminus \mathcal{N}_{\mathbf{q}}$  in the second term of the above summation can be bounded above by  $\frac{\epsilon_1}{4} |\mathbf{g}(\mathbf{q})|$ . Similarly, using (2.20), the third term of the above summation can be bounded above by  $\frac{\epsilon_1}{4} |\mathbf{g}(\mathbf{q})|$ . Hence, for  $\mathbf{q}$  large enough, (2.24) becomes,

$$\mathbb{E}^{\hat{\mu}} [ G(\mathbf{Q}_{k+1}) - G(\mathbf{Q}_k) \mid \mathbf{Q}_k = \mathbf{q} ] \leq -\frac{\epsilon_1}{2} |\mathbf{g}(\mathbf{q})|$$

Since  $0 < \epsilon < |\mathbf{g}(\mathbf{q})|$  for all large  $\mathbf{q}$ , the proof is complete.  $\blacksquare$

### 2.5.2 The Log Rule

In this section we consider a class of schedulers satisfying Theorem 2.1, which we refer to as the Log rule.

**Definition 2.3** Arbitrarily fix  $\mathbf{a} = (a_n, n \in \mathcal{N}) > 0$ ,  $\mathbf{b} = (b_n, n \in \mathcal{N}) > 0$ , and  $c \geq 1$ . For all  $\mathbf{y} \in \mathbb{R}_+^{\bar{n}}$ , let  $\mathbf{g}^L(\mathbf{y}) = (g_n^L(\mathbf{y}), n \in \mathcal{N})$ , where  $g_n^L(\mathbf{y}) = b_n \log(c + a_n y_n)$ .



When the system is in state  $(\mathbf{Q}, M) = (\mathbf{q}, m)$  (see (2.1)), choose a stochastic vector  $\boldsymbol{\sigma}$  that satisfies,

$$\sigma_{n^*} = 1 \text{ for some } n^* \in \arg \max_{n \in \mathcal{N}} \{ g_n^L(\mathbf{q}) r_n(m) \} . \quad (2.25)$$

**Theorem 2.2** *The Log rule is radial sum-rate monotone w.r.t. weight vector  $\mathbf{b}$  (with  $c = 1$ ) and throughput-optimal.*

**Proof** First the throughput optimality: let  $\boldsymbol{\mu}^L(\mathbf{q}) = (\mu_n^L(\mathbf{q}), n \in \mathcal{N})$  denote the vector of service rates (under the Log-Rule) seen by the queues when  $\mathbf{Q}(t) = \mathbf{q}$ , *i.e.*,

$$\mu_n^L(\mathbf{q}) = \mathbb{E} [ r_n(M) f_n^L(\mathbf{q}, M) ] , \quad (2.26)$$

where  $\mathbf{f}^L(\mathbf{q}, m)$  denotes the stochastic vector chosen by the Log-Rule (*i.e.* satisfying (2.25)) in state  $(\mathbf{q}, m)$ . By (2.25) and Lemma 2.1, we have

$$\langle \mathbf{g}^L(\mathbf{q}), \boldsymbol{\mu}^L(\mathbf{q}) \rangle = \max_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{g}^L(\mathbf{q}), \mathbf{u} \rangle .$$

Noting that the functions  $\mathbf{g}^L$  and  $\boldsymbol{\mu}^L$  satisfy the conditions of Theorem 2.1, the throughput-optimality of Log rule follows. To verify the radial sum-rate monotonicity of the Log rule we note that for any  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$  such that  $0 < a_n q_n < a_{n'} q_{n'}$ , we have  $\frac{g_n^L(\theta \mathbf{q})}{g_{n'}^L(\theta \mathbf{q})} \uparrow \frac{b_n}{b_{n'}}$  as  $\theta \rightarrow \infty$ . ■

With  $c = 1$  and extending the domain of  $\mathbf{g}^L$  to  $\mathbb{R}_+^{\bar{n}}$ , we note that as  $\theta \rightarrow 0$ , we have  $g_n^L(\theta \mathbf{q})/g_{n'}^L(\theta \mathbf{q}) \rightarrow a_n q_n/a_{n'} q_{n'}$ , *i.e.*, close to origin in  $\mathbb{Z}_+^{\bar{n}}$ , the Log rule behaves similar to the MaxWeight with  $\alpha = 1$ , whereas, radially far away from origin (as  $\theta \rightarrow \infty$ ),  $\mathbf{g}^L(\theta \mathbf{q})$  becomes parallel to the vector  $(b_n \mathbf{1}_{\{q_n > 0\}}, n \in \mathcal{N})$  and thus the Log rule ignores queue-balancing in favor of maximizing the total weighted service rate,  $\langle \mathbf{b}, \boldsymbol{\mu}^L(\theta \mathbf{q}) \rangle$ .

Fig. 2.3 shows the curves along which the direction of the gradient field  $\mathbf{g}^L$  is constant; curves like these form the switching curves and define partition of the

queue state-space into decision regions. A good choice for  $w_n$  (hence  $b_n$ ) is  $1/\mathbb{E}[R_n]$ , as suggested for the Exp rule in [39]. The line  $\{\mathbf{q} \in \mathbb{Z}_+^{\bar{n}} : a_n q_n = a_{n'} q_{n'} \forall n, n' \in \mathcal{N}\}$  defines the axis of the weighted max-rate horn, whereas, the magnitude of the vector  $\mathbf{a}$  controls the width of the horn. Increasing the magnitude of  $\mathbf{a}$  widens the horn and reduces the emphasis of Log rule on balancing user queues (this is opposite to the role this parameter plays in the Exp rule). By choosing  $c > 1$ , the Log rule can be made to behave similar to the Exp rule, instead of MaxWeight with  $\alpha = 1$ , near the origin in  $\mathbb{Z}_+^{\bar{n}}$ .

### Asymptotic probability of sum-queue overflow under the Log rule

The proof of Log rule's asymptotic optimality is given in the next chapter. By leveraging the refined sample path large deviations principle, recently introduced in [19] to study non-homogenous schedulers such as the Exp and the Log rules, we are able to show that for a  $\bar{n} = 2$  user system, a Log-rule-like radial sum-rate monotone policy (w.r.t. a given weight vector  $(w_1, w_2)$ ) indeed minimizes the asymptotic probability of weighted-sum-queue overflow, *i.e.*,

$$\limsup_{k \rightarrow \infty} \frac{1}{k} \log \mathbb{P} \left( \sum_{n \in \mathcal{N}} w_n Q_n(0) > k \right),$$

where  $\mathbb{P}(\cdot)$  denotes the stationary distribution of the Markov chain  $\mathbf{Q}$  under a stable scheduling policy.

The *most-likely mode of queue overflow* under an RSM policy like the Log rule is in general quite different from the mode under the Exp rule. Recall that the latter minimizes the asymptotic probability of *max*-queue overflow [19], whereas the former minimizes the asymptotic probability of (weighted) *sum*-queue overflow.

This leads to basic questions as to which design objective is appropriate in designing opportunistic schedulers, and whether the asymptotic results are sufficiently

accurate to dictate which class of scheduler should be used. We consider this in the next section.

## 2.6 Evaluating opportunistic scheduler design objectives – simulations

In this section we discuss a simulation-based evaluation of opportunistic schedulers from various perspectives:

1. performance, including mean packet delays and 99<sup>th</sup> percentile delays of individual users as well the overall system;
2. sensitivity to both scheduler parameters and channel characteristics;
3. and graceful degradation, in terms of the fraction of users that meet QoS objectives under overloads.

Note we consider a system as overloaded if it can no longer meet users' QoS requirements, this might be due to a change in the channel characteristics, due to mobility etc. These perspectives are clearly interrelated yet for clarity we discuss them separately.

### 2.6.1 Simulation model and operational scenarios

We choose an HDR-like wireless downlink [32] to compare various scheduling rules, namely Log rule, MaxWeight, and Exp rule. Performance comparisons for an HDR downlink under Proportional Fair scheduling, MaxWeight, and Exp rule were presented in [39], and showed the Exp rule to be superior to the others. Note that the HDR downlink model differs from the system model presented in Section 2.2, however, we choose this as our simulation model to demonstrate the practical significance of our proposed scheduling rule and allow comparison with other simulations and theoretical

User $i$	1	2	3	4	5	6
$\mathbb{E}[R_n]$ kbps	572.8	392.1	304.6	250.1	215.1	187.9
User $i$	7	8	9	10	11	12
$\mathbb{E}[R_n]$ kbps	167.6	151.3	138.0	127.2	117.1	109.6

Table 2.1: Mean data rate supported by wireless channel of each user

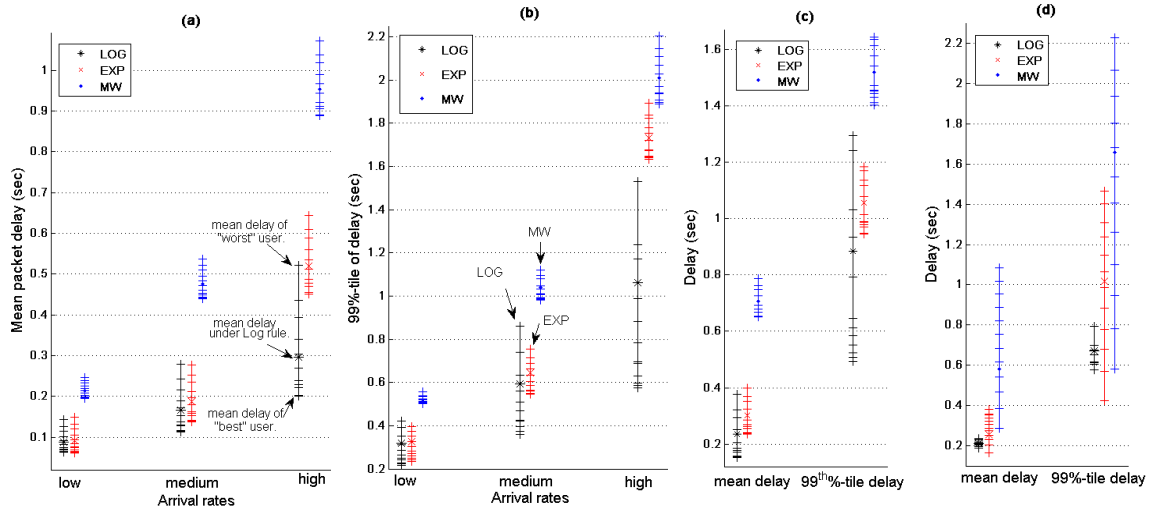


Figure 2.4: Simulation-based performance comparisons for three opportunistic scheduling policies, Log Rule, Exp rule and Max Weight: (a) mean delay and (b) 99<sup>th</sup> %-tile delay for each user and overall system, under *low*, *medium*, and *high* symmetric loads; (c) mean and 99<sup>th</sup> %-tile delay for each user and overall system under *low* symmetric traffic but when User 7 is moved to the cell edge; and (d) mean and 99<sup>th</sup> %-tile delay for users and overall system for the asymmetric traffic. Each cross-tick on vertical line marks a user's performance.

work in the literature. Thus, instead of i.i.d. channel, continuous time scheduling, and Poisson arrivals with exponentially distributed packets sizes, here we assume that channels are *correlated* over time, scheduling decisions are made once in each time slot of duration 1.67 ms, and each user's packets are 1Kb and arrive as i.i.d. Bernoulli processes.

We consider  $\bar{n} = 12$  heterogenous users connected to a single access point. The locations of the  $\bar{n}$  users are taken to be uniformly distributed in a circular cell. The wireless link between the access point and each user is taken as an independent

LOG	EXP	MW
$b_n = \frac{1}{\mathbb{E}[R_n]}$	$b_n = \frac{1}{\mathbb{E}[R_n]}$	$b_n = \frac{1}{\mathbb{E}[R_n]}$
$a_n = 10$	$a_n = 0.05,$	$\alpha = 1$
$c = 10$	$c = 1, \eta = 0.5$	

Table 2.2: Parameters used for each scheduling policy

Rayleigh fading channel with a Doppler frequency of 18 Hz. Specifically, in any time slot  $t \in \mathbb{Z}$ , the channel state (rate supported by the channel) of  $n^{\text{th}}$  user is given by,

$$R_n(t) \equiv \text{BW} \times \log_2(1 + \text{SINR}_n(t)) \quad \text{bits/sec}$$

and SINR (signal-to-interference-plus-noise ratio) is assumed to hold its value over the duration of the time slot. During each time slot, data is transmitted to a single user who is selected according to the scheduling policy. If user  $n$  is selected in time slot  $t$ , then (at most)  $1.67\text{ms} \times R_n(t)$  bits are transmitted from its queue. Table 2.1 gives the mean data rate  $\mathbb{E}[R_n]$  in bits/sec that the wireless channel of each user can support.

Let  $\mathcal{C}^{no} \in \mathbb{R}_+^{\bar{n}}$  denote the simplex obtained as the convex hull of origin and the following  $\bar{n}$  points,

$$\mathbb{E}[R_1]\mathbf{e}_1, \dots, \mathbb{E}[R_{\bar{n}}]\mathbf{e}_{\bar{n}}.$$

That is,  $\mathcal{C}^{no} \subset \mathcal{C}$  is the capacity region achievable by *non-opportunistic* (but possibly channel-aware) schedulers. Let  $\mathbf{r}^* \equiv (r_n^*, n \in \mathcal{N}) \in \mathcal{C}^{no}$  be the maximal point of  $\mathcal{C}^{no}$  that satisfies  $r_1^* = r_2^* = \dots = r_{\bar{n}}^*$ , then we have [32]

$$r_n^* = \left( \sum_{j=1}^{\bar{n}} \frac{1}{\mathbb{E}[R_j]} \right)^{-1} \text{ bits/sec}, \quad n \in \mathcal{N}.$$

We present simulation results for five operational scenarios. In the first three scenarios users see heterogenous channels but have homogenous traffic with *low*  $\boldsymbol{\lambda}^{(s,l)}$ ,

medium  $\lambda^{(s,m)}$ , or high  $\lambda^{(s,h)}$  packet arrival rates given by,

$$\begin{aligned}\lambda^{(s,m)} &= 2.3r^* \times \frac{1}{1024 \text{ bits/packet}} \text{ packets/sec,} \\ \lambda^{(s,l)} &= 0.98\lambda^{(s,m)}, \\ \lambda^{(s,h)} &= 1.02\lambda^{(s,m)}.\end{aligned}$$

In words, for the medium case a user's arrival rate is 2.3 times higher than that is stabilizable by a non-opportunistic scheduler; the low and the high arrival rates are respectively 2% lower and higher than the medium. Fig. 2.4-(a) and (b) show performance results under these three homogenous load scenarios – see caption for detailed explanation. In the fourth scenario the arrival rate is kept *low* but User 7 (see Table 2.1) is moved to the edge of cell, which increases the system load. Fig. 2.4-(c) exhibits the results for this case. For the fifth scenario, users have heterogenous arrival rates given by,

$$\lambda_n = 2.35 \times \frac{\mathbb{E}[R_n]}{n} \times \frac{1}{1024} \text{ packets/sec}$$

*i.e.*, arrival rate vector  $\lambda$  is proportional to the mean channel rate vector  $\mathbb{E}[\mathbf{R}]$  and 2.35 times higher than that is stabilizable by a non-opportunistic scheduler. Fig. 2.4-(d) exhibits the performance results for this case.

## 2.6.2 Discussion of results and insights

**Performance comparison** As seen in Fig. 2.4-(a) and (b) under *low* traffic, Log and Exp rules are comparable and outperform MaxWeight. Although users see heterogeneous channels the performance they see is very similar verifying we have a good choice for the scheduling policy parameters, see Table 2.2. However, as the traffic rate increases there are clear trends: the users' and overall means are better under the Log rule (up to 20% reduction), while the variability or spread of the 99<sup>th</sup> percentile delay across users is lower under the Exp rule (the 99<sup>th</sup> percentile

delay spread is halved). Note, however, that all but two users have 5-70% better 99<sup>th</sup> percentile delay under the Log rule versus the Exp rule. The situation is even more favorable to the Log rule at higher loads, where all users experience 20-80% lower mean and 99<sup>th</sup> percentile delays versus the Exp rule (which still maintains a lower delay spread than the Log rule). Clearly for heterogenous channels, the Exp Rule's strong bias towards balancing queues is excessively compromising the realized throughput, and eventually the mean delays and tails for almost all users. Although asymptotically Exp rule should be optimal, the pre-exponent must also be playing a role in determining the systems performance.

**Sensitivity** Another way to view this is that the actual performance (not the theoretical asymptotic tail) achieved by the Exp rule is more sensitive to the absolute values of  $\mathbf{a}$ . Fig. 2.4-(a) and (b) exhibit the degeneration in the relative performance of Exp vs Log rule for a set of fixed parameters as the load is scaled up. The RSM property of the Log rule naturally calibrates the scheduler to increased load. Similarly, comparing the *low* and the *medium* results in Fig. 2.4-(a) and (b) to those in Fig. 2.4-(c) and Fig. 2.4-(d), we see the performance sensitivity to changes in the channel or load characteristics. In both cases for most users the mean and 99<sup>th</sup> percentile delays are better under the Log rule and in the case of heterogenous loads, *i.e.* Fig. 2.4-(d), the delay spreads are also improved. So unless parameters can be carefully tuned to possibly changing loads and unpredictable channel capacities, the Log rule appears to be a more robust scheduling policy. Intuitively, this is what one would expect from optimizing for the overall average versus worst case asymptotic tail.

**Graceful degradation** Suppose the user flows correspond to buffered streaming audio sessions with a QoS requirement of 99<sup>th</sup> percentile delay below 1 sec, see *e.g.*, [39]. Under medium traffic (Fig. 2.4-b), all users comfortably meet the QoS requirement for both the Log and the Exp rule. However, if User 7 moves to the cell

edge (Fig. 2.4-c), then under the Log rule, 9 out of 12 users versus 6 out of 12 for Exp rule meet the QoS requirement. If instead, the traffic loads associated with the users were to change, then as shown in Fig. 2.4-(d) all users meet the QoS requirement under the Log rule versus only 6 out of 12 under the Exp rule. Unless system resource is provisioned extremely conservatively, *i.e.* for worst case, we can expect such scenarios to arise, and this work suggests Log rule would provide a more graceful degradation of service.

## 2.7 Scheduling in Multichannel Systems

This section focuses on implementation of the Log rule for scheduling and resource allocation in OFDMA-based multichannel systems, *e.g.*, WiMax, LTE. We begin by appropriately modifying the single-channel HDR-like system model used in the previous section to now capture an OFDMA-based multichannel system where power and bandwidth can be shared across multiple users over a scheduling/transmission time interval.

So far, the implied meaning of a channel state  $m \in \mathcal{M}$  has been “a collection of quantized SNRs measured and reported by each user.” Since we were considering a TDMA system where the scheduled user was allocated all the resources (power and bandwidth), we implicitly converted the SNR reported by the  $n^{\text{th}}$  user into the supported transmission rate  $r_n(m)$ . Therefore, the region  $\mathcal{C}_m$  – available service rate region conditional on channel being in state  $m$  – was given by a simplex defined by rates  $(r_n(m), n \in \mathcal{N})$ . All scheduling policies considered so far (see (2.15), (2.17), (2.18), and (2.25)) picked a vertex of the simplex  $\mathcal{C}_m$ , or equivalently, scheduled a single user.

However, in wideband/multichannel systems it is undesirable (and oftentimes even infeasible) to allocate all the resources to one user over a scheduling time in-



terval [40]. The main reasons are as follows and will be addressed by our proposed implementation:

- allocating all resources to one user in the presence of typically hundreds of active users will result in bursty service with long delays between successive allocations to a user, and
- the service rate region available under all possible power and bandwidth allocations to *multiple* users over a scheduling time interval, is larger than the simplex  $\mathcal{C}_m$  defined above.

### 2.7.1 Modifications to the system model

As in the previous section, we consider a time slotted system. We can capture a multichannel system by appropriately redefining the meaning of channel state  $m$ , and associating with it a suitable service region  $\mathcal{C}_m$ :

- The channel state is now defined as the collection of quantized SNRs reported by each user and measured at a reference power level for each *resource block*<sup>4</sup> (RB) group (collection of a few consecutive resource blocks). We continue to denote by  $\mathcal{M}$  the set of all possible channel states.
- For each  $m \in \mathcal{M}$ , the service region  $\mathcal{C}_m$  is the convex hull of the service rates (in bits/time-slot) that can be jointly offered to the  $\bar{n}$  users under all *feasible* resource (power and bandwidth) allocations, conditional on the channel being in state  $m$ .

Feasibility is determined by the system specifications and computational complexity afforded, *e.g.*, limits on the minimum and maximum bandwidth that can be allocated to a user, and limits on power per user and per RB; see, *e.g.*, [41] for a formal

description of the rate region  $\mathcal{C}_m$  in terms of various feasibility constraints. Also we let  $\mathbf{q}$  denote the queue length vector in number of *bits* rather than packets.

### 2.7.2 Scheduling and resource allocation polices

The general channel-aware (but queue-oblivious) rate-adaptive scheduling and resource allocation problem for an OFDMA system is typically defined as follows (see [40,41] and references therein): when the channel is in state  $m$ , allocate resources to users so that the long-run average offered service rate conditional on the channel being in state  $m$  solves the following program,

$$\begin{aligned} & \text{maximize} && \mathcal{U}(\mathbf{u}) , \\ & \text{subject to} && \mathbf{u} \in \mathcal{C}_m , \end{aligned} \tag{2.27}$$

where  $\mathcal{U} : \mathbb{R}_+^{\bar{n}} \rightarrow \mathbb{R}$  is a given *utility* function satisfying concavity, smoothness, and separability properties. An optimal rate  $\mathbf{u}^*$  (*i.e.*, maximizer of (2.27)) corresponds to an allocation of RBs and transmit power across users. However, much like the single-channel case, such a queue-oblivious scheduler will not be throughput-optimal.

Using a convex program formulation like that in (2.27), we shall define the Log rule for multichannel systems as follows.

**Definition 2.4** *When the system is in state  $(\mathbf{q}, m)$ , allocate resources so that the long-run average offered service rate conditional on the system being in state  $(\mathbf{q}, m)$  is given by the solution to the following program,*

$$\begin{aligned} & \text{minimize} && h^L(\mathbf{q} - \mathbf{u}) , \\ & \text{subject to} && \mathbf{u} \in \mathcal{C}_m , \end{aligned} \tag{2.28}$$

where  $h^L : \mathbb{R}_+^{\bar{n}} \rightarrow \mathbb{R}$  is given by,

$$h^L(\mathbf{y}) \equiv \sum_{n \in \mathcal{N}} b_n \left( \left( \frac{c}{a_n} + y_n \right) \log(c + a_n y_n) - y_n \right) , \tag{2.29}$$

---

<sup>4</sup>Resource blocks are the smallest chunks of bandwidth that can be allocated to a user over a scheduling time interval.

and the constants  $(a_n, n \in \mathcal{N}), (b_n, n \in \mathcal{N})$ , and  $c$  are as in the definition of the single-channel Log rule (see Definition 2.3).

The function  $h^L$  is convex increasing and can be viewed as an approximation for the relative *cost function* on the queue state space, satisfying the Bellman's equation. Also, note that

$$\begin{aligned} \nabla h^L(\mathbf{q} - \mathbf{0}) &\equiv \left( \frac{\partial h^L(\mathbf{q} - \mathbf{u})}{\partial u_n}, n \in \mathcal{N} \right) \Big|_{\mathbf{u}=\mathbf{0}}, \\ &= -\mathbf{g}^L(\mathbf{q}). \end{aligned}$$

Therefore, for a single-channel system where  $\mathcal{C}_m$  was a *small* simplex, the single-channel Log rule (see (2.25)) can be viewed as linearizing the convex program (2.28) using the first-order Taylor expansion of  $h^L(\mathbf{q} - \mathbf{u})$  at  $\mathbf{u} = \mathbf{0}$ , *i.e.*,

$$\begin{aligned} &\text{maximize} \quad \langle \mathbf{g}^L(\mathbf{q}), \mathbf{u} \rangle, \\ &\text{subject to} \quad \mathbf{u} \in \mathcal{C}_m, \end{aligned}$$

**Remark 2.3** *The linearized version stated above may not be suitable in a multichannel system because, if the region  $\mathcal{C}_m$  is a simplex (or close to a simplex), the linearized program reduces to picking a vertex and thus a single user (or allocating most of the resources to a single user) even if all weighted queues  $a_n q_n$  are equal. As mentioned earlier, this is undesirable in a multichannel system. The region  $\mathcal{C}_m$  in a multichannel system can still be a simplex if, for instance, the users only report one effective SNR over the entire bandwidth (e.g. wideband CQI in LTE [2]) and the power allocated per resource block is fixed (i.e., the mapping from the reported CQI to the chosen modulation and coding scheme is fixed). Moreover, the analysis in [42] shows that for certain symmetric on-off multichannel systems, any resource allocation policy given by a linear program will have a zero large-deviation rate function associated with the max-queue (asymptotically in the number of users and channels, and in the small buffer regime.)*

Fast computation algorithms to solve program (2.27) (or (2.28)) for a general concave increasing separable utility function  $\mathcal{U}(\cdot)$  are given in [40] and [41] (and references therein). The complexity of the algorithm depends on the feasibility constraints that define the region  $\mathcal{C}_m$ . For example, assuming that service rate at any SNR is equal to the Shannon's capacity (with a possible gap), a total power constraint, and that each resource block group can be shared by an arbitrary number of users, the algorithm obtained in [40] has a complexity of  $O(\bar{n}\bar{b})$  per *iteration* where  $\bar{b}$  is the number of resource block groups over which the users report their measured SNR/CQI. It is reported that typically about 25 iterations are needed for the algorithm to converge. For a similar system where users only report a single effective SNR measured across the entire bandwidth (however the power and/or bandwidth can still be shared by multiple users), the complexity of the above algorithm reduces to  $O(\bar{n})$  per iteration, which is the complexity of single-channel scheduling algorithms (see (2.17), (2.18), (2.25)) which pick only a single user.

See [43] for an implementation of a complete queue-and-channel-aware scheduler for an LTE downlink, and a performance comparison through simulation of the Log and Exp rules. The simulation results presented in [43] for a multichannel system agree with ones presented in Section 2.6 for a single-channel system, and reinforce the observations made in Section 2.6.2.

## 2.8 Conclusion

We have made the case not only for a new class of opportunistic scheduling policies, but also for new metrics to design and evaluate such schedulers. Our conclusion is simple, and in retrospect intuitive, a scheduler 'optimized' for the overall system performance is likely to be more robust to changes in the traffic and channel statistics than the one optimized for the worst case. The numerical results show that

mean delay optimal schedulers are weakly RSM, and in some cases, even RSM. The proposed Log rule policy is RSM and although not necessarily mean delay-optimal for a given scenario, exhibits the promised robustness vs the Exp and MaxWeight rules. The set of presented simulations (and others given in [43]) lend support to the practical benefits of this new class of policies.

Further asymptotic results, briefly discussed in this chapter and presented in detail in the next, show that an RSM policy minimizes the tail of sum-queue distribution.

## Chapter 3

# Large deviations sum-queue optimality of a radial sum-rate monotone opportunistic scheduler

### 3.1 Overview, main contributions, and organization

A centralized wireless system is considered that is serving a fixed set of users with time varying channel capacities. An opportunistic scheduling rule in this context selects a user (or users) to serve based on the current channel state and user queues. Unless the user traffic is symmetric and/or the underlying capacity region a polymatroid, little is known concerning how performance optimal schedulers should tradeoff *maximizing current service rate* (being opportunistic) versus *balancing unequal queues* (enhancing user-diversity to enable future high service rate opportunities). By contrast with currently proposed opportunistic schedulers, *e.g.*, MaxWeight and Exp Rule, a radial sum-rate monotonic (RSM) scheduler de-emphasizes queue-balancing in favor of greedily maximizing the system service rate as the queue-lengths are scaled up linearly. In this chapter, it is shown that an RSM opportunistic scheduler, p-Log Rule, is not only throughput-optimal, but also maximizes the asymptotic exponential decay rate of the sum-queue distribution for a two-queue system. The result complements existing optimality results for opportunistic scheduling and point to RSM schedulers as a good design choice given the need for robustness in wireless systems with both heterogeneity and high degree of uncertainty.

## Relation to other chapter

As a corollary of the Lemmas 3.1–3.3 of this chapter, we show in Appendix A that under sufficient load, RSM policies are *fluid-scale asymptotic optimal* (FSAO).

## Contributions

In the introductory chapter, we pointed out that Exp rule minimizes the asymptotic probability of *max*-queue overflow (APMO). However, unlike LCQ and LQHPR policies in their respective setting, in the presence of a tradeoff between *service rate maximization* and *queue balancing*, the asymptotic optimality of Exp rule does not translate to minimizing the asymptotic probability of *sum*-queue overflow (APSO) or the mean delay. In order to minimize APMO, the desired<sup>1</sup> mode of overflow is one where all queues (more precisely, the set of overflowing queues which then exclusively share the server) grow at the same rate and overflow at the same time [19]. This constrains the system throughput, while, of course, keeping the queue lengths *equal* across users.

By contrast, we will show that in order to minimize APSO, the desired mode of overflow is one where the total service rate (system throughput in packet/sec) is the highest possible while queues may build up at different rates. Using this, we give a tight lower bound on APSO under *any* scheduler.

Next we show that a radial sum-rate monotone scheduler, called the pseudo-Log rule (p-Log), minimizes APSO by achieving the lower bound mentioned above. Although our focus is on overflows of the sum-queue instead of overflows of the max-queue as in [19], the general technique of proof in [19] lends itself well to our problem and we rely heavily on the results developed therein. Other desirable features of radial

---

<sup>1</sup>By the “desired” mode of max-queue overflow we mean the mode which gives the lower bound on the asymptotic probability of max-queue overflow under *any* scheduler, as given in [19]; the likely mode of max-queue overflow under the Exp rule matches the desired mode.

sum-rate monotone schedulers have been explained in Chapter 2, which include,

- i. reducing mean delay,
- ii. graceful degradation of service in terms of fraction of users that can meet their Quality-of-Service requirements during transient overloads,
- iii. robustness to uncertainty in traffic and channel statistics.

## Organization

Rest of the chapter is organized as follows. The system model is described in Section 3.2. Queue-and-channel aware schedulers of interest, namely, MaxWeight, Exp rule, and Log rule, and the property of radial sum-rate monotonicity are reviewed in Section 3.3, followed by the introduction of pseudo-Log scheduling rule in Section 3.4. The ‘main result’ is summarized in Section 3.5. Some preliminary discussion and relevant large deviation principles follow in Section 3.6. The proofs for the lower and the upper bounds stated in the main result are given in Section 3.7 and 3.8 respectively. After defining local fluid sample paths and developing essential results (summarized in Table 3.1) in Section 3.9, the optimality of the p-Log rule, *i.e.*, the last part of main result is proved in Section 3.10. Immediate extensions of the main result to some other interesting system models are presented in the concluding Section 3.11.

## 3.2 System model

We consider the problem of dynamically allocating a time-varying server to two queues. Each Queue  $i \in I = \{1, 2\}$  is fed by an independent arrival process  $(\mathbf{A}_i(t), t = 0, 1, \dots)$  that is i.i.d. over  $t$  and where  $\mathbf{A}_i(t) \in \mathbb{Z}_+$  denotes the number of packets arriving in time slot  $[t, t + 1)$ . We assume that the arrivals are bounded, *i.e.*,



$\mathbf{A}_i(\cdot) \leq C$  for some finite  $C > 0$ . Let  $\mathbf{A}(t) = (\mathbf{A}_i(t), i \in I)$ , and vector  $\bar{\lambda} = \mathbb{E}[\mathbf{A}(0)]$  denote the mean arrivals to the queues. We use bold face, *e.g.*,  $(\mathbf{A}(t), t = 0, 1, \dots)$ , to denote the random process and plain font, *e.g.*,  $(A(t), t = 0, 1, \dots)$ , to denote a realization of the process.

A server with randomly varying service rates is available to the two queues and modeled as follows. The server has a time-varying state that is modeled by an i.i.d. random process  $(\mathbf{m}(t), t = 0, 1, \dots)$ , where  $\mathbf{m}(t) \in \mathcal{M} = \{1, 2, \dots, M\}$  for some finite  $M > 0$  denotes the state of the server over  $[t, t + 1)$ , and is drawn from distribution  $\pi = (\pi_1, \dots, \pi_M) > 0$ . Associated with each server state  $m \in \mathcal{M}$  is a vector  $\mu^m \in \mathbb{Z}_+^2$ . When in state  $m$  over a time slot, the server can either serve at most  $\mu_1^m$  packets from Queue 1, or at most  $\mu_2^m$  packets from Queue 2. The scheduling problem is thus to allocate the server to one of the queues for each time slot such that a given optimality criterion is met. This will be formally described later.

At any integer time  $t$ ,  $\mathbf{Q}(t) = (\mathbf{Q}_i(t), i \in I) \in \mathbb{Z}_+^2$  is a random vector, where  $\mathbf{Q}_i(t)$  denotes the number of packets in the  $i^{\text{th}}$  queue at the end of time slot  $[t - 1, t)$ . Let  $S_t$  denote a system sample path up to time  $t$ , *i.e.*,

$$S_t \equiv \left( (m(\tau), Q(\tau), A(\tau - 1)), \tau \leq t \right),$$

and  $\mathcal{S}_t$  denote the space of all feasible realizations  $S_t$ . Let  $i_t^* : \mathcal{S}_t \rightarrow I$  denote the queue scheduled to receive service during time slot  $[t, t + 1)$ , *i.e.*, we assume that the system sample path  $S_t$  is available for making the scheduling decision for time slot  $[t, t + 1)$ . The evolution of the queue process under the scheduling decision  $i_t^*(S_t)$  is given by,

$$\mathbf{Q}_i(t + 1) = \left( \mathbf{Q}_i(t) - \mu_i^{m(t)} \mathbf{1}_{\{i_t^*(S_t) = i\}} \right)^+ + \mathbf{A}_i(t) .$$

The sequence of functions  $(i_t^*(\cdot), t = 0, 1, \dots)$  is called a *scheduler* or *scheduling policy*. It is easy to see that under a static state-feedback scheduler, *i.e.*, one where

$i_t^*(S_t) = i^*(Q(t), m(t))$ , the process  $(\mathbf{Q}(t), t = 0, 1, \dots)$  forms a discrete time Markov chain on  $\mathbb{Z}_+^2$ .

In the sequel, we will extend the domain of all discrete time processes and functions to continuous time: a function originally defined on integer times has the same value at any real  $t$  that it takes at  $\lfloor t \rfloor$ . All such processes and functions lie in the space of *real-valued right continuous functions with left limits*, denoted by  $\mathcal{D}$ . We assume that  $\mathcal{D}$  is endowed with the topology of uniform convergence over compact sets (u.o.c), and the  $k$ -times product space  $\mathcal{D}^k$  with the product topology. Lastly, let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a probability space that is large enough to define all the random processes in this chapter.

For a given weight vector  $b = (b_1, b_2) > 0$ , we are interested in finding a scheduler which, informally speaking, minimizes the tail of the distribution of weighted sum-queue  $\sum_{i \in I} b_i \mathbf{Q}_i(\cdot)$ . We will show that under a static state-feedback scheduler, namely the p-Log rule described in Section 3.4, the asymptotic probability of weighted sum-queue overflow in the steady state, *i.e.*,

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( \sum_{i \in I} b_i \mathbf{Q}_i(0) \geq n \right),$$

is *minimized* (see Theorem 3.1 for a formal statement.) The p-Log rule depends only on weight vector  $b$  and does not require any knowledge of arrival or server-state distributions.

### Capacity region

For each server state  $m \in \mathcal{M}$ , let  $V^m \in \mathbb{R}_+^2$  denote the convex hull of vertices  $(0, 0)$ ,  $(0, \mu_2^m)$ , and  $(\mu_1^m, 0)$ . Then, conditional on the server being in state  $m$ , the average service jointly offered to the two queues under any scheduling rule (such that the average exists) lies in the *triangle*  $V^m$ . Define the capacity region  $V_\pi$  as the set of average service vectors offered to the two queues under all possible scheduling rules,

then  $V_\pi$  is a convex polyhedron given by the weighted Minkowski sum of regions  $V^m$ , *i.e.*,

$$\begin{aligned} V_\pi &= \pi_1 V^1 \oplus \cdots \oplus \pi_M V^M, \\ &= \left\{ \sum_{m \in \mathcal{M}} \pi_m v(m) : v(m) \in V^m, m \in \mathcal{M} \right\}. \end{aligned} \quad (3.1)$$

See Fig. 3.1-*a* for a graphical illustration of the capacity region for a server with  $M = 6$  states.

Let  $\{r_1, \dots, r_{M'}\}$  for some  $M' \leq M$  be the set of strictly positive and finite slopes of the outer normal vectors to the facets of capacity region  $V_\pi$ . The slopes are indexed such that  $0 < r_1 < r_2 < \dots < r_{M'} < \infty$ . Also, let  $r_0 = 0$  and  $r_{M'+1} = \infty$ . For example, see Fig. 3.1-*a* for a depiction of a capacity region with  $M' = 5$  facets with outer normal slopes in  $(0, \infty)$ . Finally, let  $\hat{V}_\pi = \{\hat{v}^1, \dots, \hat{v}^{M'+1}\}$  denote the set of maximal vertices of the capacity region  $V_\pi$ . The vertices in set  $\hat{V}_\pi$  are indexed such that  $\hat{v}_1^1 > \hat{v}_1^2 > \dots > \hat{v}_1^{M'+1}$ , *i.e.*, the vertex  $\hat{v}^m$  lies at the intersection of the facets with outer normal slopes  $r_{m-1}$  and  $r_m$ ; *e.g.*, see vertex  $\hat{v}^2$  in Fig. 3.1-*a*.

We assume that there exists a  $v \in V_\pi$  such that  $\bar{\lambda} < v$ , which is a sufficient condition for stabilizability of the queues [10]. If the above condition is met, then there exists at least one static state feedback scheduler under which the Markov chain  $(Q(t), t = 0, 1, \dots)$  is ergodic.

**Remark 3.1** *Since we assumed that the server can be allocated to at most one queue per time slot, the region  $V^m$  is obtained by taking the convex hull of service vectors in the set  $\{(0, 0), (\mu_1^m, 0), (0, \mu_2^m)\}$ . However, we can relax this assumption and allow the server to be shared between the two queues during a time slot. That is, we can associate with each server state  $m$  a set of  $k_m$  service vectors*

$$\{(\mu_1^m(1), \mu_2^m(1)), \dots, (\mu_1^m(k_m), \mu_2^m(k_m))\},$$

and allow the server to operate at any one of these service vectors. In this more general case, each region  $V^m$  will be an arbitrary convex polyhedron obtained by taking the convex hull of all feasible service vectors associated with server state  $m$ . For example,  $V^m$  can be information theoretic polymatroids as in [13]. The optimality results presented in this chapter will still hold with such a relaxation; see Section 3.11 for some details.

### 3.3 Throughput-optimal schedulers and radial sum-rate monotonicity

The throughput-optimal schedulers MaxWeight, Exp rule, and Log rule, introduced in the previous chapter, are all static state-feedback. Let the vector fields  $h^{mw}(\cdot)$ ,  $h^{exp}(\cdot)$  and  $h^{log}(\cdot)$  on  $\mathbb{R}_+^2$  be given as follows: for all  $x \in \mathbb{R}_+^2$ ,

$$\begin{aligned} h^{mw}(x) &= (b_i x_i^\alpha, i \in I), \\ h^{exp}(x) &= \left( b_i \exp\left(\frac{a_i x_i}{c + (0.5(a_1 x_1 + a_2 x_2))^\eta}\right), i \in I \right), \\ h^{log}(x) &= (b_i \log(1 + a_i x_i), i \in I), \end{aligned}$$

for any fixed positive  $b_i$ 's,  $a_i$ 's,  $\alpha$ ,  $c$ , and  $0 < \eta < 1$ . Then, when the system is in state  $(\mathbf{Q}(t), \mathbf{m}(t)) = (Q, m) \in \mathbb{Z}_+^2 \times \mathcal{M}$ , the MaxWeight scheduler serves a queue  $i_{mw}^*$  given by,

$$i_{mw}^*(Q, m) \in \arg \max_{i \in I} h_i^{mw}(Q) \mu_i^m, \quad (3.2)$$

augmented with any fixed tie-breaking rule. The Exp rule  $i_{exp}^*$  and the Log rule  $i_{log}^*$  are defined similarly by substituting  $h^{exp}$  and  $h^{log}$  respectively in place of  $h^{mw}$ . Indeed numerous vector field based throughput-optimal schedulers can be engineered so as to respond differently to the disparity among the users' queue lengths, *i.e.*, make different tradeoffs between *service rate maximization* and *queue balancing*. For reference see, *e.g.*, [37], which gives necessary and sufficient conditions for a vector field based scheduler to be throughput-optimal.

We refer to a scheduler as *radial sum-rate monotone* if, as the queues scale up linearly, the scheduling rule allocates the server in a manner that de-emphasizes queue-balancing in favor of greedily maximizing the current service rate. More formally, let  $v(Q) \in V_\pi$  be the vector of average service offered to the queues under a static state-feedback scheduler  $i^*$ , conditional on queue state being  $Q$ , *i.e.*,

$$v(Q) = \left( \mathbb{E}[\mu_i^{\mathbf{m}} \mathbf{1}_{\{i^*(Q, \mathbf{m})=i\}}], i \in I \right) , \quad (3.3)$$

where expectation is with respect to (random)  $\mathbf{m}$  drawn from distribution  $\pi$ .

**Definition 3.1** *A scheduling policy  $i^*$  is radial sum-rate monotone with respect to weight vector  $b > 0$  if for any  $Q$  and scalar  $\theta$  such that  $\theta Q \in \mathbb{Z}_+^n$ , the weighted sum of expected offered service,  $\langle b, v(\theta Q) \rangle$ , is an increasing function of  $\theta$ , and*

$$\lim_{\theta \rightarrow \infty} \langle b, v(\theta Q) \rangle = \max_y \left( \langle b, y \rangle \mid y \in V_\pi \text{ and } y_i = 0 \text{ if } Q_i = 0 \right).$$

Let  $v^{mw}(Q)$  denote the expected service vector under the MaxWeight scheduler, *i.e.*, the vector obtained by substituting  $i_{mw}^*$  for  $i^*$  in (3.3); similarly, let  $v^{exp}(Q)$  and  $v^{log}(Q)$  denote the expected service vectors under the Exp and Log rule respectively. Also, fix a weight vector  $b > 0$ . Next, under schedulers Maxweight, Exp rule, and Log rule respectively, we will identify the sets  $\mathcal{S}_0^{mw}$ ,  $\mathcal{S}_0^{exp}$ , and  $\mathcal{S}_0^{log}$  given *approximately*<sup>2</sup> by,

$$\mathcal{S}_0^{(\cdot)} \approx \left\{ Q \in \mathbb{Z}_+^2 : v^{(\cdot)}(Q) \in \arg \max_{y \in V_\pi} \langle y, b \rangle \right\} ,$$

in words, the set of queue states such that the expected service vector  $v^{(\cdot)}(Q)$  has the maximum weighted sum with respect to weight vector  $b$ . For example, for the capacity region shown in Fig. 3.1-*a* and weight vector  $b = (1, 1)$ , Fig. 3.1-*b-d* illustrates the

---

<sup>2</sup>Under the formal definition of set  $\mathcal{S}_0^{(\cdot)}$  given later, the queue states on the boundary of set  $\{Q \in \mathbb{Z}_+^2 : v^{(\cdot)}(Q) \in \arg \max_{y \in V_\pi} \langle y, b \rangle\}$  may or may not lie in  $\mathcal{S}_0^{(\cdot)}$ .

sets  $\mathcal{S}_0^{mw}$ ,  $\mathcal{S}_0^{exp}$ , and  $\mathcal{S}_0^{log}$ . The set  $\mathcal{S}_0^{mw}$  is a cone, the set  $\mathcal{S}_0^{exp}$  a cylinder with gradually increasing diameter, and the set  $\mathcal{S}_0^{log}$  resembles a French horn. The set  $\mathcal{S}_0^{log}$  is such that for any  $Q > 0$ , we have  $\theta Q \in \mathcal{S}_0^{log}$  for all  $\theta$  large enough, indicating that the Log rule is radial sum-rate monotone. The MaxWeight and Exp rule are not radial sum-rate monotone. In fact, Exp rule is the ‘opposite’ of radial sum-rate monotone in that, for any  $Q$  such that  $a_1 Q_1 \neq a_2 Q_2$ , we have  $\theta Q \notin \mathcal{S}_0^{exp}$  for all  $\theta$  large enough. A formal description of the sets  $\mathcal{S}_0^{(\cdot)}$  is as follows.

The vector  $h^{(\cdot)}(Q)$  can be shown to be an outer normal vector to the capacity region  $V_\pi$  at point  $v^{(\cdot)}(Q)$  for each scheduler  $(\cdot) \in \{mw, exp, log\}$ , *i.e.*,

$$v^{(\cdot)}(Q) \in \arg \max_{y \in V_\pi} \langle y, h^{(\cdot)}(Q) \rangle, \quad (3.4)$$

See, *e.g.*, Lemma 2.1 of [22]. Recall the set of outer normal slopes,

$$\{r_0, r_1, \dots, r_{M'}, r_{M'+1}\},$$

and let  $r_k$  be the largest slope strictly less than  $b_2/b_1$  (*i.e.* the slope of vector  $b$ ) and  $r_l$  the smallest slope strictly greater than  $b_2/b_1$ ; thus if  $r_{k+1} \neq b_2/b_1$  then  $l = k + 1$ , otherwise  $l = k + 2$ . For example, in Fig. 3.1-a, for weight vector  $b = (1, 1)$ , we have  $k = 2$  and  $l = 4$ . For each scheduler  $(\cdot) \in \{mw, exp, log\}$  we define the set,

$$\mathcal{S}_0^{(\cdot)} = \left\{ x \in \mathbb{R}_+^2 : r_k < \frac{h_2^{(\cdot)}(x)}{h_1^{(\cdot)}(x)} < r_l \right\}.$$

Then, by (3.4), for any  $Q \in \mathcal{S}_0^{(\cdot)} \cap \mathbb{Z}_+^2$ , we must have  $v^{(\cdot)}(Q) \in \arg \max_{y \in V_\pi} \langle y, b \rangle$ . Each region  $\mathcal{S}_0^{(\cdot)}$  is bounded by *switching curves* given by,

$$\left\{ x \in \mathbb{R}_+^2 : \frac{h_2^{(\cdot)}(x)}{h_1^{(\cdot)}(x)} = r_k \right\} \quad \text{and} \quad \left\{ x \in \mathbb{R}_+^2 : \frac{h_2^{(\cdot)}(x)}{h_1^{(\cdot)}(x)} = r_l \right\}.$$

For any queue state  $Q$  lying on these switching curves, the argmax in (3.4) is not unique and whether  $v^{(\cdot)}(Q)$  lies in the set  $\arg \max_{y \in V_\pi} \langle y, b \rangle$  depends on the tie breaking rule associated with (3.2). For each  $m \in \{1, \dots, k, l+1, \dots, M'+1\}$ , we can also

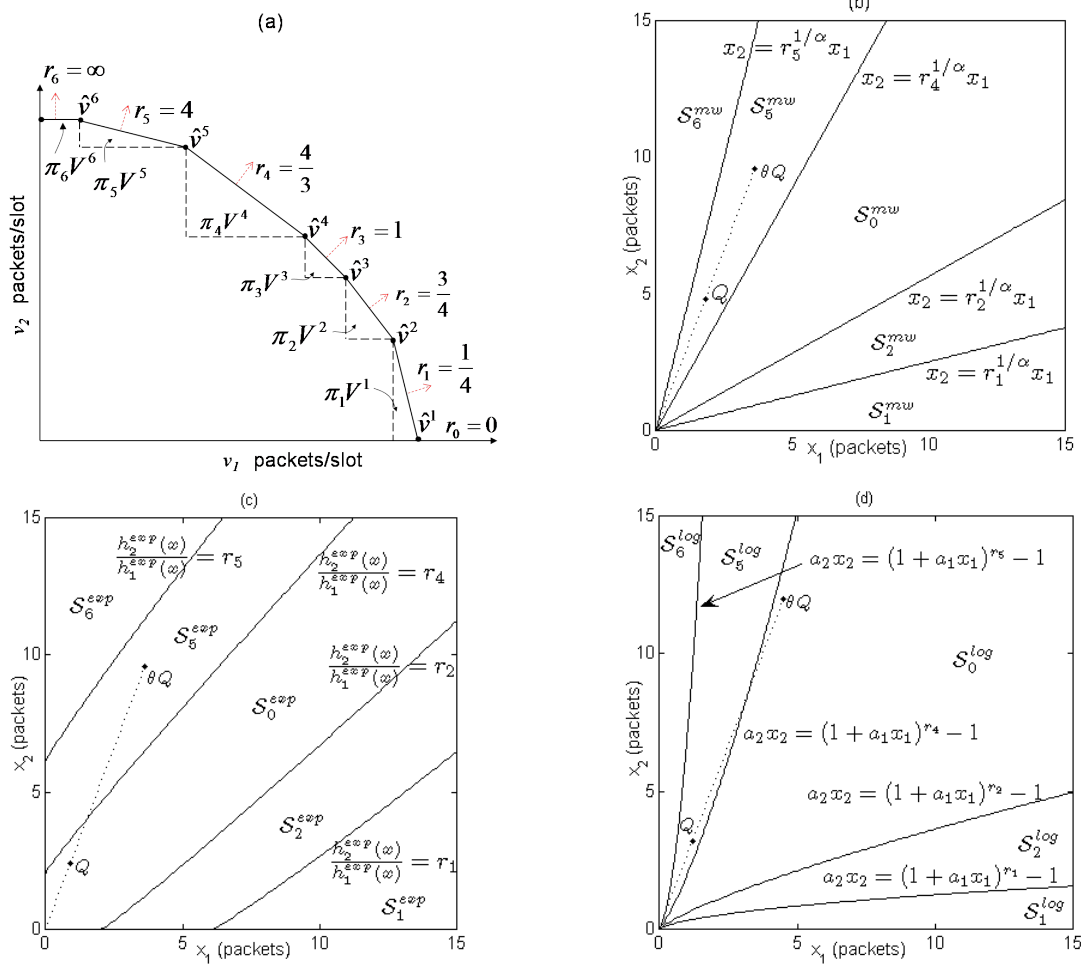


Figure 3.1: (a) Capacity region for  $\mu^m \in \{(1, 4), (3, 4), (1, 1), (4, 3), (4, 1), (1, 0)\}$ , depicting Minkowski addition, outer-normal vectors and maximal vertices; the resulting switching curves under (b) MaxWeight, (c) Exp rule, (d) Log rule, for weight vector  $b = (1, 1)$ .

define a set,

$$\mathcal{S}_m^{(\cdot)} = \left\{ x \in \mathbb{R}_+^2 : r_{m-1} < \frac{h_2^{(\cdot)}(x)}{h_1^{(\cdot)}(x)} < r_m \right\} .$$

Then, again by (3.4), for any  $Q \in \mathcal{S}_m^{(\cdot)} \cap \mathbb{Z}_+^2$ , we must have  $v^{(\cdot)}(Q) = \hat{v}^m$ . That is, each set  $\mathcal{S}_m^{(\cdot)}$  for  $m \in \{1, \dots, k, l+1, \dots, M'+1\}$  is associated with a unique vertex. For  $m \in \{1, \dots, k-1, l+1, \dots, M'\}$ , the switching curve between any two regions  $\mathcal{S}_m^{(\cdot)}$  and  $\mathcal{S}_{m+1}^{(\cdot)}$  is given by,

$$\left\{ x \in \mathbb{R}_+^2 : \frac{h_2^{(\cdot)}(x)}{h_1^{(\cdot)}(x)} = r_m \right\} ,$$

and as before, for any queue state  $Q$  lying on this switching curve, the argmax in (3.4) is not unique and whether it lies in set  $\mathcal{S}_m^{(\cdot)}$  or  $\mathcal{S}_{m+1}^{(\cdot)}$  depends on the tie breaking rule associated with (3.2). Fig. 3.1-*b-d* also show the switching curves and sets  $\mathcal{S}_m^{(\cdot)}$  under MaxWeight, Exp rule, and Log rule. In fact, for MaxWeight and Log rule, the switching curves can be given in closed form, *e.g.*, for Log rule, the switching curves are given by,

$$a_2 x_2 = (1 + a_1 x_1)^{\frac{b_1}{b_2} r} - 1 ,$$

for  $r \in \{r_1, \dots, r_{M'}\}$ .

### 3.4 The pseudo-Log scheduling rule

In this section we introduce a static state-feedback and radial sum-rate monotone scheduler, denoted the pseudo-Log (p-Log) rule. We subsequently show in Theorem 3.1 that the p-Log rule minimizes the asymptotic probability of weighted sum-queue overflow for any weight vector  $b$ . The p-Log rule takes weight vector  $b$  as a parameter but does not require any knowledge of arrival or server-state distributions.

We will define the p-Log scheduling rule through a vector field  $h = (h_1, h_2)$  on



$\mathbb{R}_+^2$ . For any  $x \in [0, 1)^2$ , let  $h(x) = 0$ . For any  $x \in \mathbb{R}_+^2 \setminus [0, 1)^2$ ,

$$\begin{aligned} & \text{if } x_1 \geq x_2, \text{ then let } \begin{cases} h_1(x) = b_1\sqrt{x_1}, \\ h_2(x) = b_2 \min(x_2, \sqrt{x_1}); \end{cases} \\ & \text{if } x_1 < x_2, \text{ then let } \begin{cases} h_1(x) = b_1 \min(x_1, \sqrt{x_2}), \\ h_2(x) = b_2\sqrt{x_2}. \end{cases} \end{aligned} \quad (3.5)$$

The p-Log rule is given as follows: when the queues are in state  $Q \in \mathbb{Z}_+^2$  and the server in state  $m \in \mathcal{M}$ , then the server is allocated to queue  $i_{pLog}^*(Q, m)$  given by,

$$i_{pLog}^*(Q, m) \in \arg \max_{i \in I} h_i(Q) \mu_i^m, \quad (3.6)$$

where, in the case of a tie, if  $Q_1 \geq Q_2$  the server is allocated to Queue 1, otherwise to Queue 2.

Note that it is only the slope,  $\frac{h_2(Q)}{h_1(Q)}$ , of the vector  $h(Q)$  that determines the scheduling decision, and so, for example, when  $Q_1 \geq Q_2$ ,  $Q_1 \neq 0$ , the slope is given by,

$$\frac{h_2(Q)}{h_1(Q)} = \frac{b_2}{b_1} \min \left( 1, \frac{Q_2}{\sqrt{Q_1}} \right).$$

### Switching curves under the p-Log rule

Let  $v^{pLog}(Q)$  denote the vector of average service offered to the queues under p-Log rule, conditional on queue state being  $Q$ , *i.e.*, the vector obtain by substituting  $i_{pLog}^*$  for  $i^*$  in (3.3). Recall the set of outer normal slopes  $\{r_0, r_1, \dots, r_{M'}, r_{M'+1}\}$  and, in particular, the slopes  $r_k$  and  $r_l$  from this set as defined in Section 3.3. Similar to the sets  $\mathcal{S}_0^{(\cdot)}$  for  $(\cdot) \in \{mw, exp, log\}$ , we define the set  $\mathcal{S}_0$  for p-Log rule as,

$$\begin{aligned} \mathcal{S}_0 &= \left\{ x \in \mathbb{R}_+^2 : r_k < \frac{h_2(x)}{h_1(x)} < r_l \right\}, \\ &= \left\{ x \in \mathbb{R}_+^2 : \frac{b_1}{b_2} r_k \sqrt{x_1} < x_2 < \left( \frac{b_1}{b_2} r_l x_1 \right)^2 \right\}. \end{aligned}$$

Similar to the region  $\mathcal{S}_0^{log}$ , the region  $\mathcal{S}_0$  too is shaped like a French horn, see Fig. 3.2 for an illustration. For any  $Q \in \mathcal{S}_0 \cap \mathbb{Z}_+^2$ , we have  $v^{pLog}(Q) \in \arg \max_{y \in V_\pi} \langle y, b \rangle$ .

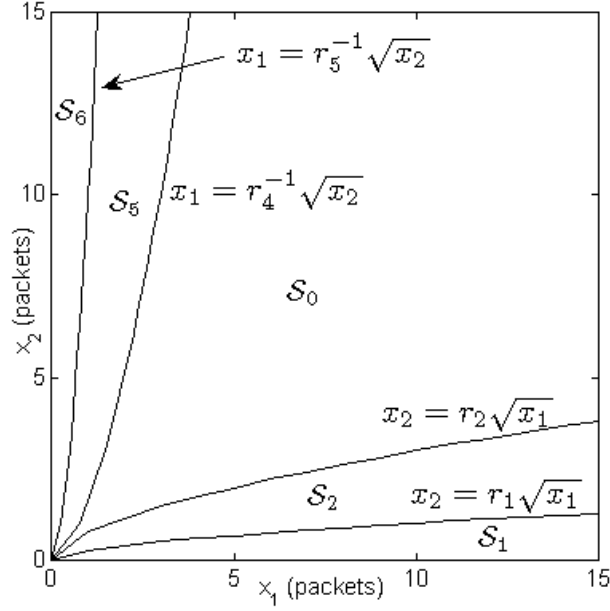


Figure 3.2: Switching curves under p-Log rule for the capacity region depicted in Fig. 3.1-a and weight vector  $b = (1, 1)$ .

Therefore, we will refer to the region  $\mathcal{S}_0$  as the weighted-max-sum rate region (with respect to weight vector  $b$ ). Moreover, for any  $Q > 0$ , we have  $\theta Q \in \mathcal{S}_0$  for all  $\theta$  large enough, thus indicating that p-Log rule is radial sum-rate monotone. Also, for each  $m \in \{1, \dots, k, l+1, \dots, M'+1\}$ , we define a set  $\mathcal{S}_m$  as,

$$\mathcal{S}_m = \left\{ x \in \mathbb{R}_+^2 : r_{m-1} < \frac{h_2(x)}{h_1(x)} < r_m \right\} .$$

Then for any  $Q \in \mathcal{S}_m \cap \mathbb{Z}_+^2$ , we have  $v^{p\text{Log}}(Q) = \hat{v}^m$ . All switching curves in the half plane  $\{x_1 \geq x_2\}$  are given by,

$$x_2 = \frac{b_1}{b_2} r \sqrt{x_1} ,$$

for  $r \in \{r_0, \dots, r_k\}$  and  $x_1 \geq 1$ . Similarly, all switching curves in the half plane  $x_2 > x_1$  are given by,

$$x_1 = \frac{b_2}{b_1} r^{-1} \sqrt{x_2} ,$$

for  $r \in \{r_l, \dots, r_{M'+1}\}$  and  $x_2 \geq 1$ . We will refer to the collection of switching curves and regions  $\mathcal{S}_m$  as a *partition* of  $\mathbb{R}_+^2$  (or the queue state space  $\mathbb{Z}_+^2$ ) under the p-Log

rule. It will be useful to note that this partition, as well as the partitions under MaxWeight, Exp rule, and Log rule, depend only on the set of vectors  $\{\mu^1, \dots, \mu^M\}$  associated with the  $M$  server states and not on the distribution  $\pi = (\pi_1, \dots, \pi_M) > 0$  over these states.

### 3.5 Main result

The following three-part theorem summarizes the main results. It includes a lower bound on the tail of the weighted sum-queue overflow probability, an upper bound on the same, and the optimality of the p-Log rule. The first part is proved in Section 3.7, the second in Section 3.8 (and Appendix), while the last in Sections 3.9 and 3.10.

**Theorem 3.1** *For the system model detailed in Section 3.2, the following hold.*

(i) *Given a weight vector  $b = (b_i : b_i > 0, i \in I)$ , there exists finite  $T_0 > 0$  such that for any  $t > T_0$  and under any scheduling rule starting in any initial state  $Q(0)$ , we have the following lower bound:*

$$\liminf_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( \sum_{i \in I} b_i Q_i(nt) \geq n \right) \geq -J_* ,$$

where  $J_*$  is defined in Section 3.7.

(ii) *For a stabilizable system, under p-Log scheduling rule the process  $(Q(t), t = 0, 1, \dots)$  forms an ergodic Markov chain, and we have the following upper bound for a random vector  $Q$  drawn from the stationary distribution of the Markov chain:*

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( \sum_{i \in I} b_i Q_i \geq n \right) \leq -J_{**} ,$$

where  $J_{**}$  is defined in Section 3.8.

(iii) *The p-Log rule maximizes the asymptotic exponential decay rate of the weighted sum-queue distribution, i.e.,*

$$J_{**} = J_* .$$

**Remark 3.2** *Since a scheduler can, in principle, be non-stationary, the lower bound in (i) is expressed in a more general form than the upper bound in (ii) which is specific to a static state-feedback scheduler, namely the p-Log rule. For stationary schedulers under which the process  $(\mathbf{Q}(t), t = 0, 1, \dots)$  forms an ergodic Markov chain, the lower bound in (i) also implies the same lower bound under the steady state distribution of the Markov chain. This is because the lower bound will hold if the initial state  $Q(0)$  were random and drawn from the steady state distribution.*

### 3.6 Fluid-scaled processes and large deviation principles

In this section, we define sequences of fluid-scaled processes and functions, and a Large Deviation Principle [44] on those sequences, which is used in proving Theorem 3.1. Define the cumulative arrivals process  $\mathbf{F} = (\mathbf{F}(t) = (\mathbf{F}_i(t), i \in I), t \geq 0)$  obtained from the process  $(\mathbf{A}(t), t \geq 0)$  as,

$$\mathbf{F}_i(t) = \sum_{k=0}^{\lfloor t-1 \rfloor} \mathbf{A}_i(k),$$

and cumulative time the channel is in each state  $\mathbf{G} = (\mathbf{G}(t) = (\mathbf{G}_m(t), m \in \mathcal{M}), t \geq 0)$  obtained from the process  $(\mathbf{m}(t), t \geq 0)$  as,

$$\mathbf{G}_m(t) = \sum_{k=0}^{\lfloor t-1 \rfloor} \mathbf{1}_{\{\mathbf{m}(k)=m\}}.$$

The triplet  $(Q, F, G)$ , where  $Q = (Q(t), t \geq 0)$  is the queue sample path (under a fixed scheduling rule) corresponding to the sample paths  $(F, G)$  and initial state  $Q(0)$ , denotes a realization of the system  $(\mathbf{Q}, \mathbf{F}, \mathbf{G})$ . For each  $n = 0, 1, \dots$ , let  $(\mathbf{Q}^{(n)}, \mathbf{F}^{(n)}, \mathbf{G}^{(n)})$  denote an independent and identically distributed system. We define a corresponding sequence of fluid-scaled processes, denoted by  $(\mathbf{q}^{(n)}, \mathbf{f}^{(n)}, \mathbf{g}^{(n)})$ , as,

$$\mathbf{q}^{(n)} = (\mathbf{q}^{(n)}(t), t \geq 0) = \left( \frac{1}{n} \mathbf{Q}^{(n)}(nt), t \geq 0 \right),$$

with  $\mathbf{f}^{(n)}$  and  $\mathbf{g}^{(n)}$  similarly defined.

The arrival and service processes are i.i.d. and bounded and, therefore, satisfy large deviation principles [44]. In particular, for each  $i \in I$ , define for any scalar  $\lambda_i \geq 0$  the rate function  $L_i(\cdot)$  for the sequence  $\mathbf{f}_i^{(n)}(1)$  as,

$$L_i(\lambda_i) = \sup_{\theta \geq 0} ( \theta \lambda_i - \log \mathbb{E} [ e^{\theta A_i(1)} ] ) ,$$

where  $L_i(\cdot) = \infty$  over  $(-\infty, 0)$  and  $(C, \infty)$ . Also, for any vector  $\lambda \in \mathbb{R}_+^2$ , let,

$$L_{(f)}(\lambda) = \sum_{i \in I} L_i(\lambda_i) .$$

For any probability distribution  $\gamma = (\gamma_m, m \in \mathcal{M})$ , we define the relative entropy  $L_{(g)}(\cdot)$  of  $\gamma$  with respect to distribution  $\pi$  as,

$$L_{(g)}(\gamma) = \sum_{m \in \mathcal{M}} \gamma_m \log \frac{\gamma_m}{\pi_m} .$$

where  $L_{(g)}(\cdot) = \infty$  everywhere outside the standard simplex in  $\mathbb{R}_+^M$ . Now consider any functions  $(f, g) \in \mathcal{D}^{2+M}$ . If  $(f, g)$  are absolutely continuous, then they are differentiable *a.e.* and we let  $(f'(t), g'(t)) = \frac{d}{dt}(f(t), g(t))$ . For any  $t > 0$ , if  $(f(0), g(0)) = 0$  and  $(f, g)$  are absolutely continuous on the interval  $[0, t]$ , then let,

$$J_t(f, g) = \int_0^t L_{(f)}(f'(s)) + L_{(g)}(g'(s)) ds ,$$

otherwise  $J_t(f, g) = \infty$ . The functional  $J_t(f, g)$  is referred to as the cost of the trajectories  $(f, g)$  over the time interval  $[0, t]$ . The following is a form of Borovkov/Mogulskii's theorem [44].

**Proposition 3.1** *For any fixed  $T > 0$ , consider a sequence in  $n$  of the fluid-scaled processes  $(\mathbf{f}^{(n)}, \mathbf{g}^{(n)}) = ((\mathbf{f}^{(n)}(t), \mathbf{g}^{(n)}(t)), t \in [0, T])$ , then for any measurable  $B \subseteq \mathcal{D}^{2+M}$ ,*

we have that,

$$\begin{aligned}
& - \inf_{(f,g)} \{ J_T(f,g) | (f,g) \in B^\circ \} \\
& \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}((\mathbf{f}^{(n)}, \mathbf{g}^{(n)}) \in B) \\
& \leq \limsup_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}((\mathbf{f}^{(n)}, \mathbf{g}^{(n)}) \in B) \\
& \leq - \inf_{(f,g)} \{ J_T(f,g) | (f,g) \in \overline{B} \} ,
\end{aligned}$$

where,  $B^\circ$  and  $\overline{B}$  denote the interior and closure of set  $B$  respectively.

Let  $u(n) = \lceil n^\alpha \rceil$  for some fixed  $\alpha \in (0, 0.5)$ . For any function  $d \in \mathcal{D}^{2+M}$ , let  $U^n d$  denote the piece-wise linear function obtained by linear interpolation over samples  $(d(\frac{ku(n)}{n}), k = 0, 1, \dots)$ . The following upper bound is Stolyar's refinement of Mogulskii's theorem and was first introduced in [19].

**Proposition 3.2** *For any fixed  $T > 0$ , consider a sequence in  $n$  of the fluid-scaled processes  $(\mathbf{f}^{(n)}, \mathbf{g}^{(n)}) = ((\mathbf{f}^{(n)}(t), \mathbf{g}^{(n)}(t)), t \in [0, T])$ . Suppose, for each  $n$  there is a fixed measurable  $B^{(n)} \subseteq \mathcal{D}^{2+M}$ , that is a subset of the set of feasible realizations of  $(\mathbf{f}^{(n)}, \mathbf{g}^{(n)})$  in  $[0, T]$ . Then,*

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}((\mathbf{f}^{(n)}, \mathbf{g}^{(n)}) \in B^{(n)}) \leq - \liminf_{n \rightarrow \infty} \inf_{(f,g)} \{ J_{T^{(n)}} U^n(f,g) | (f,g) \in B^{(n)} \} , \tag{3.7}$$

where  $T^{(n)} = \frac{u(n)}{n} \lfloor \frac{nT}{u(n)} \rfloor$ .

Note that by contrast with Proposition 3.1,  $\{B^{(n)}\}$  in Proposition 3.2 corresponds to a sequence of sets of scaled feasible trajectories, and the bound on the right side is an infimum over the cost of sampled and linearly interpolated scaled trajectories. Thus this proposition provides a refinement allowing, for example, the consideration of large deviations for sets  $\{B^{(n)}\}$  which can distinguish among the trajectories in the set  $\{f^{(k)} \in \mathcal{D}^2 : f^{(k)} \text{ converges u.o.c. to a Lipschitz } f\}$ . That is to say,

even if all the fluid scaled trajectories in a set converge to the same limiting trajectory, the events  $B^{(n)}$  can be defined to include only a subset of these trajectories.

Also introduced in [19] is a notion of generalized fluid sample path (GFSP) which naturally appears when applying the bound in Proposition 3.2; we describe this next. Consider a sequence of realizations  $(q^{(n)}, f^{(n)}, g^{(n)})$   $n = 1, 2, \dots$  such that along some subsequence (still denoted by  $\{n\}$ ), we have u.o.c convergence

$$(q^{(n)}, f^{(n)}, g^{(n)}) \rightarrow (q, f, g)$$

to some Lipschitz continuous functions  $(q, f, g)$ , and u.o.c convergence

$$\begin{aligned} \bar{J}^{(n)} &= \left( \bar{J}_t^{(n)}, t \geq 0 \right), \\ &= \left( J_{t^{(n)}} U^n(f^{(n)}, g^{(n)}), t \geq 0 \right) \rightarrow \bar{J} = \left( \bar{J}_t, t \geq 0 \right) \end{aligned}$$

to some non-negative increasing Lipschitz continuous function  $\bar{J}$ . Then the following construct is called a GFSP,

$$\psi = \left[ (q^{(n)}, f^{(n)}, g^{(n)}), \bar{J}^{(n)}, n = 0, 1, \dots; (q, f, g); \bar{J} \right]$$

and the function  $\bar{J}$  is referred to as the refined cost function of the GFSP. Since this construct contains not only the limiting trajectory  $(q, f, g)$  but also the sequence converging to it, the construct is useful when one needs to *zoom in* and study the limiting trajectories  $(q, f, g)$  at a finer-than-fluid scaling. Moreover, we will see that for the events  $B^{(n)}$  of interest to us, the bound in Proposition 3.2 reduces to an infimum over the refined cost of a well-defined set of GFSPs.

### 3.7 Lower bound on overflow probability under any scheduling rule

For any distribution  $\gamma$  on the set  $\mathcal{M}$  of server states, let  $V_\gamma$  denote the corresponding capacity region, see (3.1). Let  $b = (b_1, b_2) > 0$  be the given weight vector.

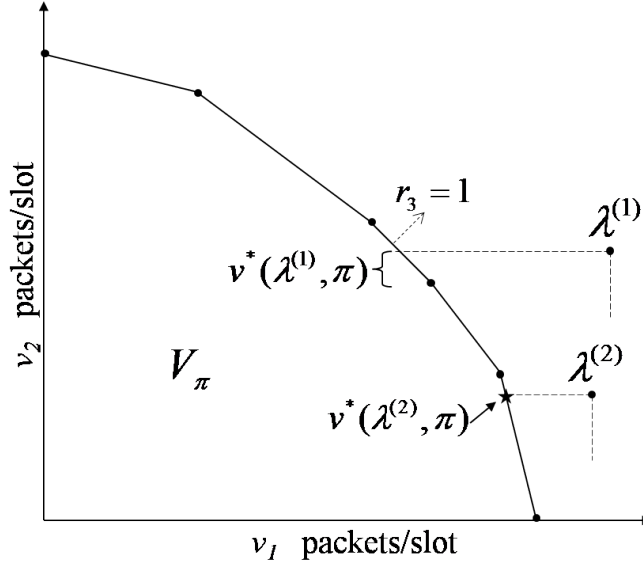


Figure 3.3: For weights  $b_i = 1$ , the *optimal* capacity vector  $v^*(\lambda^{(2)}, \pi)$  is unique, whereas, vector  $v^*(\lambda^{(1)}, \pi)$  is any point on the annotated part of max-sum-rate face of region  $V_\pi$ .

For any vector  $\lambda \in \mathbb{R}_+^2$ , let  $v^*(\lambda, \gamma)$  denote a service vector such that,

$$v^*(\lambda, \gamma) \in \arg \max_v \{ \langle b, v \rangle : v \leq \lambda, v \in V_\gamma \} , \quad (3.8)$$

where, if the argmax is not unique, then  $v^*(\lambda, \gamma)$  can be set to any maximizer. For example, see Fig. 3.3 depicting  $v^*(\lambda, \pi)$  for weight vector  $b = (1, 1)$  and two hypothetical vectors  $\lambda^{(1)}$  and  $\lambda^{(2)}$  lying outside capacity region  $V_\gamma$ . The interpretation is as follows: if the arrival process were to exhibit an empirical mean of  $\lambda$  and the server state were to exhibit an empirical distribution  $\gamma$ , then serving the queues according to the service vector  $v^*(\lambda, \gamma)$  minimizes the rate of weighted-sum-queue build-up, *i.e.*,  $\langle b, \lambda - v^*(\lambda, \gamma) \rangle$ .

Finally, we define the minimum cost (per unit increase of the weighted-sum-queue)  $J_*$  as,

$$J_* = \min_{\gamma, \lambda} \frac{L_{(g)}(\gamma) + L_{(f)}(\lambda)}{\langle b, \lambda - v^*(\lambda, \gamma) \rangle} , \quad (3.9)$$



### 3.7.1 Proof of Theorem 3.1-(i)

Let  $(\lambda^*, \gamma^*)$  be a point that achieves the minimum in (3.9), and let

$$T_0 = \langle b, \lambda^* - v^*(\lambda^*, \gamma^*) \rangle^{-1}.$$

We will show that regardless of the scheduling rule, all realizations  $(f, g)$  sufficiently *close* to  $(\lambda^*t, \gamma^*t)$  over the interval  $[0, T_0]$ – and thus having cost  $J_{T_0}(f, g)$  close to  $J_*$ – lead to an overflow at time  $T_0$ .

Let  $\|\cdot\|$  denote the  $L_\infty$  norm. For any  $\epsilon > 0$ , define a set of trajectories over the interval  $[0, T_0]$ ,

$$B_\epsilon = \left\{ \begin{array}{l} (f, g) \in \mathcal{D}^{2+M} : (f, g) \text{ is absolutely continuous;} \\ \forall t \in [0, T_0], \\ \|L_{(f)}(f'(t)) - L_{(f)}(\lambda^*)\| \leq \frac{\epsilon}{2T_0}; \\ \|L_{(g)}(g'(t)) - L_{(g)}(\gamma^*)\| \leq \frac{\epsilon}{2T_0}; \\ \forall u \in V_{g'(t)}, \inf_{v \in V_{\gamma^*}} \|u - v\| \leq \frac{\epsilon}{5T_0}; \text{ and} \end{array} \right. \quad (3.10)$$

$$\left. \left\| \lambda^* - \frac{1}{T_0} f(T_0) \right\| \leq \frac{\epsilon}{5T_0} \right\} \quad (3.11)$$

The set  $B_\epsilon$  is measurable, compact, and the cost for any  $(f, g) \in B_\epsilon$  satisfies

$$\|J_{T_0}(f, g) - J_*\| \leq \epsilon.$$

Next, we will show that any sequence of realizations  $(q^{(n)}, f^{(n)}, g^{(n)})$ , such that  $q^{(n)}(0) = 0$  and  $(f^{(n)}, g^{(n)})$  converge uniformly in  $[0, T_0]$  to some  $(f, g) \in B_\epsilon$ , must have

$$\sum_{i \in I} b_i q_i^{(n)}(T_0) > 1 - 2\epsilon$$

for all sufficiently large  $n$ . Therefore,

$$\begin{aligned} & \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( \sum_{i \in I} b_i q_i^{(n)}(T_0) > 1 - 2\epsilon \right) \\ & \geq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( (f^{(n)}, g^{(n)}) \in B_\epsilon \right) \geq -J_* , \end{aligned}$$

where the rightmost inequality follows from the Mogulskii's theorem (Proposition 3.1).

Let  $\lambda^{(n)} = \frac{1}{T_0} f^{(n)}(T_0)$ , and  $u^{(n)}$  denote the average service vector seen by the queues; that is to say,  $u_i^{(n)} T_0$  is equal to the number of packets served from the  $i^{\text{th}}$  queue over the interval  $[0, T_0]$ . By (3.11) in the definition of  $B_\epsilon$ , for all  $n$  sufficiently large,

$$\lambda^* - \left( \frac{\epsilon}{4T_0}, \frac{\epsilon}{4T_0} \right) \stackrel{(i)}{\leq} \lambda^{(n)} \stackrel{(ii)}{\leq} \lambda^* + \left( \frac{\epsilon}{4T_0}, \frac{\epsilon}{4T_0} \right), \quad (3.12)$$

and by (3.10) there exists a  $v \in V_{\gamma^*}$  such that

$$v - \left( \frac{\epsilon}{4T_0}, \frac{\epsilon}{4T_0} \right) \stackrel{(i)}{\leq} u^{(n)} \stackrel{(ii)}{\leq} v + \left( \frac{\epsilon}{4T_0}, \frac{\epsilon}{4T_0} \right). \quad (3.13)$$

We must also have  $u^{(n)} \leq \lambda^{(n)}$ , since the total service cannot exceed the total arrivals. This, along with inequalities (3.12-ii) and (3.13-i), implies  $v \leq \lambda^* + \left( \frac{\epsilon}{2T_0}, \frac{\epsilon}{2T_0} \right)$ . Without loss of generality, we assume that the weight vector  $b$  is normalized to have  $\sum_{i \in I} b_i = 2$ . Then using (3.13-ii), we get,

$$\begin{aligned} \langle b, u^{(n)} \rangle &\leq \langle b, v \rangle + \frac{\epsilon}{2T_0}, \\ &\leq \max \left( \langle b, y \rangle : y \in V_{\gamma^*}, y \leq \lambda^* + \left( \frac{\epsilon}{2T_0}, \frac{\epsilon}{2T_0} \right) \right) + \frac{\epsilon}{2T_0}, \\ &\leq \langle b, v^*(\lambda^*, \gamma^*) \rangle + \frac{\epsilon}{T_0}. \end{aligned} \quad (3.14)$$

Finally, using (3.12-i) and (3.14), we have that,

$$\begin{aligned} \langle b, q^{(n)}(T_0) \rangle &= (\langle b, \lambda^{(n)} \rangle - \langle b, u^{(n)} \rangle) T_0, \\ &> (\langle b, \lambda^* \rangle - \langle b, v^*(\lambda^*, \gamma^*) \rangle) T_0 - 2\epsilon, \\ &= 1 - 2\epsilon. \end{aligned}$$

In order to obtain an overflow at any time after  $T_0$  while still incurring a cost close to  $J_*$ , the trajectory  $(\lambda^* t, \gamma^* t)$  can be prepended with the zero-cost trajectory for an appropriate amount of time. ■

### 3.8 Upper bound on overflow probability under the p-Log rule

Recall the notion of a GFSP and its refined cost function from Section 3.6. Let  $J_{**}$  denote the lowest refined cost of a GFSP that, under p-Log rule, raises  $\sum_{i \in I} b_i q_i(t)$  to 1 from the initial state  $q(0) = 0$ , *i.e.*,

$$J_{**} = \inf_{t \geq 0} J_{**,t} , \quad (3.15)$$

where,

$$J_{**,t} = \inf_{\psi} \left\{ \bar{J}_t | \psi : q(0) = 0, \sum_{i \in I} b_i q_i(t) \geq 1 \right\} .$$

The following is a restatement of Theorem 3.1-(ii) in terms of a sequence of fluid scaled queues.

**Theorem 3.2** *For each  $n = 1, 2, \dots$ , consider the system under the p-Log scheduling rule in a stationary regime, then, the corresponding sequence of fluid-scaled processes is such that,*

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left( \sum_{i \in I} b_i \mathbf{q}_i^{(n)}(0) \geq 1 \right) \leq -J_{**} .$$

**Remark 3.3** *This is the equivalent of Theorem 8.4 of [19], and its proof follows the same framework and uses classical Wentzel-Freidlin constructions [45]. The theorem establishes two things: firstly, that the upper bound on the probability of overflow when starting with empty queues, given by Stolyar's refinement of Mogulskii's upper bound, indeed reduces to inf over the cost of GFSPs of interest; and secondly, that a GFSP with the cheapest limiting trajectories  $(f, g)$  that can raise the sum queue  $\sum_{i \in I} b_i q_i$  to 1, starting with empty queues, indeed has a cost arbitrarily close to the cost of the cheapest trajectory starting in the stationary regime. See the appendix at the end of the chapter for a proof.*

It is clear that  $J_{**} \leq J_*$  (since  $-J_*$  was lower bound under any scheduling rule.) To prove the optimality of p-Log rule, we need show  $J_{**} = J_*$ . In the following section, we develop the results needed to show this; these intermediate steps are summarized in Table 3.1.

### 3.9 Local fluid sample path

Let us first motivate the need for defining LFSP (local fluid sample path.) For each  $n$ , define the set  $\mathcal{S}_m^{(n)}$  as the “fluid” scaled version of set  $\mathcal{S}_m$  of the state space partition associated with the p-Log rule, *i.e.*,

$$\mathcal{S}_m^{(n)} = \{x \in \mathbb{R}_+^2 : nx \in \mathcal{S}_m\}.$$

Then for the  $n^{\text{th}}$  system, the scheduling decision at time  $t$  depends on which set  $\mathcal{S}_m^{(n)}$ ,  $m \in \{0, \dots, k, l+1, \dots, M'+1\}$  (or the corresponding switching curve) the fluid scaled queue  $q^{(n)}(t)$  lies in. As  $n \rightarrow \infty$ , the characteristic function of set  $\mathcal{S}_0^{(n)}$  converges pointwise to the characteristic function of  $\mathcal{S}_0^{(\infty)} = \{x \in \mathbb{R}_+^2 : x > 0\}$ ; while all other scaled sets from the partition collapse to one of the axes. Note that this is true for the partition under any radial sum-rate monotone scheduling rule. Now consider a Lipschitz continuous limiting trajectory  $(q, f, g)$  for the fluid scaled process  $(q^{(n)}, f^{(n)}, g^{(n)})$ . One can show that,

$$\text{if } q(t) \in \mathcal{S}_0^{(\infty)}, \text{ then } \frac{d}{dt} \langle b, q(t) \rangle = \langle b, f'(t) \rangle - \max_{v \in V_{g'(t)}} \langle b, v \rangle,$$

but if  $q(t)$  hits an axis, we lose information about service rates of the queues. Hence, we define a LFSP using a *finer*-than-fluid scaling such that the sets of the partition do not collapse and we are able to state the proper derivative of the limiting queue trajectory.

Consider a GFSP over some interval  $[0, T]$  and fix any  $\tau \in (0, T)$  such that  $q(\tau) \neq 0$ . Any sequence  $\tau^{(n)} \rightarrow \tau$  has a subsequence along which  $q^{(n)}(\tau^{(n)}) \rightarrow q(\tau)$ .

Description	Stated in	Relies on
Under the usual fluid scaling of the queue state space, some sets of the partition collapse (or merge), resulting in loss of information concerning the service rate seen by the queue ( <i>e.g.</i> , when the queue lies in any of the collapsed sets.) Therefore, the fluid-scaled state space is locally <i>magnified</i> enough to recover the merged sets of the partition; these magnified or finer-than-fluid-scaled trajectories are called Local Fluid Sample Paths (LFSP).	Section 3.9, Lemma 3.1.	Technique first introduced in [11]
Although the vector field $h(\cdot)$ associated with the p-Log rule is not a gradient field, it appears as a gradient field on the state space of finer-than-fluid-scaled queue. A (globally) Lipschitz continuous Lyapunov function is then constructed on this state space. Moreover, under the p-Log scheduling rule, the Lyapunov function is shown to have a strictly negative drift for all LFSP trajectories having low <i>cost per unit time</i> .	Lemma 3.2	Lemma 3.1
The strictly negative drift of a Lipschitz continuous Lyapunov function translates into a stronger implication: for all LFSP trajectories with low average cost per unit time over a given time interval, the decrease in the Lyapunov function of the finer-than-fluid-scaled queue must be proportional to the length of the interval. Moreover, a sufficient decrease in the Lyapunov function also implies at least a proportional decrease in the weighted-sum-queue.	Lemma 3.3	Lemma 3.2
The above result is used to show that any fluid-scaled trajectory (GFSP) of interest can be <i>magnified</i> to obtain a finer-than-fluid-scaled trajectory (LFSP) such that the cost (per unit increase in weighted sum-queue) of the fluid-scaled trajectory and that of the finer-than-fluid-scaled trajectory are arbitrarily close.	Lemma 3.4	Lemma 3.3 and using technique of Section 11 of [19]
Under the p-Log rule, no LFSP exists whose <i>cost per unit increase in weighted-sum-queue</i> is strictly less than $J_*$ , therefore, the least possible cost under p-Log rule, <i>i.e.</i> $J_{**}$ , must be equal to $J_*$ – the upper bound on the cost under any scheduler.	Section 10	Lemma 3.4

Table 3.1: Intermediate steps towards proving Theorem 1-(iii).

Let

$$\sigma_n = \sqrt{q_*^{(n)}(\tau^{(n)})}/\sqrt{n},$$

where,  $q_*^{(n)}(\cdot) = \max_{i \in I} q_i^{(n)}(\cdot)$ . To obtain a local fluid sample path, we will *magnify* in both space and time the fluid scaled trajectories  $(q^{(n)}, f^{(n)}, g^{(n)})$  by a factor of  $\sigma_n^{-1}$ , *i.e.*, an order  $O(\sqrt{n})$  term. More formally, for any fixed  $S > 0$ , the following re-scaled functions (and their limits mentioned subsequently) over the interval  $[\tau^{(n)}, \tau^{(n)} + \sigma_n S]$ , parameterized by  $s \in [0, S]$ , are called the local fluid sample paths<sup>3</sup>: for all  $i \in I$  and  $m \in \mathcal{M}$ ,

$$\begin{aligned} \diamond q_i^{(n)}(s) &= \frac{1}{\sigma_n} \left( q_i^{(n)}(\tau^{(n)} + \sigma_n s) - q_i^{(n)}(\tau^{(n)}) \right), \\ \diamond \hat{q}_i^{(n)}(s) &= \frac{1}{\sigma_n} q_i^{(n)}(\tau^{(n)} + \sigma_n s), \\ \diamond d^{(n)}(s) &= \diamond \hat{q}_1^{(n)}(s) - \diamond \hat{q}_2^{(n)}(s), \\ \diamond f_i^{(n)}(s) &= \frac{1}{\sigma_n} \left( f_i^{(n)}(\tau^{(n)} + \sigma_n s) - f_i^{(n)}(\tau^{(n)}) \right), \\ \diamond g_m^{(n)}(s) &= \frac{1}{\sigma_n} \left( g_m^{(n)}(\tau^{(n)} + \sigma_n s) - g_m^{(n)}(\tau^{(n)}) \right). \end{aligned}$$

Then along some subsequence in  $n$ ,

- the functions  $(\diamond q^{(n)}, \diamond f^{(n)}, \diamond g^{(n)})$  converge uniformly over  $[0, S]$  to Lipschitz continuous functions  $(\diamond q, \diamond f, \diamond g)$ ;
- for each  $i \in I$ , the function  $\diamond \hat{q}_i^{(n)}$  either converges uniformly over  $[0, S]$  to a finite Lipschitz continuous function  $\diamond \hat{q}_i$ , or is identically equal to  $\infty$ ;
- the function  $\diamond d^{(n)}$  converges uniformly over  $[0, S]$  to a finite Lipschitz continuous function  $\diamond d$ , or is identically equal to  $+\infty$  or  $-\infty$ .

---

<sup>3</sup>The definitions of  $\diamond f_i^{(n)}(\cdot)$  and  $\diamond g_m^{(n)}(\cdot)$  are the same as in [19], whereas,  $\diamond q_i^{(n)}(\cdot)$  are *scaled* as in [19] but *centered* differently.

We will refer to the point  $(q(\tau), f(\tau), g(\tau))$  as the *GFSP source point* of the above defined LFSP. Since  $q(\tau) \neq 0$ , it must be that  $\diamond\hat{q}_i(\cdot) = \infty$  for at least one  $i \in I$ . Note that the local fluid queue,  $\diamond q$ , is merely a *re-centered* version of  $\diamond\hat{q}$ , *i.e.*,  $\diamond\hat{q}(s) = \diamond q(s) + \diamond\hat{q}(0)$ , and is always finite by virtue of this re-centering. Moreover, the trajectory  $\diamond q$  dwells in the set  $\{x \in \mathbb{R}^2 | x \geq -\diamond\hat{q}(0)\}$ , which is at least a half-plane. Lastly, we have the following relation between the *cost* of LFSP over  $[0, S]$  and the refined cost sequence of GFSP over  $[\tau^{(n)}, \tau^{(n)} + \sigma_n S]$  (see (9.4) of [19]),

$$J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g) \leq \liminf_{n \rightarrow \infty} \frac{1}{\sigma_n} (\bar{J}_{\tau^{(n)} + \sigma_n S}^{(n)} - \bar{J}_{\tau^{(n)}}^{(n)}) . \quad (3.16)$$

### 3.9.1 Scheduling over time scales of LFSP

By (3.6) the scheduling decision in the interval  $[\tau^{(n)}, \tau^{(n)} + \sigma_n S]$  depends on the slope,

$$\frac{h_2(Q(n\tau^{(n)} + n\sigma_n s))}{h_1(Q(n\tau^{(n)} + n\sigma_n s))} ,$$

and the sign of  $\diamond d^{(n)}(s)$ , recall the tie-breaking rule mentioned in the description of p-Log rule in Section 3.4.

Without loss of generality suppose  $q_1(\tau) \geq q_2(\tau)$  (and recall that we had  $q(\tau) \neq 0$ .) Then by (3.5), for  $n$  large enough, we have that,

$$\begin{aligned} \frac{h_2(Q(n\tau^{(n)} + n\sigma_n s))}{h_1(Q(n\tau^{(n)} + n\sigma_n s))} &= \frac{b_2}{b_1} \min \left( 1, \frac{Q_2(n\tau^{(n)} + n\sigma_n s)}{\sqrt{Q_1(n\tau^{(n)} + n\sigma_n s)}} \right) , \\ &= \frac{b_2}{b_1} \min \left( 1, \frac{\diamond\hat{q}_2^{(n)}(s) \sqrt{q_1^{(n)}(\tau^{(n)})}}{\sqrt{q_1^{(n)}(\tau^{(n)} + \sigma_n s)}} \right) . \end{aligned}$$

Then, as  $n \rightarrow \infty$ , the above converges to,

$$\frac{b_2}{b_1} \min (1, \diamond q_2(s) + \diamond\hat{q}_2(0)) ,$$

where the convergence is uniform on  $[0, S]$ . Let us define a vector field  $\diamond h$  over the

state space of  $\diamond q$ , *i.e.*,  $\{x \in \mathbb{R}^2 | x \geq -\diamond \hat{q}(0)\}$ , as follows:

$$\begin{aligned} \diamond h_1(x) &= b_1 , \\ \diamond h_2(x) &= b_2 \min(1, x_2 + \diamond \hat{q}_2(0)) . \end{aligned} \tag{3.17}$$

That is, we can restate the above convergence as,

$$\frac{h_2(Q(n\tau^{(n)} + n\sigma_n s))}{h_1(Q(n\tau^{(n)} + n\sigma_n s))} \rightarrow \frac{\diamond h_2(\diamond q(s))}{\diamond h_1(\diamond q(s))}$$

uniformly on  $[0, S]$ . Hence, the switching curves on the space of  $\diamond q$  are given by,

$$\begin{aligned} &\left\{ (x_1, x_2) \in \mathbb{R}^2 : \frac{\diamond h_2(x)}{\diamond h_1(x)} = r_m \right\} \\ &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : x_2 = -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_m \right\} \end{aligned}$$

for  $m \in \{0, \dots, k\}$ , and are now parallel to  $x_1$  axis (*i.e.*, the axis of  $\diamond q_1(s)$ , see Fig. 3.4)<sup>4</sup>. For each  $m \in \{1, \dots, k\}$ , define the set  $\diamond \mathcal{S}_m$  by appropriately re-scaling the set  $\mathcal{S}_m$ , *i.e.*,

$$\begin{aligned} \diamond \mathcal{S}_m &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : r_{m-1} < \frac{\diamond h_2(x)}{\diamond h_1(x)} < r_m \right\} , \\ &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_{m-1} < x_2 < -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_m \right\} , \end{aligned} \tag{3.18}$$

and define,

$$\begin{aligned} \diamond \mathcal{S}_0 &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : r_k < \frac{\diamond h_2(x)}{\diamond h_1(x)} \right\} , \\ &= \left\{ (x_1, x_2) \in \mathbb{R}^2 : -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_k < x_2 \right\} . \end{aligned} \tag{3.19}$$

See Fig. 3.4 for a graphical illustration of sets  $\diamond \mathcal{S}_0, \dots, \diamond \mathcal{S}_k$ . The service rate  $\mu(s)$  allocated to the local fluid queue  $\diamond q(s)$  at time  $s$ , depends on which re-scaled set  $\diamond \mathcal{S}_m$  (or the associated switching curve) the local fluid queue  $\diamond q(s)$  lies in. More formally, the following lemma relates the service rate  $\mu(s)$  to the vector field  $\diamond h(\diamond q(s))$ , and

---

<sup>4</sup>Recall that we have assumed, without loss of generality, that  $q_1(\tau) \geq q_2(\tau)$ ; had we assumed otherwise, we would have found that the switching curves on the space of  $\diamond q$ , *i.e.*,  $\{x \in \mathbb{R}^2 | x \geq -\diamond \hat{q}(0)\}$  are parallel to the  $x_2$  axis (*i.e.*, the axis of  $\diamond q_2(s)$ ). In the sequel, we continue to assume that at the GFSP source point, we have  $q_1(\tau) \geq q_2(\tau)$ , and therefore,  $\diamond h$  is given by (3.17).



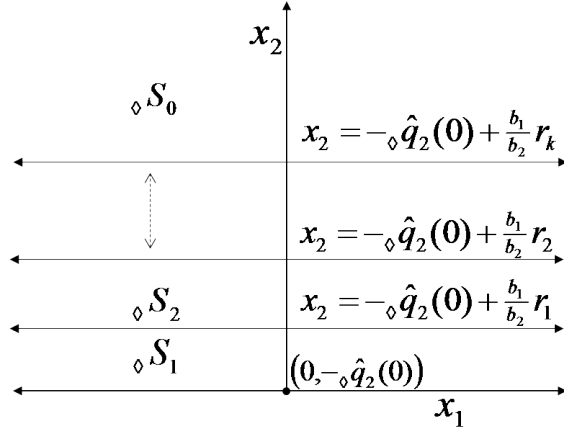


Figure 3.4: Partitions and switching curves on the space of local fluid queue  $\diamond q$ .

can be derived without much effort from the results shown for the Exponential rule scheduler in [11]<sup>5</sup>.

**Lemma 3.1** *For any LFSP, the following derivatives exist a.e. in  $[0, S]$  and are finite:*

$$\begin{aligned}\lambda(s) &= \frac{d}{ds} \diamond f(s) , \\ \gamma(s) &= \frac{d}{ds} \diamond g(s) , \\ \frac{d}{ds} \diamond q(s) &= \lambda(s) - \mu(s) ,\end{aligned}$$

for some,

$$\mu(s) \in \arg \max_{v \in V_{\gamma(s)}} \langle \diamond h(\diamond q(s)), v \rangle . \quad (3.20)$$

**Remark 3.4** *Note that if  $\diamond q_2(s) \geq -\diamond \hat{q}_2(0) + 1$ , then  $\diamond h(\diamond q(s)) = b$ , and the argmax in (3.20) may not be unique. However,  $\mu(s)$  can still be uniquely identified as long as  $\diamond d(s) \neq 0$ . That is, if  $\diamond d(s) > 0$ , then  $\mu(s)$  is such that  $\mu_1(s)$  is the largest possible among all points achieving max in (3.20); similarly, if  $\diamond d(s) < 0$ , then  $\mu(s)$  is such that  $\mu_2(s)$  is the largest. The argmax in (3.20) may again be non-unique if  $\diamond q(s)$*

<sup>5</sup>Specifically, Proposition 1 and results leading from (34) to (35) of [11] relate the service rate seen by the local fluid queue under Exponential rule scheduler to the vector field defining the scheduler.

lies on a switching curve, i.e.,  $\diamond q_2(s) = -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_m$  for some  $m \in \{0, 1, \dots, r_k\}$ , however, in this case  $\mu(s)$  can not be uniquely identified using only the components of LFSP, nor will we need to uniquely identify  $\mu(s)$ .

We will also need the following two crucial lemmas that show for the p-Log rule what Lemmas 9.2 and 9.3 of [19] show for the Exponential rule. The following two lemmas also implicitly prove the throughput-optimality of p-Log rule, see the appendix at the end of the chapter for details.

**Lemma 3.2** *There exist fixed constants  $\epsilon_1 > 0$  and  $\delta_1 > 0$ , and a Lipschitz continuous Lyapunov function  $H$  (constructed in the proof below) such that for any regular point  $s \in [0, S]$ , if*

$$\frac{d}{ds} J_s(\diamond f, \diamond g) \leq \epsilon_1, \quad \text{then} \quad \frac{d}{ds} H(\diamond q(s)) \leq -\delta_1.$$

**Proof** If  $\frac{d}{ds} J_s(\diamond f, \diamond g)$  is small, then  $\lambda(s)$  must be close to the mean arrival rate  $\bar{\lambda}$ , and  $\gamma(s)$  close to the server state distribution  $\pi$ . Since the capacity region  $V_\gamma$  is continuous in the distribution  $\gamma$  (see (3.1)), there exist vectors  $\lambda^* < v^*$  such that uniformly on all sufficiently small values of  $\epsilon_1$ , we have  $\lambda(s) < \lambda^*$ , and  $v^*$  lies in the interior of  $V_{\gamma(s)}$ . Let  $\delta_1 = \min_{i \in I} (v_i^* - \lambda_i^*)$ . By Lemma 3.1, we have,

$$\begin{aligned} \langle \diamond h(\diamond q(s)), \mu(s) \rangle &= \arg \max_{v \in V_{\gamma(s)}} \langle \diamond h(\diamond q(s)), v \rangle, \\ &> \langle \diamond h(\diamond q(s)), v^* \rangle. \end{aligned} \tag{3.21}$$

Note that since  $\diamond h_i(x)$  is a function only of  $x_i$ , the vector field  $\diamond h$  is in fact a gradient field associated with a (continuously differentiable) function<sup>6</sup>  $H$ , i.e.,  $\nabla H = \diamond \hat{h}$  on  $\{x \in \mathbb{R}^2 | x \geq -\diamond \hat{q}(0)\}$ . Moreover, since  $\diamond h_i(\cdot)$  is increasing, positive, and bounded, therefore, the function  $H(\cdot)$  is convex, increasing in each direction, and Lipschitz continuous. Then we have,

$$\frac{d}{ds} H(\diamond q(s)) = \langle \diamond h(\diamond q(s)), \diamond q'(s) \rangle,$$

$$\begin{aligned}
&= \langle \diamond h(\diamond q(s)), \lambda(s) - \mu(s) \rangle , \\
&\leq \langle \diamond h(\diamond q(s)), \lambda^* - v^* \rangle , \\
&\leq -\delta_1 ,
\end{aligned}$$

where the first inequality follows from (3.21) and the second from the definition of  $\delta_1$ .

■

**Lemma 3.3** *There exist fixed constants  $\epsilon_2 > 0$  and  $\hat{\delta}_2 > \delta_2 > 0$  such that, if*

$$J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g) \leq \epsilon_2 S ,$$

then,

$$H(\diamond q(S)) - H(\diamond q(0)) \leq -\hat{\delta}_2 S ,$$

which further implies that uniformly for all large  $S$ , we have the following bound on the change in the weighted sum queue,

$$\langle b, \diamond q(S) - \diamond q(0) \rangle \leq -\delta_2 S.$$

**Proof** Noting that  $H(\diamond q(s))$  (as a function of  $s$ ) is Lipschitz continuous with some Lipschitz constant denoted by  $c$ , the proof of first statement is identical to that of Lemma 9.3 of [19]: pick a positive  $\epsilon_2 < \epsilon_1$ , let  $B_1 = \{s \in [0, S] : \frac{d}{ds} J_s(\diamond f, \diamond g) \geq \epsilon_1\}$  and  $B_2 = S \setminus B_1$ . Then the Lebesgue measures of  $B_1$  and  $B_2$  satisfy,  $\nu(B_1) \leq \frac{\epsilon_2}{\epsilon_1} S$  and  $\nu(B_2) \geq (1 - \frac{\epsilon_2}{\epsilon_1}) S$ . Finally,

$$\begin{aligned}
H(\diamond q(S)) - H(\diamond q(0)) &= \int_{B_1} \frac{d}{ds} H(\diamond q(s)) + \int_{B_2} \frac{d}{ds} H(\diamond q(s)) , \\
&\leq c \frac{\epsilon_2}{\epsilon_1} S - \delta_1 (1 - \frac{\epsilon_2}{\epsilon_1}) S , \\
&= -S (\delta_1 - \frac{\epsilon_2}{\epsilon_1} (c + \delta_1)) .
\end{aligned}$$

---

<sup>6</sup>General conditions for a vector field to form a gradient field can be found in, *e.g.*, [46, pp. 944-945]. In our case,  $H(x)$  is simply additive separable, *i.e.*,  $H(x) \equiv H_1(x_1) + H_2(x_2)$ .

Fix any positive  $\hat{\delta}_2 < \delta_1$ , then a sufficiently small  $\epsilon_2$  can be chosen to prove the first statement.

To prove the last statement, we proceed as follows. Without loss of generality suppose that the GFSP source point of the LFSP being considered satisfies  $q_1(\tau) \geq q_2(\tau)$ , and therefore,  $\diamond h$  is given by (3.17). That is,  $\nabla_1 H(\cdot) \equiv \diamond h_1(\cdot) = b_1$  and  $\nabla_2 H(\cdot) \equiv \diamond h_2(\cdot) \leq b_2$ . Then for any  $x \geq y \geq \diamond \hat{q}(0)$ , we have the upper bound

$$H(x) - H(y) \leq \langle b, x - y \rangle . \quad (3.22)$$

If  $y_2 \geq -\diamond \hat{q}_2(0) + 1$  (see Fig. 3.4), then for all  $z \geq y$ , we have  $\diamond h(z) = b$  and therefore,

$$H(x) - H(y) = \langle b, x - y \rangle . \quad (3.23)$$

If  $-\diamond \hat{q}_2(0) \leq y_2 < -\diamond \hat{q}_2(0) + 1$ , then we have the lower bound

$$\begin{aligned} H(x) - H(y) &= \int_{y_1}^{x_1} \nabla_1 H(z_1, y_2) dz_1 + \int_{y_2}^{x_2} \nabla_2 H(x_1, z_2) dz_2 , \\ &\geq b_1(x_1 - y_1) + \int_{-\diamond \hat{q}_2(0)+1}^{x_2} b_2 dz_2 , \\ &= b_1(x_1 - y_1) + \int_{y_2}^{x_2} b_2 dz_2 - \int_{y_2}^{-\diamond \hat{q}_2(0)+1} b_2 dz_2 , \\ &\geq \langle b, x - y \rangle - b_2 . \end{aligned} \quad (3.24)$$

Combining (3.22–3.24), we have the following bounds for any  $x \geq y \geq \diamond \hat{q}(0)$ ,

$$\langle b, x - y \rangle - (b_1 + b_2) \leq H(x) - H(y) \leq \langle b, x - y \rangle .$$

Using this we get,

$$\begin{aligned} -\hat{\delta}_2 S &\geq H(\diamond q(S)) - H(\diamond q(0)) , \\ &\geq \langle b, \diamond q(S) - \diamond q(0) \rangle - (b_1 + b_2) \\ \Rightarrow -\hat{\delta}_2 S \left( 1 - \frac{b_1 + b_2}{S} \right) &\geq \langle b, \diamond q(S) - \diamond q(0) \rangle . \end{aligned}$$

Now one can take  $S$  large enough to obtain a  $\delta_2$ . ■

Next, using Lemma 3.3 above and identical to the result in Section 11 of [19], we show that any GFSP that raises the weighted sum queue to unity contains a LFSP with cost *close* to that of the GFSP. Subsequently, we will use this lemma on a GFSP of cost close to  $J_{**}$  (see (3.15)) in order to obtain a LFSP with cost close to  $J_{**}$ .

**Lemma 3.4** *Suppose a GFSP  $\psi$  is given that satisfies*

$$q(0) = 0 \quad \text{and} \quad \langle b, q(T) \rangle = 1 \quad \text{for some } T > 0,$$

*and has a cost  $\bar{J}_T < \infty$ . Then, for an arbitrarily small  $\epsilon > 0$ , an LFSP  $(\diamond q, \diamond \hat{q}, \diamond d, \diamond f, \diamond g)$  over an arbitrarily large interval  $[0, S]$  can be constructed from the elements of  $\psi$ , such that,*

$$\langle b, \diamond q(S) - \diamond q(0) \rangle \geq \theta S, \quad (3.25)$$

*for some  $\theta > 0$  (independent of  $\epsilon$ ), and cost (per unit increase in weighted sum queue) of this LFSP is bounded above by  $\bar{J}_T + \epsilon$ , i.e.,*

$$\frac{J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S) - \diamond q(0) \rangle} \leq \bar{J}_T + \epsilon. \quad (3.26)$$

**Proof** Components of the given GFSP  $\psi$  satisfy,

$$\frac{(\bar{J}_T - \bar{J}_0)}{\langle b, q(T) - q(0) \rangle} = \bar{J}_T. \quad (3.27)$$

For any  $0 < \xi_1 < \xi_2 < 1$ , define times  $t_1 < t_2$  as follows,

$$\begin{aligned} t_1 &= \max(t : \langle b, q(t) \rangle = \xi_1), \\ t_2 &= \min(t > t_1 : \langle b, q(t) \rangle = \xi_2). \end{aligned}$$

Then for any  $\epsilon > 0$ , there must exist  $0 < \xi_1 < \xi_2 < 1$  such that,

$$\frac{\bar{J}_{t_2} - \bar{J}_{t_1}}{\langle b, q(t_2) - q(t_1) \rangle} < \bar{J}_T + \frac{\epsilon}{2}, \quad (3.28)$$

or (3.27) cannot hold (Dirichlet's box principle). Fix an  $S > 0$  large enough as required by Lemma 3.3. Then there exists a sequence  $\{\tau^{(n)}\}$  within  $[t_1, t_2]$  such that,

$$\begin{aligned} \langle b, q^{(n)}(\tau^{(n)} + \sigma_n S) - q^{(n)}(\tau^{(n)}) \rangle &> 0, \\ \frac{\bar{J}_{(\tau^{(n)} + \sigma_n S)}^{(n)} - \bar{J}_{\tau^{(n)}}^{(n)}}{\langle b, q^{(n)}(\tau^{(n)} + \sigma_n S) - q^{(n)}(\tau^{(n)}) \rangle} &< \bar{J}_T + \epsilon, \end{aligned}$$

where, as before,  $\sigma_n = \sqrt{q_*^{(n)}(\tau^{(n)})}/\sqrt{n}$ ; such a sequence must exist, otherwise (3.28) cannot hold (another application of Dirichlet's box principle, and consequence of the convergence of  $\bar{J}^{(n)}$  and  $q^{(n)}$ ). The above two inequalities can be re-written as,

$$\langle b, \diamond q^{(n)}(S) - \diamond q^{(n)}(0) \rangle > 0, \quad (3.29)$$

$$\frac{\sigma_n^{-1} (\bar{J}_{(\tau^{(n)} + \sigma_n S)}^{(n)} - \bar{J}_{\tau^{(n)}}^{(n)})}{\langle b, \diamond q^{(n)}(S) - \diamond q^{(n)}(0) \rangle} < \bar{J}_T + \epsilon. \quad (3.30)$$

We also have that  $q(\cdot) \neq 0$  over interval  $[t_1, t_2]$ . Now we can pick a subsequence in  $n$  along which  $\tau^{(n)} \rightarrow \tau \in [t_1, t_2]$  and  $(\diamond q^{(n)}, \diamond \hat{q}^{(n)}, \diamond d^{(n)}, \diamond f^{(n)}, \diamond g^{(n)})$  converge to  $(\diamond q, \diamond \hat{q}, \diamond d, \diamond f, \diamond g)$ , as described in Section 3.9, thus obtaining a LFSP (with  $(q(\tau), f(\tau), g(\tau))$  being its GFSP source point.) From (3.29) we also have that  $\langle b, \diamond q(S) - \diamond q(0) \rangle \geq 0$ . Then by Lemma 3.3, we get,

$$J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g) > \epsilon_2 S,$$

which, alongside (3.30) and (3.16), further implies that for some  $\theta > 0$ ,

$$\langle b, \diamond q(S) - \diamond q(0) \rangle \geq \theta S,$$

and finally,

$$\frac{J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S) \rangle - \langle b, \diamond q(0) \rangle} \leq \bar{J}_T + \epsilon.$$

■

### 3.10 Proof of Theorem 3.1-(iii): optimality of the p-Log rule

Recall that to prove optimality of p-Log rule (Theorem 3.1-(iii)), we need show  $J_{**} \geq J_*$ . We will do this by showing that assuming  $J_{**} < J_*$  leads to a contradiction with the definition of  $J_*$ .

Suppose  $J_{**} < J_*$ , then by definition of  $J_{**}$  in (3.15), there exists a GFSP  $\psi$  satisfying  $q(0) = 0$ ,  $\langle b, q(T) \rangle = 1$  for some finite  $T > 0$ , and having a cost  $\bar{J}_T < J_*$ . Then by Lemma 3.4, from the components of GFSP  $\psi$ , we can construct an LFSP  $(\diamond q, \diamond \hat{q}, \diamond d, \diamond f, \diamond g)$  satisfying (3.25) and (3.26) for an arbitrarily large  $S$  and an  $\epsilon$  small enough so that  $J_{***} \equiv \bar{J}_T + \epsilon < J_*$ .

Since  $\diamond \hat{q}_i(\cdot) = \infty$  for at least one  $i \in I$ , without loss of generality suppose  $\diamond \hat{q}_1(\cdot) = \infty$ , thus all switching curves on the space of  $\diamond q$  are parallel to  $\diamond q_1$  axis. Recall that the lower boundary of the set  $\diamond \mathcal{S}_0$  is given by the switching curve  $x_2 = -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_k$  (see Fig. 3.4). Let  $S_1$  and  $S_2$  respectively be the first and the last time in  $[0, S]$  such that the trajectory  $\diamond q_2(s) \leq -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_k$ , with  $S_1 = S_2 = S$  if  $\diamond q_2(s)$  never hits  $[-\diamond \hat{q}_2(0), -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_k]$ . Note that the trajectory of  $\diamond q(s)$  lies in  $\diamond \mathcal{S}_0$  in  $(0, S_1)$  and  $(S_2, S)$ . Then one of the following must be true over the interval  $[0, S_1]$  (similarly  $[S_2, S]$ ):

1.  $\langle b, \diamond q(0) \rangle < \langle b, \diamond q(S_1) \rangle$  and the cost per unit increase in weighted-sum-queue over the interval  $[0, S_1]$  is less than  $J_{***}$ , *i.e.*,

$$\frac{J_{S_1}(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S_1) - \diamond q(0) \rangle} \leq J_{***} .$$

2.  $\langle b, \diamond q(0) \rangle < \langle b, \diamond q(S_1) \rangle$  and the cost per unit increase in sum-queue over the interval  $[0, S_1]$  is strictly greater than  $J_{***}$ , *i.e.*,

$$\frac{J_{S_1}(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S_1) - \diamond q(0) \rangle} > J_{***} .$$

3.  $\langle b, \diamond q(0) \rangle \geq \langle b, \diamond q(S_1) \rangle$ .

If (a) is true for either one of the intervals (suppose its true for  $[0, S_1]$ ), we proceed as follows: define vectors  $\hat{\gamma}$ ,  $\hat{\lambda}$ , and  $\hat{\mu}$  as the *average* server state distribution, arrival rate, and service rate respectively over  $[0, S_1]$ , *i.e.*,

$$\left(\hat{\lambda}, \hat{\gamma}, \hat{\mu}\right) = \frac{1}{S_1} \int_0^{S_1} (f'(s), g'(s), \mu(s)) ds.$$

By Lemma 3.1 and the fact that  $(\diamond q(s) : 0 < s < S_1)$  lies in  $\diamond \mathcal{S}_0$ , we have that

$$\mu(s) \in \arg \max_{v \in V_{\gamma(s)}} \langle b, v \rangle.$$

This and the *linearity* of  $\mu(s)$  in  $\gamma(s)$  (see (3.1)) implies  $\hat{\mu} \in \arg \max_{v \in V_{\hat{\gamma}}} \langle b, v \rangle$ . Then,

$$\begin{aligned} \langle b, \diamond q(S_1) - q(0) \rangle &= \langle b, \hat{\lambda} - \hat{\mu} \rangle S_1, \\ &\leq \langle b, \hat{\lambda} - v^*(\hat{\lambda}, \hat{\gamma}) \rangle S_1, \end{aligned} \quad (3.31)$$

where  $v^*(\hat{\lambda}, \hat{\gamma})$  is as defined in (3.8). Finally,

$$\begin{aligned} J_* > J_{***} &\geq \frac{J_{S_1}(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S_1) - \diamond q(0) \rangle}, \\ &\geq \frac{\left(L_{(f)}(\hat{\lambda}) + L_{(g)}(\hat{\gamma})\right) S_1}{\langle b, \diamond q(S_1) - \diamond q(0) \rangle}, \\ &\geq \frac{L_{(f)}(\hat{\lambda}) + L_{(g)}(\hat{\gamma})}{\langle b, \hat{\lambda} - v^*(\hat{\lambda}, \hat{\gamma}) \rangle}, \end{aligned}$$

where the first inequality follows from the assumption that (a) is true, the second from convexity of rate functions, and the last one from (3.31). However, by the definition of  $J_*$  in (3.9), the right side of last inequality cannot be less than  $J_*$ , giving the contradiction we needed; therefore, we must have  $J_* = J_{**}$ .

Now, if (a) is not true for both intervals  $[0, S_1]$  and  $[S_2, S]$ , then we proceed as follows. Recall that our LFSP satisfies (3.25) and (3.26), *i.e.*,

$$\langle b, \diamond q(S) - \diamond q(0) \rangle \geq \theta S > 0, \quad (3.32)$$



and,

$$\begin{aligned}
J_{***} &\geq \frac{J_S(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S) - \diamond q(0) \rangle}, \\
&= \frac{J_{S_1}(\diamond f, \diamond g) - J_0(\diamond f, \diamond g)}{\langle b, \diamond q(S_1) - \diamond q(0) \rangle} \times \frac{\langle b, \diamond q(S_1) - \diamond q(0) \rangle}{\langle b, \diamond q(S) - \diamond q(0) \rangle} + \\
&\quad \frac{J_{S_2}(\diamond f, \diamond g) - J_{S_1}(\diamond f, \diamond g)}{\langle b, \diamond q(S_2) - \diamond q(S_1) \rangle} \times \frac{\langle b, \diamond q(S_2) - \diamond q(S_1) \rangle}{\langle b, \diamond q(S) - \diamond q(0) \rangle} + \\
&\quad \frac{J_S(\diamond f, \diamond g) - J_{S_2}(\diamond f, \diamond g)}{\langle b, \diamond q(S) - \diamond q(S_2) \rangle} \times \frac{\langle b, \diamond q(S) - \diamond q(S_2) \rangle}{\langle b, \diamond q(S) - \diamond q(0) \rangle}.
\end{aligned} \tag{3.33}$$

Therefore, if (a) is not true for both intervals  $[0, S_1]$  and  $[S_2, S]$  (equivalently, (b) and/or (c) are true over these intervals), then for (3.33) to hold, we must have,

$$J_{***} \geq \frac{J_{S_2}(\diamond f, \diamond g) - J_{S_1}(\diamond f, \diamond g)}{\langle b, \diamond q(S_2) - \diamond q(S_1) \rangle}, \tag{3.34}$$

and, for some fixed  $\theta_1 > 0$ ,

$$\langle b, \diamond q(S_2) - \diamond q(S_1) \rangle \geq \theta_1 \langle b, \diamond q(S) - \diamond q(0) \rangle.$$

The above, along with (3.32) gives,

$$\langle b, \diamond q(S_2) - \diamond q(S_1) \rangle \geq \theta_1 \theta S. \tag{3.35}$$

Moreover, since  $\langle b, \diamond q(\cdot) \rangle$  is Lipschitz, for some fixed  $\theta_2 > 0$ ,

$$S_2 - S_1 \geq \theta_2 \langle b, \diamond q(S_2) - \diamond q(S_1) \rangle \geq \theta_2 \theta_1 \theta S.$$

This, along with (3.35) and Lemma 3.3, imply,

$$\begin{aligned}
J_{S_2}(\diamond f, \diamond g) - J_{S_1}(\diamond f, \diamond g) &\geq \epsilon_2 (S_2 - S_1), \\
&\geq \epsilon_2 \theta_2 \theta_1 \theta S.
\end{aligned} \tag{3.36}$$

Subsequently, we will use (3.35) and (3.36) to make the quantities on the left side of these inequalities as large as needed by choosing a large  $S$ .

Consider the trajectory,

$$(\diamond q(s), s \in [S_1, S_2]),$$

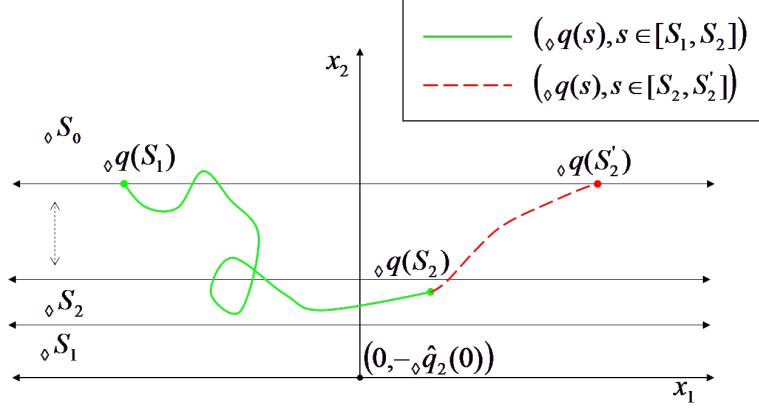


Figure 3.5: Illustration of original trajectory  $(\diamond q(s), s \in [S_1, S_2])$  and the extension  $(\diamond q(s), s \in [S_2, S'_2])$ , in order to obtain  $\diamond q_2(S'_2) = \diamond q_2(S_1)$ .

which is an element of the LFSP  $(\diamond q, \diamond \hat{q}, \diamond d, \diamond f, \diamond g)$  over  $[S_1, S_2]$ . For some  $S'_2 \geq S_2$ , we can append to this LFSP an *extension* over the time interval  $[S_2, S'_2]$  so as to obtain  $\diamond q_2(S'_2) = \diamond q_2(S_1)$  (see Fig. 3.5). Moreover, since the terminal values  $\diamond q_2(S_1)$  and  $\diamond q_2(S_2)$ , lie within the bounded interval  $[-\diamond \hat{q}_2(0), -\diamond \hat{q}_2(0) + \frac{b_1}{b_2} r_k]$ , therefore the extension LFSP can be constructed such that it has a bounded cost, *i.e.*,

$$J_{S'_2}(\diamond f, \diamond g) - J_{S_2}(\diamond f, \diamond g) \leq \Delta , \quad (3.37)$$

and a bounded increase in weighted-sum-queue, *i.e.*,

$$|\langle b, \diamond q(S'_2) - \diamond q(S_2) \rangle| \leq \Delta , \quad (3.38)$$

for some large fixed  $\Delta < \infty$  which is independent of any components of the LFSP over  $[0, S]$ , (*e.g.* the value of  $S$  or the terminal values  $\diamond q_2(S_1)$  and  $\diamond q_2(S_2)$ .) Finally, the constant  $S$  can be chosen large enough such that, by (3.35) and (3.38), we have,

$$\langle b, \diamond q(S'_2) - \diamond q(S_1) \rangle > 0 ,$$

and by (3.34)–(3.38), for some small  $\epsilon_4 > 0$  that satisfies  $J_{***} + \epsilon_4 < J_*$ , we have,

$$\frac{J_{S'_2}(\diamond f, \diamond g) - J_{S_1}(\diamond f, \diamond g)}{\langle b, \diamond q(S'_2) \rangle - \langle b, \diamond q(S_1) \rangle} \leq J_{***} + \epsilon_4 < J_* . \quad (3.39)$$

Set  $c_0 = -\diamond\hat{q}_2(0)$  and for each  $m \in \{1, \dots, k\}$ , choose a

$$c_m \in \left( -\diamond\hat{q}_2(s) + \frac{b_1}{b_2}r_{m-1}, -\diamond\hat{q}_2(s) + \frac{b_1}{b_2}r_m \right)$$

such that the counting measure of set  $\{s \in [0, S'_2] : \diamond q(s) = c_m\}$  is finite; such  $\{c_m\}$  exist since  $\diamond q$  is Lipschitz. Lastly, choose a  $c_{k+1} < \infty$  large enough such that

$$\max_{s \in [S_1, S'_2]} \diamond q_2(s) < c_{k+1}.$$

For each  $m \in \{1, \dots, k+1\}$ , define a set

$$C_m = \{s \in [S_1, S'_2] : \diamond q_2(s) \in [c_{m-1}, c_m]\}.$$

We make the following three observations which will be used in the sequel: for each  $m \in \{1, \dots, k+1\}$ ,

- (i) the trajectory  $(\diamond q_2(s), s \in [S_1, S'_2])$  intersects with end points of interval  $[c_{m-1}, c_m]$  only finitely many times, therefore, the corresponding set  $C_m$  can be written as a union of finitely many intervals;
- (ii) if set  $C_m$  is non-empty, then

$$\diamond q_2(\min_{s \in C_m} s) = \diamond q_2(\max_{s \in C_m} s);$$

- (iii) the trajectory  $(\diamond q(s), s \in C_m)$  can intersect with at most one switching curve, namely  $x_2 = -\diamond\hat{q}_2(s) + \frac{b_1}{b_2}r_{m-1}$ , or equivalently, with at most two adjacent regions  $\diamond\mathcal{S}_{(\cdot)}$ , (*e.g.*, for  $m = k+1$  the above trajectory can intersect with adjacent regions  $\diamond\mathcal{S}_k$  and  $\diamond\mathcal{S}_0$ ).

Then using (i) and (ii) above, for all sets  $C_m$ , we must have,

$$\int_{C_m} \diamond q'_2(s) ds = 0. \tag{3.40}$$

Moreover, there exists a set  $C_m$  such that,

$$\int_{C_m} \langle b, \diamond q'(s) \rangle ds > 0 ,$$

and,

$$\frac{\int_{C_m} \left( L_{(f)}(\diamond f'(s)) + L_{(g)}(\diamond g'(s)) \right) ds}{\int_{C_m} \langle b, \diamond q'(s) \rangle ds} < J_* ,$$

otherwise (3.39) will not hold. With a set  $C_m$  for which the above two hold, let,

$$\left( \hat{\lambda}, \hat{\gamma}, \hat{\mu} \right) = \frac{1}{\nu(C_m)} \int_{C_m} (\diamond f'(s), \diamond g'(s), \mu(s)) ds .$$

By (iii) above, (3.18–3.20), and the fact that  $\diamond d(s) > 0$  in  $[S_1, S'_2]$ , the service vector  $\hat{\mu}$  is a convex combination of at most two adjacent vertices of capacity region  $V_{\hat{\gamma}}$  and lies on the *facet* with outer normal slope  $r_{m-1}$ . That is,  $\hat{\mu}$  is necessarily a maximal element of  $V_{\hat{\gamma}}$ . This, together with the fact that  $\hat{\lambda}_2 = \hat{\mu}_2$  which follows from (3.40), gives  $\hat{\mu} = \arg \min_{v \in V_{\hat{\gamma}}} \langle b, (\hat{\lambda} - v)^+ \rangle = v^*(\hat{\lambda}, \hat{\gamma})$ , and then,

$$\int_{C_m} \langle b, \diamond q'(s) \rangle ds = \langle b, \hat{\lambda} - v^*(\hat{\lambda}, \hat{\gamma}) \rangle \nu(C_m) .$$

Finally,

$$\begin{aligned} J_* &> \frac{\int_{C_m} \left( L_{(f)}(\diamond f'(s)) + L_{(g)}(\diamond g'(s)) \right) ds}{\int_{C_m} \langle b, \diamond q'(s) \rangle ds} , \\ &\geq \frac{L_{(f)}(\hat{\lambda}) + L_{(g)}(\hat{\gamma})}{\langle b, \hat{\lambda} - v^*(\hat{\lambda}, \hat{\gamma}) \rangle} . \end{aligned}$$

By the definition of  $J_*$ , the right side of last inequality cannot be less than  $J_*$ , giving the required contradiction; hence we conclude that  $J_* = J_{**}$ . ■

### 3.11 Conclusion and extensions

In order to minimize the asymptotic probability of weighted-sum-queue overflow, the desirable mode of overflow is one where queues may build up at different

rates, however, the total weighted service rate seen by the queues is the highest possible, (service rate subject to being not more than the arrival rate.) The p-Log scheduling rule minimizes the asymptotic probability of weighted-sum-queue overflow and exhibits such a mode of overflow. This property of p-Log rule is related to the collapse under fluid scaling of all but one set of the state space partition to either of the axes; the set that does not collapse is the horn-shaped weighted-max-sum-rate set. This collapse under fluid scaling is typical of the partition under any radial sum-rate monotone scheduler. However, the scaling or magnifying factor required to obtain a useful LFSP, and the shape of the sets of partition on the local fluid state space will vary for different radial sum-rate monotone schedulers. In this regard, p-Log rule yields an easily tractable partition of the local fluid space where all switching curves are parallel to one of the axes.

Recently in [47], the authors have reported a promising framework to relate the gradient field associated with a MaxWeight-type scheduler to the modes of overflow and large deviations of the appropriately scaled queue process. They are able to show that the Log rule indeed minimizes the asymptotic probability of sum-queue overflow; the framework, however, does not cover the p-Log or Exp rule since the vector field associated with either of these schedulers is not a gradient field. In this regard, Lemmas 3.2 and 3.3 of this thesis suggest that it may be possible in some cases to *locally* replace the vector field with a gradient field and thus obtain a suitable Lyapunov function, and relate the negative drift of this Lyapunov function to that of the quantity of interest (weighted sum-queue in our case).

The following extensions of the main result as well as the system model are possible without much effort.

First, the lower bound (*i.e.*, Theorem 3.1-i and its proof) goes through without any changes for any fixed number of queues (instead of only two) sharing the time-

varying server.

Second, the main result (Theorem 3.1) is also applicable to the following different and *simpler* system model. Instead of a single server with time-varying state

$$\mathbf{m}(t) \in \{1, 2, \dots, M\}$$

having distribution  $\pi$ , suppose there are  $M$  distinct servers with fixed but asymmetric capacities across the two queues; such servers are typically called parallel “unrelated” machines (see, *e.g.*, [4]). More specifically, the  $m^{\text{th}}$  server, if allocated to the  $i^{\text{th}} \in I$  queue, can serve  $\pi_i \mu_i^m \in \mathbb{Z}_+$  packets from the queue. The total service offered to a queue is taken to be the sum of service offered by each server assigned to that queue. Then the scheduling problem is to dynamically assign the servers to the queues based on the queue state. When queue is in state  $Q$ , the p-Log scheduler in this context allocates the  $m^{\text{th}}$  server to the queue  $i_{pLog}^*(Q, m)$  as given by (3.6). This system model is simpler in that the only random process now driving the system are the arrivals, but is also different from the original model in that there are multiple parallel servers with each server having asymmetric capacities across the two queues.

Third, the main result also goes through if the capacity regions  $V^m$  are permitted to be arbitrary convex polyhedra instead of just triangles obtained as a convex hull of service vectors  $(0, 0)$ ,  $(\mu_1^m, 0)$ , and  $(0, \mu_2^m)$ . That is, in a more general model, in any server state  $m$ , the server can be permitted to operate at any one of the  $k_m$  service vectors from the set,

$$\left\{ (\mu_1^m(1), \mu_2^m(1)), \dots, (\mu_1^m(k_m), \mu_2^m(k_m)) \right\} .$$

The region  $V^m$  then will be the convex hull of the  $k_m$  vertices associated with state  $m$ . The only change needed is to generalize the definition of p-Log rule as follows. When the system is in state  $(Q, m)$ , operate the server at a service vector  $\mu_{pLog}^*(Q, m) \in V^m$

given by,

$$\mu_{pLog}^*(Q, m) \in \arg \max_{y \in V^m} \langle y, h(Q) \rangle ,$$

where, in the case of a tie, if  $Q_1 \geq Q_2$ , then  $\mu_{pLog}^*(Q, m)$  maps to the maximizer with the largest capacity for Queue 1, otherwise  $\mu_{pLog}^*(Q, m)$  maps to the maximizer with the largest capacity for Queue 2. This generalization affects the fluid and local fluid sample paths through (3.20) in Lemma 3.1, which can be shown to hold exactly as in [48, 49].

Besides the above extensions, we will conclude by stating one more interesting application of the p-Log scheduler. A throughput-optimal scheduler can also be used to offer minimum and maximum average service rate guarantees to *infinitely backlogged* queues, referred to as tasks, sharing a time-varying server/wireless channel [39, 50]. This is done by using virtual token queues that are fed by deterministic arrivals at a constant rate  $\lambda_i$ , and making scheduling decisions to serve tasks based on the virtual token queues (augmented with a scheduling rule to use when all token queues are empty). If rates  $\lambda_i$  are feasible (*i.e.*, vector  $\lambda$  lies in the interior of capacity region  $V_\pi$  associated with the time-varying server), then under any throughput-optimal scheduler, each task  $i$  will be offered an average service rate  $v_i \geq \lambda_i$  (such that  $v \in V_\pi$ ). However, if rates  $\lambda_i$  are not feasible, then main result of this chapter implies that the average service rate vector  $v$  has the following interesting and desirable property under p-Log rule:  $\langle b, v \rangle$  is maximized subject to  $v_i \leq \lambda_i$ . That is, p-Log rule splits the tasks in two sets, for one set of tasks  $v_i = \lambda_i$ , whereas for the other  $v_i < \lambda_i$ , and the sets are chosen such that the total weighted service rate  $\langle b, v \rangle$  is maximized.

## Appendix – Proof of Theorem 3.2

We begin by developing the necessary results need for the proof of Theorem 3.2. The corresponding proof in [19] has some parts– lemmas and theorems– that are not specific to the Exp rule; these mostly go through by interpreting  $q_*(t)$  (which is the notation for max-queue in [19]) as weighted-sum-queue  $\sum_{i \in I} b_i q_i(t)$ , while other lemmas and theorems require more work specific to the p-Log rule. We proceed by stating the following two theorems whose proofs are short and identical to those of Theorem 8.5 and Theorem 8.6 in [19] by interpreting  $q_*(t)$  as  $\sum_{i \in I} b_i q_i(t)$ .

**Theorem 3.3** (See Theorem 8.5 of [19]) *For any fixed  $T \geq 0$  and  $0 \leq c < 1$ , let us denote*

$$J_{**, \leq T, c} = \inf_{\psi} \left\{ \bar{J}_t | \psi : \sum_{i \in I} b_i q_i(0) \leq c \text{ and } \sum_{i \in I} b_i q_i(t) \geq 1 \text{ for some } t \leq T \right\}.$$

Then, we have,

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \sup_{\sum_{i \in I} b_i q_i^{(n)}(0) \leq c} \mathbb{P} \left( \sup_{t \in [0, T]} \sum_{i \in I} b_i \mathbf{q}_i^{(n)}(t) > 1 \right) \leq -J_{**, \leq T, c},$$

and as  $c \rightarrow 0$ ,

$$J_{**, \leq T, c} \nearrow J_{**, \leq T, 0} = \inf_{t \leq T} J_{**, t}. \quad (3.41)$$

**Theorem 3.4** (See Theorem 8.6 of [19]) *For any fixed  $C^* > \delta > 0$ , and  $T > 0$ , let us denote*

$$K(C^*, \delta, T) = \inf_{\psi} \left\{ \bar{J}_T | \psi : \sum_{i \in I} b_i q_i(0) \leq C^* \text{ and } \sum_{i \in I} b_i q_i(t) \geq \delta \text{ for all } t \in [0, T] \right\}.$$

Then, we have,

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \sup_{\sum_{i \in I} b_i q_i^{(n)}(0) \leq C^*} \mathbb{P} \left( \inf_{t \in [0, T]} \sum_{i \in I} b_i \mathbf{q}_i^{(n)}(t) \geq \delta \right) \leq -K(C^*, \delta, T).$$

**Theorem 3.5** (See Theorem 8.7 of [19]) *For any  $C^* > 0$ , there exists  $\Delta_1 > 0$  such that for all sufficiently large  $T$  and all  $\delta \in (0, C^*)$ , we have  $K(C^*, \delta, T) \geq \Delta_1 T$ .*



**Proof** It is clear that  $K(C^*, \delta, T)$  is increasing in  $\delta$ . We will show that there exists an  $\epsilon_5 > 0$  and  $\delta_5 > 0$  such that for any GFSP satisfying  $\langle b, q(\cdot) \rangle > 0$  over  $[0, T]$ ,

$$\text{if } \bar{J}_T - \bar{J}_0 \leq \epsilon_5 T, \quad \text{then } \langle b, q(T) - q(0) \rangle \leq -\delta_5 T, \quad (3.42)$$

hence for all  $T > C^*/\delta_5$  and all  $\delta \in (0, C^*)$ , we must have  $K(C^*, \delta, T) > \epsilon_5 T$ , thus the desired result.

Fix  $S$  large enough as required by Lemma 2.2 and recall  $\epsilon_2$  and  $\delta_2$  therein. For each  $n$ , define sequence  $\{\tau_l^{(n)}, l = 0, 1, \dots, l^*\}$  in interval  $[0, T]$  such that  $\tau_0^{(n)} = 0$ , and,

$$\tau_{l+1}^{(n)} = \tau_l^{(n)} + \sigma_n(\tau_l^{(n)})S,$$

where  $\sigma_n(\cdot) = \sqrt{q_*^{(n)}(\cdot)}/\sqrt{n}$ , and  $l^*$  is the largest integer (that depends on  $n$ ) such that  $\tau_{l^*}^{(n)} \leq T$ . Let  $B_2^{(n)}$  be the union of intervals  $[\tau_l^{(n)}, \tau_{l+1}^{(n)}]$  over which the refined cost is strictly less than  $\epsilon_2 \sigma_n(\tau_l^{(n)})S$ , *i.e.*,

$$B_2^{(n)} = \left\{ \cup [\tau_l^{(n)}, \tau_{l+1}^{(n)}] : \bar{J}_{\tau_{l+1}^{(n)}}^{(n)} - \bar{J}_{\tau_l^{(n)}}^{(n)} < \epsilon_2 \sigma_n(\tau_l^{(n)})S \right\},$$

and  $B_1^{(n)} = [0, T] \setminus B_2^{(n)}$ . Now pick a positive  $\epsilon_5 < \epsilon_2/2$ . From this point on, we assume  $n$  large enough. The Lebesgue measures of  $B_1^{(n)}$  and  $B_2^{(n)}$  satisfy  $\nu(B_1^{(n)}) \leq \frac{2\epsilon_5}{\epsilon_2}T$  and  $\nu(B_2^{(n)}) \geq (1 - \frac{2\epsilon_5}{\epsilon_2})T$ . Moreover, over any interval  $[\tau_l^{(n)}, \tau_{l+1}^{(n)}] \subseteq B_2^{(n)}$ , we must have,

$$\left\langle b, q^{(n)}(\tau_{l+1}^{(n)}) - q^{(n)}(\tau_l^{(n)}) \right\rangle \leq -\frac{\delta_2}{2} \sigma_n(\tau_l^{(n)})S.$$

otherwise we could construct a LFSP contradicting Lemma 2.2 by choosing a subsequence in  $n$  along which the above inequality does not hold and  $\tau_l^{(n)}$  converges to some  $\tau \in [0, T]$ . Then we have the following bound on the total increment of  $\langle b, q^{(n)}(\cdot) \rangle$  over  $B_2^{(n)}$ ,

$$\sum_{[\tau_l^{(n)}, \tau_{l+1}^{(n)}] \subseteq B_2^{(n)}} \left\langle b, q^{(n)}(\tau_{l+1}^{(n)}) - q^{(n)}(\tau_l^{(n)}) \right\rangle \leq -\frac{\delta_2}{2} \left(1 - \frac{2\epsilon_5}{\epsilon_2}\right)T.$$

Also, because we have assumed bounded arrivals, there is a finite  $c_1 > 0$  such that the total increment of  $\langle b, q^{(n)}(\cdot) \rangle$  over  $[0, T] \setminus B_2^{(n)}$  satisfies,

$$\left\langle b, q^{(n)}(T) - q^{(n)}(\tau_{l^*}) \right\rangle + \sum_{[\tau_l^{(n)}, \tau_{l+1}^{(n)}] \subseteq B_1^{(n)}} \left\langle b, q^{(n)}(\tau_{l+1}^{(n)}) - q^{(n)}(\tau_l^{(n)}) \right\rangle \leq c_1 \frac{2\epsilon_5}{\epsilon_2} T .$$

Finally, summing the above two inequalities and taking limit, we get,

$$\begin{aligned} \langle b, q(T) - q(0) \rangle &\leq \limsup_{n \rightarrow \infty} \langle b, q^{(n)}(T) - q^{(n)}(0) \rangle , \\ &\leq -T \left( \frac{\delta_2}{2} - \frac{\epsilon_5}{\epsilon_2} (\delta_2 + 2c_1) \right) . \end{aligned}$$

Now we can fix a positive  $\delta_5 < \delta_2/2$  and choose  $\epsilon_5 > 0$  small enough to satisfy (3.42), completing the proof.  $\blacksquare$

**Remark 3.5** *While the reader may conclude the throughput-optimality of the p-Log rule from Lemma 3.2 alone, nevertheless, the throughput-optimality explicitly bears out as a corollary of the above theorem. Specifically, (3.42) shows that for any zero cost limiting trajectories  $(f, g, q)$ —i.e.  $(\mathbf{f}^{(n)}, \mathbf{g}^{(n)}, \mathbf{q}^{(n)})$  converge u.o.c to  $(f, g, q)$  with probability 1— with initial condition  $\langle b, q(0) \rangle = 1$ , we have  $\langle b, q(T) \rangle = 0$  for all  $T > \frac{1}{\delta_5}$ . That is, for any deterministic initial state satisfying  $\langle b, q^{(n)}(0) \rangle \leq 1$ , we have  $\langle b, \mathbf{q}^{(n)}(T) \rangle \rightarrow 0$  with probability 1. Convergence with probability 1, along with the bound  $\langle b, \mathbf{q}^{(n)}(T) - q^{(n)}(0) \rangle < C(b_1 + b_2)(T + 1)$  (recall the assumption of bounded arrivals), implies convergence in the mean too, and therefore,*

$$\limsup_{n \rightarrow \infty} \sup_{\sum_{i \in I} b_i q_i^{(n)}(0) \leq 1} \mathbb{E} \left[ \sum_{i \in I} b_i \mathbf{q}_i^{(n)}(T) \right] = 0. \quad (3.43)$$

By [51] (or see Theorem 4.1 of [11],) this proves the throughput-optimality of p-Log rule.

Fix a time  $T > 0$  and constants  $C^* > \delta > 0$ . Just as (3.43) was obtained from (3.42), the following too can be shown. For all large  $n$ , uniformly on  $\langle b, q^{(n)}(0) \rangle \leq C^*$ ,

if  $\langle b, \mathbf{q}^{(n)}(\cdot) \rangle > \delta$  over  $[t, t + T]$ , then,

$$\sup_{\langle b, \mathbf{q}^{(n)}(0) \rangle \leq C^*} \mathbb{E} [\langle b, \mathbf{q}^{(n)}(t + T) \rangle - \langle b, \mathbf{q}^{(n)}(t) \rangle] \leq -\frac{\delta_5}{2} T .$$

The above along with Dynkin's formula (see Theorem 19.1.2 of [69]) implies the following result.

**Lemma 3.5** *Let constants  $C^* > \delta > 0$  be fixed. Consider the stopping time,*

$$\beta^{(n)} = \inf \{ t \geq 0 : \langle b, \mathbf{q}^{(n)}(t) \rangle \leq \delta \} .$$

*Then, for all sufficiently large  $n$ , uniformly on the initial states with  $\langle b, \mathbf{q}^{(n)}(0) \rangle \leq C^*$ , we have,*

$$\mathbb{E}[\beta^{(n)}] \leq \Delta_2 C^* ,$$

*for some finite  $\Delta_2 > 0$ .*

**Proof of Theorem 3.2** Now we can proceed with the proof of Theorem 3.2. Consider the scaled (random) process  $\mathbf{q}^{(n)}$ . For any fixed constants

$$C^* > 1 > \epsilon^* > \epsilon > \delta > 0,$$

define the following stopping times,

$$\begin{aligned} \alpha^{(n)} &= \inf \{ t > 0 : \langle b, \mathbf{q}^{(n)}(t) \rangle \geq 1 \} , \\ \beta^{(n)} &= \inf \{ t > 0 : \langle b, \mathbf{q}^{(n)}(t) \rangle \leq \delta \} , \\ \eta^{(n)} &= \inf \{ t > \beta^{(n)} : \langle b, \mathbf{q}^{(n)}(t) \rangle \geq \epsilon \} . \end{aligned}$$

Let  $p^{(n)}$  denote the stationary distribution of process  $\mathbf{q}^{(n)}(t)$ ,  $p_x^{(n)}$  its distribution conditional on  $\mathbf{q}^{(n)}(0) = x$ , and  $\mathbb{E}_x$  expectation under  $p_x^{(n)}$ . Then, for any arbitrary

$T > 0$ , it is easy to show (see (8.15) [19]) the following upper bound on the probability of overflow,

$$p^{(n)}\left(\sum_{i \in I} b_i \mathbf{q}_i^{(n)} > 1\right) \leq \frac{\sup_{y: \langle b, y \rangle \leq C^*} \mathbb{E}_y[\boldsymbol{\beta}^{(n)}]}{\inf_{z: \langle b, z \rangle \geq \epsilon} \mathbb{E}_z[\boldsymbol{\eta}^{(n)}]} \sup_{x: \langle b, x \rangle \leq \epsilon^*} \left(p_x^{(n)}(\boldsymbol{\beta}^{(n)} \geq T) + p_x^{(n)}(\boldsymbol{\alpha}^{(n)} \leq T)\right). \quad (3.44)$$

Now, by uniform upper bound in Lemma 3.5, we have

$$\limsup_{n \rightarrow \infty} \sup_{y: \langle b, y \rangle \leq C^*} \mathbb{E}_y[\boldsymbol{\beta}^{(n)}] \leq \Delta_2 C^*,$$

and by bounded arrivals, we have the lower bound

$$\liminf_{n \rightarrow \infty} \inf_{z: \langle b, z \rangle \geq \epsilon} \mathbb{E}_z[\boldsymbol{\eta}^{(n)}] > 0.$$

Therefore, the terms that are going to decide the limit,

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log p^{(n)}\left(\sum_{i \in I} b_i \mathbf{q}_i^{(n)} > 1\right),$$

in (3.44) are  $p_x^{(n)}(\boldsymbol{\beta}^{(n)} \geq T)$  and  $p_x^{(n)}(\boldsymbol{\alpha}^{(n)} \leq T)$ .

By Theorem 3.5, we can choose  $T$  large enough such that

$$K(\epsilon^*, \delta, T) \geq K(C^*, \delta, T) \geq J_{**}.$$

Then by Theorem 3.4, we have

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \sup_{x: \langle b, x \rangle \leq \epsilon^*} p_x^{(n)}(\boldsymbol{\beta}^{(n)} \geq T) \leq -K(C^*, \delta, T) \leq -J_{**}.$$

By (3.41) and the definition of  $J_{**}$  in (3.15), for any  $\epsilon_6 > 0$ , we can choose an even larger a  $T$  (if required) and an  $\epsilon^* > 0$  small enough such that  $J_{**, \leq T, \epsilon^*} > J_{**} - \epsilon_6$ .

Then by Theorem 3.3,

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \sup_{x: \langle b, x \rangle \leq \epsilon^*} p_x^{(n)}(\boldsymbol{\alpha}^{(n)} \leq T) \leq -J_{**, \leq T, \epsilon^*} < -(J_{**} - \epsilon_6).$$

Since we can choose  $\epsilon_6$  arbitrarily small (and subsequently fix constants  $1 > \epsilon^* > \epsilon > \delta > 0$ ), substituting the above two bounds in (3.44) completes the proof.  $\blacksquare$

## Part II

# Best-effort scheduling

## Chapter 4

# Throughput optimality of *delay*-driven MaxWeight scheduler for wireless systems with *flow dynamics*

### 4.1 Overview and main contributions

In the second half of this thesis, we consider a wireless link shared by a *dynamic* population of *best-effort* flows. Flows of random size (bits) arrive at the base station at random times, and leave when they have been completely transmitted. The transmission rate supported by the wireless channel of each flow while the flow awaits transmission varies randomly over time and is independent of that of the other flows. The scheduling problem in this context is to select which flow to serve based on the current system state (*e.g.*, residual flow sizes, sojourn times, and channel states of the contending flows), with the objective of minimizing the sojourn time/file transfer delay. It has recently been shown that for such a system, the well-known (backlog-driven) MaxWeight scheduler is *not* throughput optimal. That is to say, the MaxWeight scheduler will not stabilize a given system even though it is possible to construct a stabilizing scheduler using the various flow- and channel-related statistics. However, in this chapter, we show that the *delay*-driven MaxWeight scheduler is, nevertheless, throughput optimal for such a system. The delay-driven MaxWeight, like its backlog-driven version, does not require any knowledge of the flow- or channel-related statistics.

## Contributions

For a system with flow-level dynamics, [28] recently showed that the queue-driven MaxWeight scheduler is *not* throughput optimal.<sup>1</sup> In this chapter, we show that the *delay*-driven MaxWeight still is throughput optimal in the dynamic flow setting. The critical observation which explains why the queue-driven version is not throughput optimal but the delay-driven version is, is as follows. In the setting where there is a fixed number of flows, a *linear* relation can be established between the head-of-line packet delay and the queue length of a flow (Little’s law) as either one gets large. By contrast, in the setting with dynamic number of flows, while the head-of-line delay of an un-served flow will continue to increase, its queue length will not due to the finite size of flows. So, in the former setting, the queue-driven and the delay-driven versions of MaxWeight are equivalent in some sense, whereas, in the latter, the queue-driven version may perpetually fail to exploit good channel states of small queues (*i.e.* files with few residual bits) *irrespective of how long these small files wait* while the bigger newer files may get scheduled (because of their longer queue lengths) even when their channels are poor; see excellent illustrative examples in [28].

## 4.2 System model

Let random  $\mathbf{A}(t) \in \mathbb{Z}_+$  denote the number of files arriving in time slot  $[t, t+1)$ , these files will not be available for service until the next time slot. We assume  $\mathbf{A}(\cdot)$  are i.i.d. and bounded, with mean  $\lambda \equiv \mathbb{E}\mathbf{A}(0)$ . For  $0 < i \leq \mathbf{A}(t)$ , let  $\mathbf{B}_i(t)$  denote the file size in bits of the  $i^{\text{th}}$  arriving file. We assume that  $\mathbf{B}_i(t)$  are i.i.d. (across both  $t$  and  $i$ ), are bounded, and have mean  $\beta \equiv \mathbb{E}\mathbf{B}_1(0)$ . We will use bold face, *e.g.*  $(\mathbf{A}(t), t = 0, 1, \dots)$ , to denote the random process and plain font, *e.g.*  $(A(t), t = 0, 1, \dots)$ , to denote a realization of the process. Also, we will make

---

<sup>1</sup>Queue-driven Exponential rule and Log rule can similarly be shown to be not throughput optimal in the dynamic flow setting.

a distinction between “increasing” and “strictly increasing” and between “positive” and “strictly positive” etc.

Let  $\mathbf{Q}(t) \in \mathbb{Z}_+$  denote the number of files present in the system at the end of time slot  $[t - 1, t)$ . Then in the time slot  $[t, t + 1)$ , at most one of the  $\mathbf{Q}(t)$  files present in the system can be scheduled to receive service. For each file  $0 < i \leq \mathbf{Q}(t)$  present in the system at the end of time slot  $[t - 1, t)$ , let

- $[\mathbf{T}_i(t), \mathbf{T}_i(t) + 1)$  denote the time slot in which the  $i^{\text{th}}$  file arrived. We index the files in order of their arrival time, *i.e.*,  $\mathbf{T}_1(t) \leq \mathbf{T}_2(t) \leq \dots \leq \mathbf{T}_{\mathbf{Q}(t)}(t)$ . Let

$$\mathbf{T}(t) \equiv (\mathbf{T}_i(t), 1 \leq i \leq \mathbf{Q}(t)) \quad \text{and} \quad \mathbf{W}(t) \equiv (\mathbf{W}_i(t), 1 \leq i \leq \mathbf{Q}(t)),$$

where  $\mathbf{W}_i(t) \equiv t - \mathbf{T}_i(t) > 0$  denotes the current sojourn time of the  $i^{\text{th}}$  file. Also, let  $\mathbf{W}(t) \equiv 0$  if  $\mathbf{Q}(t) = 0$ .

- random  $\mathbf{R}_i(t)$  denote the maximum number of bits that can be served/transmitted from file  $i$ , if it is scheduled in time slot  $[t, t + 1)$ . We will sometimes refer to  $\mathbf{R}_i(t)$  as the state of the  $i^{\text{th}}$  files channel. We assume that  $\mathbf{R}_i(t)$  lies in a finite set  $\{0, 1, \dots, r^{\max}\}$  and is i.i.d. (across both  $t$  and  $i$ ). Also, let  $p_0 \equiv \mathbb{P}(\mathbf{R}_1(0) > 0)$  and  $p \equiv \mathbb{P}(\mathbf{R}_1(0) = r^{\max}) > 0$ .
- $\mathbf{L}_i(t) > 0$  denote the number of bits left in the  $i^{\text{th}}$  file at the end of slot  $[t - 1, t)$ , or, equivalently, the number of bits available for transmission in time slot  $[t, t + 1)$ . A file leaves the system once all its bits have been served/transmitted. Also let  $\mathbf{L}(t) \equiv (\mathbf{L}_i(t), 1 \leq i \leq \mathbf{Q}(t))$ .

See Fig. 4.1 for a graphical illustration of the system.

Let  $\gamma \equiv \mathbb{E} \left[ \frac{\mathbf{B}_1(0)}{r^{\max}} \right]$  and  $\rho \equiv \lambda\gamma$ , *i.e.*, the average amount of work (in number of slots) per slot that is entering the system if the files could always be served at



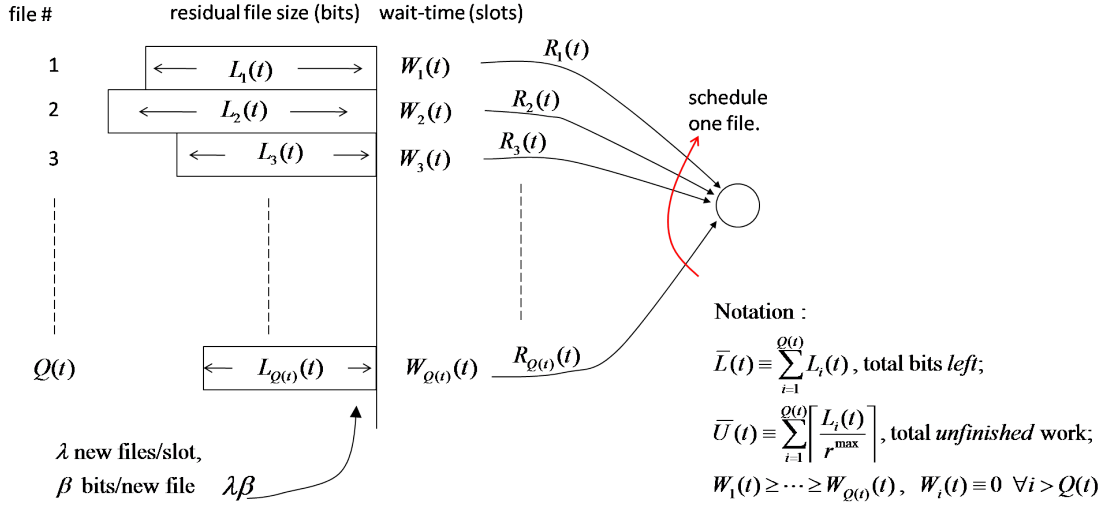


Figure 4.1: System model.

$r^{\max}$  bits/slot. We assume that  $\epsilon_0 \equiv 1 - \rho > 0$  which is a sufficient condition for stabilizability [28].

**Remark 4.1** *We have assumed that file size and channel state are i.i.d. across files, however, consider the following generalization. Suppose there are a fixed number  $K$  of classes of files, where the class determines the distribution of the arrival and the file size processes, as well as the distribution of the channel seen by the file. Then for each class  $k \in \{1, 2, \dots, K\}$ , suppose a file in the  $k^{\text{th}}$  class sees an i.i.d. channel that has the same distribution as  $\mathbf{R}_1^k(0)$ . So the distribution of channel state may differ across classes (as indicated by the superscript  $k$ ), however, we restrict that for all  $k$ ,  $\mathbb{P}(\mathbf{R}_1^k(0) = r^{\max}) > 0$  and  $\mathbb{P}(\mathbf{R}_1^k(0) > r^{\max}) = 0$ . That is, the highest possible rate supported by the channel of any file of any class is  $r^{\max}$ . Using this  $r^{\max}$ , let  $\lambda^k, \beta^k, \gamma^k$  and  $\rho_k$  respectively denote the mean arrival rate, file size, work load per file, and total work load associated with the  $k^{\text{th}}$  class. Then, the results presented in this chapter are applicable, without modification, to this more general multi-class system model,*

by appropriately defining  $\lambda, \beta, \gamma, \rho$  and  $p$  as follows:

$$\lambda = \sum_{k=1}^K \lambda_k, \quad \beta = \sum_{k=1}^K \frac{\lambda_k}{\lambda} \beta_k, \quad \gamma = \sum_{k=1}^K \frac{\lambda_k}{\lambda} \gamma_k,$$

$$\rho = \sum_{k=1}^K \rho_k, \quad p = \min_{1 \leq k \leq K} \mathbb{P}(\mathbf{R}_1^k(0) = r^{\max}).$$

#### 4.2.1 Delay-driven MaxWeight scheduler

**Definition 4.1** For any time slot  $[t, t+1)$ , when there are  $Q(t) > 0$  files present in the system (at the end of the last slot), and the corresponding sojourn time and channel state vectors are  $W(t) = (W_i(t), 1 \leq i \leq Q(t))$  and  $R(t) = (R_i(t), 0 \leq i \leq Q(t))$  respectively, then schedule for service a file  $i^*(W(t), R(t))$  that satisfies,

$$i^*(W(t), R(t)) \in \arg \max_{1 \leq i \leq Q(t)} W_i(t) R_i(t), \quad (4.1)$$

with ties broken in favor of the smallest index  $i$  achieving the max in above.

**Remark 4.2** Under MaxWeight, the process  $\mathbf{Q}(\cdot)$  evolves as follows,

$$\mathbf{Q}(t+1) = (\mathbf{Q}(t) - \mathbf{1}_{\{\mathbf{L}_*(t) \leq \mathbf{R}_*(t)\}})^+ + \mathbf{A}(t).$$

where  $\mathbf{L}_*(\cdot) \equiv \mathbf{L}_{i^*(\mathbf{W}(\cdot), \mathbf{R}(\cdot))}(\cdot)$  and  $\mathbf{R}_*(\cdot) \equiv \mathbf{R}_{i^*(\mathbf{W}(\cdot), \mathbf{R}(\cdot))}(\cdot)$ . For later use, let us also define  $\mathbf{T}_*(\cdot) \equiv \mathbf{T}_{i^*(\mathbf{W}(\cdot), \mathbf{R}(\cdot))}(\cdot)$  and  $\mathbf{W}_*(\cdot) \equiv \mathbf{W}_{i^*(\mathbf{W}(\cdot), \mathbf{R}(\cdot))}(\cdot)$ .

**Remark 4.3** At any  $t \in \mathbb{Z}_+$ , the state,  $\mathbf{S}(t)$ , of the system given by,

$$\mathbf{S}(t) \equiv \left( \mathbf{Q}(t); \mathbf{W}(t) = (\mathbf{W}_i(t), 1 \leq i \leq \mathbf{Q}(t)); \right. \\ \left. \mathbf{L}(t) = (\mathbf{L}_i(t), 1 \leq i \leq \mathbf{Q}(t)) \right),$$

forms a discrete time homogeneous Markov chain under the MaxWeight scheduler.

### 4.3 Main result

The main result of this chapter is given below and in the rest of the chapter we will give the proof of this result.

**Theorem 4.1** *Delay-driven MaxWeight scheduler is throughput optimal, i.e., for any  $\rho < 1$  in the system described in Section 4.2, the delay-driven MaxWeight scheduler stabilizes the system.*

We will need the following quantities all of which are derived from a system sample-path<sup>2</sup>  $\mathcal{X} \equiv (S(t), A(t), R(t), t \in \mathbb{Z})$ .

- Let  $U(t) \equiv (U_i(t), 1 \leq i \leq Q(t))$ , where  $U_i(t) \equiv \left\lceil \frac{L_i(t)}{r^{\max}} \right\rceil$ , i.e., the number of slots it will take to serve the  $i^{\text{th}}$  present file if served at rate  $r^{\max}$  bits/slots.
- Let  $\bar{U}(t) \equiv \sum_{i=1}^{Q(t)} U_i(t)$ , i.e., the total unfinished work present in the system (assuming the service rate is always  $r^{\max}$ ) at the end of slot  $[t-1, t)$ .
- Let  $\bar{L}(t) \equiv \sum_{i=1}^{Q(t)} L_i(t)$ , i.e., the total number of bits present at the end of slot  $[t-1, t)$ .

In the sequel, we will extend the domain of all discrete time processes and functions to continuous time: a function originally defined on integer times has the same value at any real  $t$  that it takes at  $\lceil t \rceil$ . Also, we will extend the definition of  $W_i(t)$  to all  $i \in \{1, 2, \dots\}$  by letting  $W_i(t) = 0$  for  $i \in \{Q(t) + 1, Q(t) + 2, \dots\}$ .

For each  $n \in \mathbb{Z}_+$ , consider a independent and stochastically equivalent system

$$\mathcal{X}^{(n)} \equiv (S^{(n)}(0); \mathbf{S}^{(n)}(t+1), \mathbf{A}^{(n)}(t), \mathbf{R}^{(n)}(t), t \geq 0)$$

under the delay-driven MaxWeight scheduler and where the initial state  $S^{(n)}(0)$  is non-random and satisfies,

$$\|S^{(n)}(0)\| \equiv \bar{U}^{(n)}(0) + W_1^{(n)}(0) = n, \quad (4.2)$$

---

<sup>2</sup>Even though  $(A(t), t \geq 0)$  can be inferred from the other components of  $\mathcal{X}$ , for clarity we explicitly include  $(A(t), t \geq 0)$  in definition of  $\mathcal{X}$ .

*i.e.*, the total work in terms of number of slots it will take to serve the files present at  $t = 0$  if served at rate  $r^{\max}$  per slot, plus the current sojourn time of the *oldest* file in the system.

The following proposition is due to [51], and will be used in proving Theorem 4.1.

**Proposition 4.1** *Suppose there exists an  $\epsilon > 0$  and an integer  $t_1 > 0$  such that the following holds for any sequence of systems  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$  satisfying (4.2),*

$$\limsup_{n \rightarrow \infty} \mathbb{E} \left( \frac{1}{n} \|\mathbf{S}^{(n)}(nt_1)\| \right) \leq 1 - \epsilon, \quad (4.3)$$

*then the Markov chain  $(\mathbf{S}(t), t \in \mathbb{Z})$  is stable.*

See [10] for a use of this proposition to show throughput optimality of queue- and delay-driven MaxWeight in the case of a system with a *fixed* number of flows each having a corresponding stationary exogenous packet arrival process.

We will now define the fluid-scaled functions and processes obtained from the sequence  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$ ; these fluid-scaled functions and their limits will be used in showing the conditions required for Proposition 4.1, *i.e.*, the existence of  $\epsilon$  and  $t_1$  as in (4.3).

#### 4.3.0.1 Fluid limit of the *deterministic* initial state

For  $t \in (-\infty, 0]$ , let  $F^{(n)}(t)$  denote the number of files present at time 0 that had arrived by the end of time slot  $[[t] - 1, [t]]$ , *i.e.*,

$$F^{(n)}(t) \equiv \sum_{i=1}^{Q^{(n)}(0)} \mathbf{1}_{\{T_i^{(n)}(0) \leq t-1\}}.$$

Since  $W_1^{(n)}(0) \leq n$ , we have that  $F^{(n)}(-n) = 0$  and  $F^{(n)}(0) = Q^{(n)}(t)$ .

Let,

$$\bar{l}^{(n)}(0) \equiv \frac{1}{n} \bar{L}^{(n)}(0), \quad \bar{u}^{(n)}(0) \equiv \frac{1}{n} \bar{U}^{(n)}(0), \quad q^{(n)}(0) \equiv \frac{1}{n} Q^{(n)}(0),$$

and for  $t \in [-1, 0]$ , let,

$$f^{(n)}(t) \equiv \frac{1}{n} F^{(n)}(nt),$$

and for  $x \in [0, \infty)$ , let<sup>3</sup>,

$$w_x^{(n)}(0) \equiv \frac{1}{n} W_{[nx]+1}^{(n)}(0),$$

where  $w_x^{(n)}(0)$  is decreasing in  $x$ . Since,

$$0 \leq q^{(n)}(0) \leq \bar{l}^{(n)}(0) \leq r^{\max} \left( \bar{u}^{(n)}(0) + w_0^{(n)}(0) \right) = r^{\max}, \quad (4.4)$$

therefore, along some subsequence of  $n$  (for simplicity still denoted by  $n$ ), we have,

$$\begin{aligned} \bar{l}^{(n)}(0) &\rightarrow \bar{l}(0), & \bar{u}^{(n)}(0) &\rightarrow \bar{u}(0), \\ w_0^{(n)}(0) &\rightarrow w_0(0), & q^{(n)}(0) &\rightarrow q(0), \end{aligned}$$

where  $\bar{u}(0) + w_0(0) = 1$ . With  $w_0(0)$  and  $f(0) \equiv q(0)$  as above, let

$$(w_x(0), 0 \leq x \leq q(0)) \quad \text{and} \quad (f(t), -1 \leq t \leq 0)$$

be weak limits of  $(w_x^{(n)}(0), 0 \leq x \leq q(0))$  and  $(f^{(n)}(t), -1 \leq x \leq 0)$  respectively along a further subsequence of  $n$ , *i.e.*,

$$w_x^{(n)}(0) \rightarrow w_x(0), \quad f^{(n)}(t) \rightarrow f(t)$$

at the points of continuity of  $w_{(\cdot)}(0)$  and  $f(\cdot)$  respectively. To summarize, the (partial) initial state of the limiting system is captured by,

$$(q(0); w(0) \equiv (w_x(0), 0 \leq x \leq q(0)); (f(t), -1 \leq t \leq 0); \bar{u}(0) + w_0(0) = 1) .$$

---

<sup>3</sup> $(f^{(n)}(t), -1 \leq t \leq 0)$  and  $(w_x^{(n)}(0), 0 \leq x \leq q^{(n)}(0))$  are related. That is,  $w_x^{(n)}(0) = 1/n - \inf_t (t : f^{(n)}(t) \geq x + 1/n)$ .

### 4.3.0.2 Deterministic fluid limit of the *random* state for $t \geq 0$

We will also need the following fluid-scaled processes, all defined for  $t \in \mathbb{R}_+$ .

Let  $\mathbf{F}^{(n)}(t)$ ,  $\mathbf{F}_l^{(n)}(t)$ , and  $\mathbf{F}_u^{(n)}(t)$  respectively be the total files, the total bits, and the total work arriving up to the end of time slot  $[[t] - 1, [t]]$ , *i.e.*,

$$\begin{aligned}\mathbf{F}^{(n)}(t) &\equiv Q^{(n)}(0) + \sum_{k=0}^{t-1} \mathbf{A}^{(n)}(k) , \\ \mathbf{F}_l^{(n)}(t) &\equiv \bar{L}^{(n)}(0) + \sum_{k=0}^{t-1} \sum_{m=1}^{\mathbf{A}^{(n)}(k)} \mathbf{B}_m^{(n)}(k) , \\ \mathbf{F}_u^{(n)}(t) &\equiv \bar{U}^{(n)}(0) + \sum_{k=0}^{t-1} \sum_{m=1}^{\mathbf{A}^{(n)}(k)} \left\lceil \frac{\mathbf{B}_m^{(n)}(k)}{r^{\max}} \right\rceil .\end{aligned}$$

Then, along a further subsequence of  $n$ , we have the following uniform over compact sets (u.o.c) convergences (see Theorem 4.1 of [52], Lemma 1 of [10]),

$$\begin{aligned}\left( \mathbf{f}^{(n)}(t) \equiv \frac{1}{n} \mathbf{F}^{(n)}(nt), t \geq 0 \right) &\rightarrow \left( f(t) \equiv q(0) + \lambda t, t \geq 0 \right) , \\ \left( \mathbf{f}_l^{(n)}(t) \equiv \frac{1}{n} \mathbf{F}_l^{(n)}(nt), t \geq 0 \right) &\rightarrow \left( f_l(t) \equiv \bar{l}(0) + \lambda \beta t, t \geq 0 \right) , \\ \left( \mathbf{f}_u^{(n)}(t) \equiv \frac{1}{n} \mathbf{F}_u^{(n)}(nt), t \geq 0 \right) &\rightarrow \left( f_u(t) \equiv \bar{u}(0) + \rho t, t \geq 0 \right) .\end{aligned}\tag{4.5}$$

Let  $\mathbf{D}^{(n)}(\tau, t)$ ,  $\mathbf{D}_l^{(n)}(\tau, t)$ , and  $\mathbf{D}_u^{(n)}(\tau, t)$  respectively be the total files, the total bits, and the total work which

- (i) arrived before the end of time slot  $[[\tau] - 1, [\tau]]$ , *and*
- (ii) departed/completed service by the end of time slot  $[[t] - 1, [t]]$ .

These can be mathematically defined as follows: for  $t \geq 0$ ,  $t \geq \tau > -\infty$ ,

$$\begin{aligned}\mathbf{D}^{(n)}(\tau, t) &\equiv \sum_{k=0}^{t-1} \mathbf{1}_{\{\mathbf{L}_*^{(n)}(k) \leq \mathbf{R}_*^{(n)}(k), \mathbf{T}_*^{(n)}(k) \leq \tau-1\}} , \\ \mathbf{D}_l^{(n)}(\tau, t) &\equiv \sum_{k=0}^{t-1} \min(\mathbf{L}_*^{(n)}(k), \mathbf{R}_*^{(n)}(k)) \mathbf{1}_{\{\mathbf{T}_*^{(n)}(k) \leq \tau-1\}} , \\ \mathbf{D}_u^{(n)}(\tau, t) &\equiv \sum_{k=0}^{t-1} \left( \left\lceil \frac{\mathbf{L}_*^{(n)}(k)}{r^{\max}} \right\rceil - \left\lceil \frac{\mathbf{L}_*^{(n)}(k) - \mathbf{R}_*^{(n)}(k)}{r^{\max}} \right\rceil \right) \mathbf{1}_{\{\mathbf{T}_*^{(n)}(k) \leq \tau-1\}} .\end{aligned}$$

For example,  $\mathbf{D}^{(n)}(t, t)$  is simply the total number of files which have completed service by the end of slot  $[[t] - 1, [t]]$ . Moreover,  $\mathbf{D}^{(n)}(\tau, t)$  is increasing in  $\tau$  and  $t$  with  $\mathbf{D}^{(n)}(\cdot, 0) = 0$ , and for any  $(\tau_1, t_1) \leq (\tau_2, t_2)$ , we have that

$$\mathbf{D}^{(n)}(\tau_2, t_2) - \mathbf{D}^{(n)}(\tau_1, t_1) \leq \max(t_2 - t_1 + 1, \mathbf{F}^{(n)}(\tau_2) - \mathbf{F}^{(n)}(\tau_1)).$$

Similar bounds hold for  $\mathbf{D}_l^{(n)}(\tau, t)$  and  $\mathbf{D}_u^{(n)}(\tau, t)$ . Then, along a further subsequence of  $n$ , we have the following u.o.c. convergences to Lipschitz continuous (and hence differentiable *a.e.*) limiting functions,

$$\begin{aligned} \left( \mathbf{d}^{(n)}(\tau, t) = \frac{1}{n} \mathbf{D}^{(n)}(n\tau, nt), t \geq \tau \geq 0 \right) &\rightarrow (d(\tau, t), t \geq \tau \geq 0), \\ \left( \mathbf{d}_l^{(n)}(\tau, t) \equiv \frac{1}{n} \mathbf{D}_l^{(n)}(n\tau, nt), t \geq \tau \geq 0 \right) &\rightarrow (d_l(\tau, t), t \geq \tau \geq 0), \\ \left( \mathbf{d}_u^{(n)}(\tau, t) \equiv \frac{1}{n} \mathbf{D}_u^{(n)}(n\tau, nt), t \geq \tau \geq 0 \right) &\rightarrow (d_u(\tau, t), t \geq \tau \geq 0). \end{aligned}$$

The points  $(\tau, t)$  where the derivatives of the limiting functions exist are called regular. Then, for all regular  $t \geq \tau \geq 0$ , we have that  $0 \leq \frac{\partial d(\tau, t)}{\partial \tau} \leq \lambda$  and  $0 \leq \frac{\partial d(\tau, t)}{\partial t} \leq 1$ . Similar bounds hold for the derivatives of  $d_l(\tau, t)$  and  $d_u(\tau, t)$ .

Let  $\mathbf{Q}^{(n)}(\tau, t)$  denote the number of files which arrived before the end of slot  $[\tau - 1, \tau)$  but are still present at the end of slot  $[t - 1, t)$ ; so, for example,  $\mathbf{Q}^{(n)}(t) = \mathbf{Q}^{(n)}(t, t)$ . Then using the above defined processes, we have that,

$$\begin{aligned} \mathbf{Q}^{(n)}(\tau, t) &= \mathbf{F}^{(n)}(\tau) - \mathbf{D}^{(n)}(\tau, t), \\ \mathbf{Q}^{(n)}(t) &= \mathbf{Q}^{(n)}(t, t) = \mathbf{F}^{(n)}(t) - \mathbf{D}^{(n)}(t, t), \\ \bar{\mathbf{L}}^{(n)}(t) &= \mathbf{F}_l^{(n)}(t) - \mathbf{D}_l^{(n)}(t, t), \\ \bar{\mathbf{U}}^{(n)}(t) &= \mathbf{F}_u^{(n)}(t) - \mathbf{D}_u^{(n)}(t, t), \\ \mathbf{W}_i^{(n)}(t) &= [t] + 1 - \inf_{\tau} (\tau : \mathbf{Q}^{(n)}(\tau, t) \geq i), \quad i \in \{1, \dots, \mathbf{Q}^{(n)}(t)\}, \end{aligned}$$

and the following u.o.c. convergences to Lipschitz continuous limiting functions,

$$\begin{aligned} \left( \mathbf{q}^{(n)}(\tau, t) \equiv \frac{1}{n} \mathbf{Q}^{(n)}(n\tau, nt), t \geq \tau \geq 0 \right) &\rightarrow (q(\tau, t) \equiv f(\tau) - d(\tau, t), t \geq \tau \geq 0) , \\ \left( \mathbf{q}^{(n)}(t) \equiv \frac{1}{n} \mathbf{Q}^{(n)}(nt), t \geq 0 \right) &\rightarrow (q(t) \equiv f(t) - d(t, t), t \geq 0) , \\ \left( \bar{\mathbf{l}}^{(n)}(t) \equiv \frac{1}{n} \bar{\mathbf{L}}^{(n)}(nt), t \geq 0 \right) &\rightarrow (\bar{l}(t) \equiv f_l(t) - d_l(t, t), t \geq 0) , \\ \left( \bar{\mathbf{u}}^{(n)}(t) \equiv \frac{1}{n} \bar{\mathbf{U}}^{(n)}(nt), t \geq 0 \right) &\rightarrow (\bar{u}(t) \equiv f_u(t) - d_u(t, t), t \geq 0) , \end{aligned}$$

and the following weak convergence,

$$\left( \mathbf{w}_x^{(n)}(t) \equiv \frac{1}{n} \mathbf{W}_{\lfloor nx \rfloor + 1}^{(n)}(nt), x \geq 0, t \geq 0 \right) \rightarrow (w_x(t), x \geq 0, t \geq 0) ,$$

where, if  $q(t) > 0 = q(0, t)$ , then  $w_0(t)$  is given by the right-limit of  $w_x(t)$ , *i.e.*,

$$w_0(t) = \lim_{x \downarrow 0} w_x(t) = \lim_{x \downarrow 0} \left( t - \inf_{\tau} (\tau : q(\tau, t) \geq x) \right) = t - \sup_{\tau} (\tau : q(\tau, t) = 0) .$$

#### 4.3.0.3 Dynamics and derivatives of fluid limit for $t \geq 0$

The limit point obtained above,  $(q, \bar{l}, \bar{u}, w, f, f_l, f_u, d, d_l, d_u)$ , of the scaled version of the sequence  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$  is not necessarily unique. In particular, the limit point depends on the sequence of initial states  $\{S^{(n)}(0)\}$  and the convergent subsequence chosen. However, the following lemmas hold for all limit points of the scaled version of sequence  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$ .

**Lemma 4.1** *Consider any set of limiting functions derived from  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$ .*

*For any regular  $t \geq \tau \geq 0$ , if  $f_l(\tau) - d_l(\tau, t) > 0$ , then,*

$$\frac{\partial d_l(\tau, t)}{\partial t} \geq 1, \tag{4.6}$$

*and,*

$$\frac{\partial d_l(t, t)}{\partial t} = \frac{\partial d_l(\tau, t)}{\partial t} . \tag{4.7}$$



**Remark 4.4** *The condition  $f_1(\tau) - d_l(\tau, t) > 0$  means that there is a strictly positive amount of bit fluid with current sojourn time at least  $t - \tau$ . Then (4.6) ensures that this fluid will eventually get served, whereas, (4.7) ensures no service is given to any newer fluid while the older one remains in the system. This lemma however does not establish that the rate at which the fluid is being served is higher than the rate at which the fluid is entering the system; that is addressed later in Lemma 4.2.*

**Proof of Lemma 4.1** Fix a  $t \geq \tau \geq 0$  such that  $f_l(\tau) - d_l(\tau, t) > 0$ . Then by the assumption of bounded file size, we have  $q(\tau, t) > 0$ . Since  $q(\tau, \cdot)$  is Lipschitz continuous, there exists a  $\delta > 0$  and  $0 \leq t_1 < t_2$  such that  $t \in [t_1, t_2]$  and for all  $\hat{t} \in [t_1, t_2]$  we have  $q(\tau, \hat{t}) > 2\delta$ . By uniform convergence of  $q^{(n)}(\tau, \cdot) \rightarrow q(\tau, \cdot)$  over  $[t_1, t_2]$ , for all large  $n$  we have that over  $[t_1, t_2]$ ,  $\mathbf{q}^{(n)}(\tau, \cdot) > \delta$  and  $\mathbf{w}_x^{(n)}(\cdot) \geq (\tau - \cdot)$  for  $x \in [0, \delta]$ . Finally,

$$\begin{aligned}
& t_2 - t_1 + \frac{1}{n} \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{i_* (\mathbf{W}^{(n)}(k), \mathbf{R}^{(n)}(k)) \leq n\delta\}}, \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{\max_{1 \leq i \leq n\delta} \mathbf{W}_i^{(n)}(k) \mathbf{R}_i^{(n)}(k) \geq (n\tau - k)r^{\max}\}}, \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{\max_{1 \leq i \leq n\delta} \mathbf{R}_i^{(n)}(k) = r^{\max}\}} \xrightarrow{a.s.} t_2 - t_1.
\end{aligned} \tag{4.8}$$

That is,

$$\liminf_{n \rightarrow \infty} \mathbf{d}_l^{(n)}(\tau, t_2) - \mathbf{d}_l^{(n)}(\tau, t_1) \geq t_2 - t_1 \quad a.s.$$

and

$$\lim_{n \rightarrow \infty} (\mathbf{d}_l^{(n)}(t_2, t_2) - \mathbf{d}_l^{(n)}(t_1, t_1)) - (\mathbf{d}_l^{(n)}(\tau, t_2) - \mathbf{d}_l^{(n)}(\tau, t_1)) = 0 \quad a.s.$$

■

**Corollary 4.1** *By (4.4) and (4.6) in Lemma 4.1, for all  $t > T_0 \equiv r^{\max}$ , we have,*

$$w_0(t) < t,$$

i.e., all the fluid initially present in the system (recall  $\bar{l}(0)$ ) gets served by time  $T_0$ . Moreover, by (4.7) in Lemma 4.1, the fluid is served in a FCFS (first-come-first-serve) manner, therefore, for all  $t > T_0$ , we have that,

$$w_x(t) = \frac{q(t) - x}{\lambda}, \quad x \in [0, q(t)]; \quad (4.9)$$

$$\bar{l}(t) = \beta q(t); \quad (4.10)$$

$$\bar{u}(t) = \gamma q(t). \quad (4.11)$$

**Remark 4.5** To see (4.9), pick an  $x \in (0, q(t))$  and note that,

$$\tau \equiv t - w_x(t) \geq t - w_0(t) > 0 \quad \text{and} \quad q(\tau, t) \geq x > 0.$$

Then  $f(t) - f(\tau) = \lambda w_x(t)$ . By Lemma 4.1, this  $\lambda w_x(t)$  amount of file fluid that has entered the system since  $\tau$  must still be queued behind  $x$ . That is,  $q(t) - x = \lambda w_x(t)$ . Equations (4.10) and (4.11) follow similarly.

**Lemma 4.2** Consider any set of limiting functions derived from  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$ . For any regular  $t > T_0$ , if  $q(t) > 0$ , then,

$$\frac{\partial d(t, t)}{\partial t} = \gamma^{-1}, \quad \frac{\partial d_u(t, t)}{\partial t} = 1,$$

and hence,

$$\begin{aligned} q'(t) &= \lambda - \gamma^{-1} < 0, \\ \bar{u}'(t) &= \rho - 1 < 0, \\ w'_0(t) &= \frac{q'(t)}{\lambda} < 0. \end{aligned} \quad (4.12)$$

**Proof of Lemma 4.2** Fix a  $t > T_0$  such that  $q(t) > 0$ . Since  $\mathbf{w}_{(\cdot)}^{(n)}(T_0 + \cdot)$  converges u.o.c. to Lipschitz continuous  $w_{(\cdot)}(\cdot)$  (see (4.9)), we can pick a  $q(t) > \delta > 0$  and  $T_0 \leq t_1 < t_2$  such that  $t \in [t_1, t_2]$  and for all large  $n$  uniformly over  $[t_1, t_2]$ , we have  $r^{\max} \mathbf{w}_{\delta}^{(n)}(\cdot) \geq (r^{\max} - 1) \mathbf{w}_0^{(n)}(\cdot)$ . This choice of  $\delta$  implies that for any  $\hat{t} \in \{[nt_1], \dots, [nt_2] - 1\}$ , if a file in set  $\{1, \dots, [n\delta]\}$  sees a channel state of  $r^{\max}$ ,

then  $\mathbf{R}_*(\hat{t}) = r^{\max}$ . Rest of the proof proceeds similar to that of Lemma 4.1, except that we will *strengthen* the event associated with the Indicator function in (4.8). We have that,

$$\begin{aligned}
& t_2 - t_1 + \frac{1}{n} \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{ \mathbf{R}_*^{(n)}(k) = r^{\max} \}}, \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{ i^*(\mathbf{W}^{(n)}(k), \mathbf{R}^{(n)}(k)) \leq n\delta, \mathbf{R}_*^{(n)}(k) = r^{\max} \}}, \\
& = \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{ \max_{1 \leq i \leq n\delta} \mathbf{R}_i^{(n)}(k) = r^{\max} \}}, \\
& \geq \frac{1}{n} \sum_{k=nt_1}^{nt_2-1} \mathbf{1}_{\{ \max_{1 \leq i \leq n\delta} \mathbf{R}_i^{(n)}(k) \geq r^{\max} \}} \xrightarrow{a.s.} t_2 - t_1.
\end{aligned}$$

Then, along with Corollary 4.1, it follows that

$$\lim_{n \rightarrow \infty} \mathbf{d}_u^{(n)}(t_2, t_2) - \mathbf{d}_u^{(n)}(t_1, t_1) = t_2 - t_1 \quad a.s.$$

and

$$\lim_{n \rightarrow \infty} \mathbf{d}^{(n)}(t_2, t_2) - \mathbf{d}^{(n)}(t_1, t_1) = \gamma^{-1}(t_2 - t_1) \quad a.s.$$

■

**Corollary 4.2** *There exists a finite  $T_1$  (independent of the set  $\{\mathcal{X}^{(n)}, n = 1, 2, \dots\}$ ), such that for any set of limiting functions and for all  $t > T_1$ , we have that*

$$\bar{u}(t) + w_0(t) = 0 .$$

#### 4.3.0.4 Using Proposition 4.1 to conclude the proof of Theorem 4.1

Corollary 4.2 implies that,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \|\mathbf{S}^{(n)}(nT_1)\| = 0 \quad a.s. \quad (4.13)$$

Moreover, the sequence of random variables  $\{\frac{1}{n}\|\mathbf{S}^{(n)}(nT_1)\|, n = 1, 2, \dots\}$  is uniformly integrable since (see [53, p. 351]),

$$\frac{1}{n}\|\mathbf{S}^{(n)}(nT_1)\| \leq 1 + \mathbf{f}_u^{(n)}(T_1) + T_1 \xrightarrow{a.s.} \mathbb{E}(1 + \mathbf{f}_u^{(n)}(T_1) + T_1) = 1 + (\rho + 1)T_1 < \infty.$$

Then, the almost sure convergence in (4.13) along with uniform integrability implies the following convergence in the mean,

$$\lim_{n \rightarrow \infty} \mathbb{E} \left( \frac{1}{n} \|\mathbf{S}^{(n)}(nT_1)\| \right) = 0,$$

thus completing the proof of throughput optimality of the delay-driven MaxWeight.

## 4.4 Conclusion and extensions

An interesting extension of the system model is to allow that a file may arrive gradually over time instead of all at once (*i.e.*, similar to the case (ii) of [28]). Assuming the distribution of the time interval over which the first and the last bit of a file arrives has a *light* tail, the extension of the current result seems possible without much effort: the fluid limits of the arrival process (*e.g.*, the work arrival process  $f_u$ ) is indistinguishable for the two cases. Similarly, the assumption that channel  $\mathbf{R}_i(t)$  be i.i.d. over time can be relaxed to, *e.g.*,  $\mathbf{R}_i(t)$  forming a Markov chain in the set  $\{0, \dots, r^{\max}\}$  with a unique stationary distribution satisfying  $p = \mathbb{P}(\mathbf{R}_i(\cdot) = r^{\max}) > 0$ , and  $\mathbf{R}_i(\cdot)$  being drawn from the stationary distribution upon the arrival of a file. Other interesting extensions that seem possible, however, require little more effort are as follows: allowing for different classes to have a possibly different  $r^{\max}$  bits/slot, *i.e.*, the maximum rate (with non-zero probability) supported by the channel; and generalizing the scheduler to *weighted*-delay-driven MaxWeight, where the current sojourn time of each file is scaled by a fixed, class-dependant constant (see (4.1)).

Lastly, we would like to point out that the significance of the throughput-optimality result presented in this chapter is mostly technical: the proof indicates

that delay-driven MaxWeight scheduler relies on the ratio of sojourn times of a large number of users becoming close to 1, and at that point the scheduler is able to fully exploit the opportunistic gain and ensure stability. However, the ratio of sojourn times may approach unity only when sojourn times are unrealistically large. Moreover, the relation between the current sojourn time of a flow and its residual size is not clear. In the next chapter we investigate schedulers which explicitly use the residual file-size information and seek to minimize mean sojourn time.

## Chapter 5

# Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics

### 5.1 Overview and main contributions

In this chapter, we consider the scheduling of best effort flows<sup>1</sup> sharing a time varying wireless channel, with the objective of minimizing mean sojourn time. The key tradeoff involved is between *prioritizing flows with short residual sizes* and *maximizing opportunistic capacity gain by selecting flows that currently see good channels*. This tradeoff is explicitly characterized by introducing a new queueing model that involves servers with state-dependent *capacity regions*. In the transient case and for (bounding) polymatroid capacity regions, the optimal scheduler is given and used to obtain sub-optimality bounds for various heuristics. Using a mix of analysis and simulation two regimes are described: one where fully exploiting opportunistic gain is sufficient and further using residual flow sizes for scheduling will result in only minimal reduction in mean sojourn time, and the other, where the use of this information can indeed offer significant reduction. A new scheduler is proposed which performs well in both regimes.

#### Main contributions

Using a mix of analysis and simulation, we investigate features of the above-mentioned tradeoff. The main analytical results are given in Theorems 5.2 and 5.3.

---

<sup>1</sup>Each flow is associated with a unique user downloading (or uploading) a file from the base station; the terms flow, user, and file will sometimes be used interchangeably.

Theorem 5.2 characterizes the competitive ratio of flow-size-oblivious opportunistic schedulers, like Proportional Fair (PF) [6] or MaxQuantile [30] [31], for a *transient* system, and shows that the presence of opportunistic gain mitigates their sub-optimality. Using this, we characterize two regimes based on the “degree” of opportunistic gain:

- A regime – marked by high degree of opportunistic gain – where the use of residual flow-size information in scheduling will *not* result in a significant reduction in flows’ mean sojourn time.
- A regime – marked by low degree of opportunistic gain – where optimally using flow-size information alongside channel state information *may* result in a significant reduction in flows’ mean sojourn time.

More specifically, but still informally, if the opportunistic capacity of the wireless channel increases rapidly in the number of users, *e.g.*, as  $\log(n)$  or  $\log \log(n)$  where  $n$  is the number of users, then the mean sojourn time under a purely opportunistic scheduler like MaxQuantile or PF is only about 1–20% higher than the minimum possible. If, however, the opportunistic capacity increases more *slowly* (*e.g.*, as  $1 - a^n$  for  $a \in [0, 1)$ ), a significant reduction in mean sojourn time may be achievable if schedulers exploit the residual flow-size information. Using these insights, we propose a class of schedulers which offer good performance irrespective of the operating regime – this is analyzed in Theorem 5.3.

## 5.2 System model

We will first define a *heterogeneous* wireless system where the heterogeneity in the users’ wireless channels is captured by a single parameter, namely, their mean supported transmission rate. In the next subsection, we discuss how to convert the heterogeneous system into an equivalent system with *homogeneous* wireless channels.

The latter system permits certain simplifications for subsequent exposition and analysis. The details are as follows.

### 5.2.1 Wireless system with heterogeneous channels

We consider a continuous time system. Let  $\mathcal{X} \subseteq \mathbb{R}^2$  denote the connected coverage area of a base station. New users arrive at random times and random locations in  $\mathcal{X}$ , request to download (or upload) a file of random size, and leave once the download completes. The details are as follows.

Using an index set  $\mathcal{I} = \{1, 2, \dots\}$ , we uniquely index each user that enters the system. For each user  $i \in \mathcal{I}$ , let the random variables  $A_i \in \mathbb{R}$ ,  $\mathbf{X}_i \in \mathcal{X}$ , and  $\tilde{B}_i \in \mathbb{R}_+$  denote its arrival time, location, and the size of the file requested, respectively. Users are indexed in order of their arrival times, *i.e.*, for  $i < j$ , we have  $A_i \leq A_j$ . We assume that,

- $(A_i, i \in \mathcal{I})$  correspond to the jump times of a homogenous Poisson process of intensity  $\lambda$ ;
- $\mathbf{X}_i, i \in \mathcal{I}$  are i.i.d. with density  $f_{\mathbf{X}}(\cdot)$ , *i.e.*, for any measurable  $\mathcal{B} \subseteq \mathcal{X}$ , we have  $\mathbb{P}(\mathbf{X}_1 \in \mathcal{B}) = \int_{\mathcal{B}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$ ;
- $\tilde{B}_i, i \in \mathcal{I}$  are i.i.d. with density  $f_{\tilde{B}}(\cdot)$ .

The time-varying channel state of the  $i^{\text{th}}$  user is given by a stationary random process  $(\tilde{R}_i(t), t \in \mathbb{R})$ , where  $\tilde{R}_i(t) \in \mathbb{R}_+$  denotes the data/service rate supported by the user's channel at time  $t$ . Note that even though  $\tilde{R}_i(t)$  is defined for all  $t$ , it is relevant only over the time interval that user  $i$  spends in the system.

**Assumption 5.1** *We assume the following regarding the distribution of  $(\tilde{R}_i(t), t \in \mathbb{R})$ .*



- (i) *Marginal distribution:* Conditional on  $\mathbf{X}_i = \mathbf{x}$ , we assume that  $(\tilde{R}_i(t), t \in \mathbb{R})$  has the same marginal distribution as that of  $m(\mathbf{x})R$ , where  $R$  is a unit mean random variable and  $m : \mathcal{X} \rightarrow [\xi_1, \xi_2] \subset (0, \infty)$  is a given continuous function that captures the impact of pathloss/shadowing on the achievable transmission rate from the base station to location  $\mathbf{x} \in \mathcal{X}$ .
- (ii) *Separation of time-scales:* We assume that the channels change on a much faster time-scale than that of the user dynamics (arrivals and departures). So, for example, the channel coherence time – the smallest  $\tau$  such that  $\tilde{R}_i(t)$  and  $\tilde{R}_i(t + \tau)$  are independent – is much smaller than the mean inter-arrival time and mean sojourn time.

Assumption 5.1(i) will be used in the next subsection to *shift* the heterogeneity in the channels of users to their arrival processes, in the same manner as done in [22]. Assumption 5.1(ii) will be used in Section 5.4.2 to remove the randomness due to time-varying channels *without* losing any opportunistic gain associated with such channels. This is perhaps a key assumption which enables most of the results presented in this chapter and in [22].

Let random set  $Q(t) \subset \mathcal{I}$  denote the set of users with on going file transfers at time  $t$ , *i.e.*, the users which have arrived but have not completed their file download by time  $t$ . We will refer to the users in  $Q(t)$  as the users *present* at time  $t$ , and to  $Q(t)$  as the queue at time  $t$ . Cardinality of the set  $Q(t)$ , *i.e.*, the number of users present in the system at time  $t$ , is denoted by  $|Q(t)|$ . For any  $i \in Q(t)$ , let  $L_i(t) > 0$  denote the  $i^{\text{th}}$  user’s residual file size.

We will follow the convention that uppercase letters, *e.g.*  $Q(\cdot)$ , denote random quantities, whereas, lowercase letters, *e.g.*  $q(\cdot)$ , denote particular realizations. Bold letters are reserved for vectors. Also, we will make the natural distinction between “increasing” and “strictly increasing”, and so forth.

The state of the system at time  $t$  is given by,

$$S_t \equiv \left( Q(t); (L_i(t), i \in Q(t)); (\tilde{R}_i(t), i \in Q(t)); ((\mathbf{X}_i, A_i), i \in Q(t)) \right), \quad (5.1)$$

and let  $\mathcal{S}$  denote the space of all feasible realizations for the state  $s_t$ . We assume that the current system state is available for making a scheduling decision, and define a *scheduler* or a *scheduling policy* as a mapping  $i^* : \mathcal{S} \rightarrow \mathcal{I}$ . That is, when the system is in state  $s_t$ , the scheduler  $i^*(\cdot)$  selects a user  $i^*(s_t) \in q(t)$  to receive service. If for a given sample path we have  $i \in q(\cdot)$  over some interval  $[t_1, t_2]$ , then the total service received by user  $i$  over  $[t_1, t_2]$  is given by  $\int_{t_1}^{t_2} \tilde{r}_i(t) \mathbf{1}\{i^*(s_t) = i\} dt$ .

**Remark 5.1** *Since  $\tilde{R}_i(t)$  may depend on the past  $(\tilde{R}_i(\tau), \tau < t)$ , the most general definition for a scheduler would be as a mapping from the entire system sample path up to the present time (instead of just the current state). However, relying on the Assumption 5.1(ii) that the coherence time is much smaller than the time-scale of user dynamics, we restrict our attention to schedulers that use only the current system state and make no use of the entire history.*

### 5.2.2 Equivalent system with i.i.d. channels but heterogeneous file sizes

In the system described above, the channel state process of a user at location  $\mathbf{x}$  has the same marginal distribution as that of  $m(\mathbf{x})R$ , and therefore, a mean supported transmission/service rate of  $m(\mathbf{x})$ . Consider a *second* system (constructed on the same probability space) where all users' channels have the same marginal distribution as that of  $R$ , but the file sizes of users at location  $\mathbf{x}$  are scaled by a factor of  $1/m(\mathbf{x})$ . At any time  $t$ , the system state of the second system can be derived from the system state of the first system by appropriately scaling  $(L_i(t), i \in Q(t))$  and  $(\tilde{R}_i(t), i \in Q(t))$  based on  $(\mathbf{X}_i, i \in Q(t))$ , and keeping the remaining elements of  $S_t$ , in particular,  $Q(t)$ , unchanged. That is, any scheduler for the first system can be translated into a scheduler for the second system, such that the user arrival and departure times, as

captured by  $Q(t)$ , are identical in the two systems. The second system however has the *advantage* that the channel states of the users are independent of their identities and locations.

Therefore, from this point on, we exclusively focus on the *second* system where the user arrivals  $(A_i, i \in \mathcal{I})$  and locations  $(X_i, i \in \mathcal{I})$  are statistically identical to those in the first system, but the user channels  $((R_i(t), t \in \mathbb{R}), i \in \mathcal{I})$  and file sizes  $(B_i, i \in \mathcal{I})$  have the following modified marginal distributions: for all  $i \in \mathcal{I}$ ,

- the channel process  $(R_i(t), t \geq 0)$  has the same marginal distribution as that of the random variable  $R$ ,
- the file size  $B_i$  has density  $f_B(\cdot)$  given by,

$$f_B(b) \equiv \int_{\mathcal{X}} m(\mathbf{x}) f_{\bar{B}}(m(\mathbf{x})b) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \quad b \in \mathbb{R}_+.$$

We continue to model a scheduler  $i^*(\cdot)$  as a mapping from the current system state to the current queue,  $i^*(s_t) \in q(t)$ .

Next we define a new queueing model using a server with a state-dependent but deterministic *capacity region* – such a server will subsequently be used to replace the randomly-varying channels in the above system *without* losing the opportunistic capacity gains associated with such channels.

### 5.3 Servers with state-dependent capacity regions and $M/GI/\mathcal{C}$ queue

Let  $\mathcal{C}_1, \mathcal{C}_2, \dots$  be sequence of *nested* capacity regions with the following properties:

- (i) For any  $n \geq 1$ ,  $\mathcal{C}_n$  is a compact, convex, and coordinate-convex region of  $\mathbb{R}_+^n$ .

(ii)  $\mathcal{C}_n$  is symmetric, *i.e.*, if a rate vector  $\boldsymbol{\mu} \equiv (\mu_1, \dots, \mu_n) \in \mathcal{C}_n$ , then any vector given by a permutation of  $\boldsymbol{\mu}$ 's components also lies in  $\mathcal{C}_n$ .

(iii) For  $k$  and  $n$  such that  $1 \leq k \leq n$ , we have  $\mathcal{C}_k = \mathbb{R}_+^k \cap \mathcal{C}_n$ .

Let  $\mathcal{C} \equiv (\mathcal{C}_n, n \geq 1)$ . Consider a server with the following property: if there are  $n$  jobs (files) in the system, the server can process them simultaneously according to any speed (rate) vector from the region  $\mathcal{C}_n$ . For example, suppose over some interval  $[t_1, t_2]$  the  $n$  files are served at rate  $\boldsymbol{\mu}(t) \in \mathcal{C}_n$ , then the file backlog  $\mathbf{l}(t) \equiv (l_i(t), 1 \leq i \leq n)$  over  $t \in [t_1, t_2]$  evolves as,

$$\mathbf{l}(t) = \left( \mathbf{l}(t_1) - \int_{t_1}^t \boldsymbol{\mu}(\tau) d\tau \right)^+.$$

By replacing the unit capacity server in a conventional  $M/GI/1$  queue with the server described above, we obtain a new queueing model which we refer to as an  $M/GI/\mathcal{C}$  queue. Note that the queues  $M/GI/1$  and  $M/GI/\infty$  queues are special cases of the  $M/GI/\mathcal{C}$  queue. Indeed the latter reduces to  $M/GI/1$  if  $\mathcal{C}_n$  is unit  $n$ -simplex, *i.e.*,

$$\mathcal{C}_n = \{ \boldsymbol{\mu} \in \mathbb{R}_+^n : \|\boldsymbol{\mu}\|_1 \leq 1 \}, \quad n = 1, 2, \dots$$

and to  $M/GI/\infty$  if  $\mathcal{C}_n$  is unit  $n$ -cube, *i.e.*,

$$\mathcal{C}_n = \{ \boldsymbol{\mu} \in \mathbb{R}_+^n : \|\boldsymbol{\mu}\|_\infty \leq 1 \}, \quad n = 1, 2, \dots$$

As will become clear in the next section, we will be interested in server capacity regions that are *in between* unit simplices and unit cubes.

Next, we will show that the system with time-varying channels and flow dynamics, described in Section 5.2.2, can be reduced to an instance of an  $M/GI/\mathcal{C}$  queue by *appropriately defining the capacity regions*  $\mathcal{C}_n$ ,  $n \geq 1$  (Section 5.4.1), and *using a time-scale separation argument* (Section 5.4.2).

## 5.4 Dynamic wireless system as an $M/GI/\mathcal{C}$ queue

We will set  $\mathcal{C}_n$  equal to the opportunistic capacity region of a *static* wireless system, *i.e.*, a system with a fixed number  $n$  of infinitely-backlogged flows. A formal description of these regions follows.

### 5.4.1 Opportunistic capacity region

Throughout this subsection, we consider a static system with a fixed number  $n$  of users. As before, for all  $i \in \{1, \dots, n\}$ , the channel processes  $(R_i(t), t \geq 0)$  are independent, stationary and ergodic with the same marginal distribution as that of  $R$ , and recall that  $\mathbb{E}[R] = 1$ . We define the opportunistic capacity region  $\mathcal{C}_n \subset [0, 1]^n$  of such a system as the set of long-run average service rates that can be jointly allocated to the  $n$  users<sup>2</sup>.

Specifically, for any  $\mathbf{r} = (r_i, 1 \leq i \leq n) \in \mathbb{R}_+^n$ , let  $\mathcal{C}_n(\mathbf{r}) \subset \mathbb{R}_+^n$  be the convex hull of the origin and  $n$  points,

$$(r_1, 0, \dots, 0), (0, r_2, 0, \dots, 0), \dots, (0, \dots, 0, r_n).$$

Note that  $\mathcal{C}_n(\mathbf{r})$  denotes the set of long-run average service rates that can be jointly allocated to the  $n$  users, conditional on the joint channel being in state

$$\mathbf{R}(t) \equiv (R_i(t), 1 \leq i \leq n) = \mathbf{r}.$$

Let  $\mathcal{V}_n$  be the set of all measurable function  $\mathbf{v}(\cdot)$  that map a joint channel state  $\mathbf{r} \in \mathbb{R}_+^n$  to an average service rate vector in  $\mathcal{C}_n(\mathbf{r})$ , *i.e.*,  $\mathcal{V} \equiv \{\mathbf{v}(\cdot) : \forall \mathbf{r} \in \mathbb{R}_+^n, \mathbf{v}(\mathbf{r}) \in \mathcal{C}_n(\mathbf{r})\}$ .

Finally, the opportunistic capacity region of  $n$  channels,  $\mathcal{C}_n$ , is defined as,

$$\mathcal{C}_n \equiv \{\boldsymbol{\mu} \in [0, 1]^n : \boldsymbol{\mu} \leq \mathbb{E}[\mathbf{v}(\mathbf{R})] \text{ for some } \mathbf{v}(\cdot) \in \mathcal{V}_n\}, \quad (5.2)$$

---

<sup>2</sup>The capacity region is a function of both the number of users  $n$  and the distribution of random  $R$ , however, since the distribution of  $R$  is assumed fixed throughout the chapter, the notation  $\mathcal{C}_n$  explicitly shows the dependence on  $n$  only.

where expectation is with respect to random  $\mathbf{R}$  drawn from the  $n$ -product of the distribution of  $R$ .

Any extremal point of the capacity region  $\mathcal{C}_n$  is achievable by a very simple, distribution-oblivious, opportunistic scheduler; for details see [22, Lemma 2.2] or Lemma 2.1.

For later use, in the following we define a polymatroid outer bound for the capacity region  $\mathcal{C}_n$ .

### The tightest polymatroid region containing $\mathcal{C}_n$

Let  $R_1, R_2, \dots$  be i.i.d. copies of  $R$ . For any integer  $k > 0$ , let

$$g_k \equiv \mathbb{E} [ \max(R_1, \dots, R_k) ], \quad (5.3)$$

and let  $g_0 \equiv 0$ . This function has the following interpretation: no single user can get an average rate exceeding  $g_1$ , no two users can get a total average rate exceeding  $g_2$ , and so on. We will refer to  $(g_k, k \geq 0)$  as the opportunistic *capacity function*. We have that  $g_k$  is concave increasing in  $k$ , *i.e.*, for any  $k \geq 0$ ,  $g_{k+1} - g_k \geq g_{k+2} - g_{k+1} \geq 0$ , and therefore, the capacity per user  $g_k/k$  is decreasing in  $k$ .

Then using  $(g_k, k \leq n)$ , we can define a polymatroid<sup>3</sup>  $\bar{\mathcal{C}}_n$  as follows,

$$\bar{\mathcal{C}}_n \equiv \left\{ \boldsymbol{\mu} \in [0, 1]^n : \forall \mathcal{K} \subseteq \{1, \dots, n\}, \sum_{k \in \mathcal{K}} \mu_k \leq g_{|\mathcal{K}|} \right\}.$$

See Fig. 5.1 for an illustration of  $\mathcal{C}_n$  and  $\bar{\mathcal{C}}_n$ . For any  $n$ , we have that  $\mathcal{C}_n \subseteq \bar{\mathcal{C}}_n$ . We also note the following.

- In general, we cannot completely construct  $\mathcal{C}_n$  from  $(g_k, k \leq n)$  alone. Nevertheless, we can completely construct the outer bound  $\bar{\mathcal{C}}_n$ .

---

<sup>3</sup>See, *e.g.*, [54, p. 767] for definition of polymatroid.

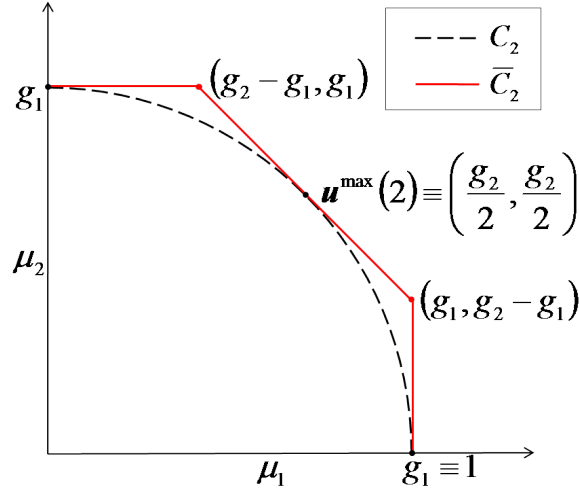


Figure 5.1: Capacity region  $\mathcal{C}_2$  and the tightest polymatroid upper bound  $\bar{\mathcal{C}}_2$ .

- Region  $\bar{\mathcal{C}}_n$  is the tightest polymatroid outer bound for  $\mathcal{C}_n$ .

In special cases of time-varying channels, *e.g.*, on-off channels or information theoretic block fading multiaccess channels (as used in [13]), we have  $\mathcal{C}_n = \bar{\mathcal{C}}_n$ . In fact, an alternative way to define the outer bound  $\bar{\mathcal{C}}_n$  is as follows:  $\bar{\mathcal{C}}_n$  is the region obtained by replacing  $\mathcal{C}_n(\mathbf{r})$  in the definition of  $\mathcal{C}_n$  by the tightest polymatroid containing  $\mathcal{C}_n(\mathbf{r})$ . Let  $\bar{\mathcal{C}} \equiv (\bar{\mathcal{C}}_n, n \geq 1)$ .

**Remark 5.2** We say a vector  $\mathbf{a} \in \mathbb{R}^n$  dominates a vector  $\mathbf{b} \in \mathbb{R}^n$  if  $\mathbf{a} \geq \mathbf{b}$ , and a point is maximal in a region if it is not dominated by any other point of that region. Note that a polymatroid capacity region has the nice property that all maximal elements have the same  $L_1$  norm (the total service rate). For example, the  $L_1$  norm of all maximal elements of  $\bar{\mathcal{C}}_n$  is  $g_n$ . Therefore, a scheduler for an  $M/GI/\bar{\mathcal{C}}$  queue will not have to tradeoff maximizing total service rate with prioritizing short files.

Let  $\mathbf{u}^{\max}(n) \equiv (\frac{g_n}{n}, \dots, \frac{g_n}{n}) \in \mathbb{R}_n$ . We have that  $\mathbf{u}^{\max}(n)$  lies in  $\mathcal{C}_n$ , and

$$g_n = \|\mathbf{u}^{\max}(n)\|_1 = \max_{\boldsymbol{\mu} \in \mathcal{C}_n} \|\boldsymbol{\mu}\|_1 = \max_{\boldsymbol{\mu} \in \bar{\mathcal{C}}_n} \|\boldsymbol{\mu}\|_1.$$

Therefore, we will refer to  $\mathbf{u}^{max}(n)$  as the *max-sum-rate point* of region  $\mathcal{C}_n$  as well as  $\bar{\mathcal{C}}_n$ .

#### 5.4.2 Time-scale separation argument and reduction to an $M/GI/\mathcal{C}$ queue

We return to the homogeneous dynamic system introduced in Section 5.2.2, and recall the Assumption 5.1(ii) that channels vary at a much faster time-scale than that of the user dynamics. Therefore, we can further assume that at any time  $t$ , conditional on  $|Q(t)| = n$ , the  $n$  users can be jointly served at any rate from the capacity region  $\mathcal{C}_n$ . That is, we can redefine the scheduler as a mapping from

$$(q(t); (l_i(t), i \in q(t)); ((\mathbf{x}_i, a_i), i \in q(t)))$$

to the capacity region  $\mathcal{C}_{|q(t)|}$ . For example, if for  $t \in [t_1, t_2]$ , we have  $|q(t)| = n$  and the scheduler serves at rate  $\boldsymbol{\mu}(t) \in \mathcal{C}_n$ , then the file backlog  $\mathbf{l}(t) \equiv (l_i(t), i \in q(t_1))$  over  $t \in [t_1, t_2]$  evolves as,

$$\mathbf{l}(t) = \mathbf{l}(t_1) - \int_{t_1}^t \boldsymbol{\mu}(\tau) d\tau .$$

This new system is simply an  $M/GI/\mathcal{C}$  queue, where the arrivals are Poisson with rate  $\lambda$ , the file sizes have density  $f_B(\cdot)$ , and the server capacity region at time  $t$  depends on only the element  $|q(t)|$  of the system state. The reason for converting the original system with heterogeneous channels to the one with homogeneous channels now becomes clear: it removed the dependence of the server capacity region  $\mathcal{C}_{|q(t)|}$  on the *locations* of the users in  $q(t)$ .

The work in [22] can be seen as analyzing an  $M/GI/\mathcal{C}$  queue under the scheduler that, conditional on  $|Q(t)| = n$ , serves at the max-sum-rate point  $\mathbf{u}^{max}(n) \in \mathcal{C}_n$ , and therefore, simultaneously serves each of the  $n$  users in the current queue at rate  $g_n/n$ . Following [22], we will call this the Opportunistic Processor Sharing (OPS) scheduler. This scheduler is clearly oblivious to file sizes.



**Remark 5.3** *A few observations regarding the usefulness of the  $M/GI/\mathbf{C}$  queueing model are in order. By reducing the original system to an  $M/GI/\mathbf{C}$  queue, we have replaced the state-dependent and randomly varying server (i.e., wireless channel) with a state-dependent but deterministic server without losing the opportunistic gain associated with a randomly varying server. In the latter system,*

- *the tradeoff mentioned in the beginning of Section 5.1 is manifested as a tradeoff between maximizing total service rate (e.g., by picking the rate point  $\mathbf{u}^{\max}(\cdot)$ ) and giving more rate to shorter files;*
- *moreover, the extent of opportunistic gain present in the channel is explicitly characterized by the marginal rate of increase of opportunistic capacity function  $(g_n, n \geq 0)$ .*

*Therefore, the  $M/GI/\mathbf{C}$  queue is better suited to investigate the scheduling problem, even if by simulation methods only.*

*The  $M/GI/\mathbf{C}$  queue also seems more amenable to an analytical solution than the original system; indeed, [22] gives analytical results for at least one particular scheduler for this queue. In the apparently distantly-related problem of admission control in a web server [55] and that of the microprocessor speed scaling [56] [57], similar queueing models appeared and perhaps some of the techniques can be borrowed from (or lent to) these fields – this will be further explored in the future.*

*We also note that the  $M/GI/\bar{\mathbf{C}}$  queue – a simpler to analyze than the  $M/GI/\mathbf{C}$  queue due to the absence of above-mentioned tradeoff – can be used to obtain a lower bound on the optimal mean sojourn time in  $M/GI/\mathbf{C}$  queue. To this end, in the next section, we will characterize the optimal scheduler under polymatroid capacity regions  $(\bar{\mathbf{C}}_k, k > 0)$  for a transient system, and use this to bound the sub-optimality of OPS.*

## 5.5 Transient system

Throughout this section, we consider a transient system that starts with a given number  $n$  of files, *i.e.*,  $q(0) = \{1, \dots, n\}$ , of arbitrary sizes

$$\mathbf{l}(0) \equiv (l_1(0), \dots, l_n(0)) \in (0, \infty)^n,$$

and there are no further arrivals. The opportunistic capacity regions  $\mathcal{C}$  and  $\bar{\mathcal{C}}$ , along with the corresponding concave increasing capacity function  $(g_k, k \geq 0)$  with  $g_0 = 0$ , are specified. Since there are no further arrivals, the system state at time  $t$  is simply given by  $(q(t); (l_i(t), i \in q(t)))$ . When there are  $k \leq n$  files in the system, they can be served at any rate from the region  $\mathcal{C}_k \subseteq \mathcal{C}_n$  (or  $\bar{\mathcal{C}}_k \subseteq \bar{\mathcal{C}}_n$ , depending upon the context). Let  $\Psi$  and  $\bar{\Psi}$  be the set of all functions (schedulers) that map any system state  $(q(t); \mathbf{l}(t))$  to a rate vector in  $\mathcal{C}_{|q(t)|}$  and  $\bar{\mathcal{C}}_{|q(t)|}$ , respectively.

For any scheduler  $\psi(\cdot)$  and initial state  $(q(0); \mathbf{l}(0))$ , let  $((q^\psi(t), \mathbf{l}^\psi(t)), t \geq 0)$  be the associated system sample path. The total sojourn time (or cost) under  $\psi(\cdot)$  is denoted by  $c_\psi(\mathbf{l}(0); |q(0)|)$  and given as follows,

$$c_\psi(\mathbf{l}(0); |q(0)|) \equiv \int_0^\infty |q^\psi(t)| dt. \quad (5.4)$$

We are interested in schedulers in  $\Psi$  that minimize the cost starting in any state  $(q(0); \mathbf{l}(0))$ . We naturally have  $\Psi \subseteq \bar{\Psi}$ , and therefore,

$$\min_{\bar{\psi} \in \bar{\Psi}} c_{\bar{\psi}}(\cdot) \leq \min_{\psi \in \Psi} c_\psi(\cdot). \quad (5.5)$$

However, only schedulers from  $\Psi$  are feasible in an actual system which has a server with capacity region  $\mathcal{C}_n$ . The rest of this section is organized as follows:

- An optimal scheduler in  $\bar{\Psi}$  along with an expression for optimal cost is given in Section 5.5.1.
- An expression for cost under OPS, which lies in  $\Psi$  (and thus also in  $\bar{\Psi}$ ), is given in Section 5.5.2.

- The competitive ratio of OPS under particular capacity functions is obtained in Section 5.5.3;
- That the performance under OPS deteriorates as  $n$  becomes large and opportunistic capacity function  $g_n$  saturates is elaborated in Section 5.5.4.
- A scheduler belonging to the set  $\Psi$  but with cost *close* to the that of the optimal scheduler in  $\bar{\Psi}$  is proposed in Section 5.5.5.

### 5.5.1 Optimal scheduler for polymatroid capacity regions $\bar{\mathcal{C}}$

We begin by defining the *Shortest Remaining Processing Time - Highest Possible Rate* (SRPT-HPR) scheduler for polymatroid capacity regions, *i.e.*, SRPT-HPR lies in  $\bar{\Psi}$ .

**Definition 5.1** Consider a system with a concave increasing opportunistic capacity function  $(g_k, k \geq 0)$  and the associated capacity regions  $(\bar{\mathcal{C}}_k, k \geq 0)$ . At any time  $t$  and for any integer  $n \equiv |q(t)| > 0$ , let  $l_1(t) \leq l_2(t) \leq \dots \leq l_n(t)$  be the remaining file sizes of the  $n$  files. Then SRPT-HPR serves the files at rate vector

$$\mathbf{s}(\mathbf{l}(t); n) = (s_i(\mathbf{l}(t); n), 1 \leq i \leq n) \in \bar{\mathcal{C}}_n$$

given as follows: for any  $i \in \{1, \dots, n\}$ , we have,

$$s_i(\mathbf{l}(t); n) \equiv g_i - g_{i-1} .$$

Note that service rate allocation under SRPT-HPR depends only on the ordering of file sizes, and the shortest file is given as much rate as possible, *i.e.*,  $g_1$ , and having made that allocation, the next shortest file is given as much rate as possible, *i.e.*,  $g_2 - g_1$ , and so on.

Note that a server with a polymatroid capacity region  $\bar{\mathcal{C}}_n$  can be decomposed into  $n$  conventional servers (like the one in queue  $M/GI/1$ ) with different speeds,

such servers are usually called parallel *uniform* servers [4]. More specifically, consider  $n$  servers where the speed of the  $k^{\text{th}}$  server is given by  $g_k - g_{k-1}$ . Then each maximal vertex of the polymatroid  $\bar{\mathcal{C}}_n$  corresponds to an allocation of the  $n$  servers to the  $n$  users (with one user per server and one server per user), and vice versa. All other maximal points of  $\bar{\mathcal{C}}_n$  can further be achieved by preempting and time-sharing the  $n$  servers among the users.

Therefore, the problem of minimizing the total sojourn time under a polymatroid capacity region  $\bar{\mathcal{C}}_n$  is the same as minimizing the total sojourn time under  $n$  parallel uniform servers with preemption permitted. For the latter system, [58] (see [4], pp. 134–136 for a proof) showed that the optimal scheduler is the one which allocates the fastest available server to the shortest available job/file – this indeed translates to SRPT-HPR in our setting. An alternative proof of the optimality of SRPT-HPR with an explicit expression for total sojourn time is given in the appendix at the end of the chapter. We summarize the results in the following lemma and theorem.

**Lemma 5.1** *For a given concave increasing capacity function ( $g_k, k \geq 0$ ), and initial state ( $q(0); (l_k(0), 1 \leq k \leq |q(0)|)$ ), the cost under SRPT-HPR  $c_s(\cdot)$  is given by,*

$$c_s(\mathbf{l}(0); |q(0)|) = \sum_{k=1}^{|q(0)|} \theta_k l_{(k)}(0) , \quad (5.6)$$

where  $l_{(k)}(0)$  denote the  $k^{\text{th}}$  largest file, i.e.,  $l_{(1)}(0) \geq \dots \geq l_{(n)}(0)$ , and  $(\theta_k, k > 0)$  are given as follows. Let  $\theta_0 \equiv 0$  and for  $k \geq 0$ , let  $\Delta_k \equiv g_{k+1} - g_k$ . Then for all  $n \geq 0$ ,

$$(\boldsymbol{\theta} * \boldsymbol{\Delta})(n) \equiv \sum_{k=0}^n \Delta_k \theta_{n-k} = n. \quad (5.7)$$

Using (5.7),  $\theta_{n+1}$  can be computed from  $(\theta_k, 0 \leq k \leq n)$  for any given capacity function  $(g_k, k \geq 0)$ .

**Remark 5.4**  $\theta_k$  can be interpreted as time (or cost) per unit data in the  $k^{\text{th}}$  file. As shown in the appendix at the end of the chapter,  $\theta_k$  is increasing in  $k$ .

**Theorem 5.1** Consider a transient system with polymatroid capacity regions  $(\bar{\mathcal{C}}_k, k \geq 0)$  and starting with any number  $|q(0)| > 0$  of files of arbitrary sizes  $\mathbf{l}(0) = (l_k(0), k \in q(0))$ . Then SRPT-HPR minimizes the cost given by (5.4), i.e.,

$$c_s(\mathbf{l}(0); |q(0)|) = \min_{\psi \in \bar{\Psi}} c_\psi(\mathbf{l}(0); |q(0)|).$$

Next, we define OPS scheduler which is not limited to polymatroid capacity regions.

### 5.5.2 OPS scheduler for capacity regions $\bar{\mathcal{C}}$ and $\mathcal{C}$

The scheduling decision under OPS depends only on  $(g_k, k \geq 0)$  and is the same for  $\mathcal{C}$  and  $\bar{\mathcal{C}}$ . It is defined as follows.

**Definition 5.2** Consider a system with a concave increasing opportunistic capacity function  $(g_k, k \geq 0)$  and the associated capacity regions  $(\mathcal{C}_k, k \geq 0)$  or  $(\bar{\mathcal{C}}_k, k \geq 0)$ . At any time  $t$  and for any integer  $n \equiv |q(t)| > 0$ , OPS serves each of the  $n$  files at rate  $\frac{g_n}{n}$ . Therefore, the scheduling decision for capacity region  $\mathcal{C}_n$  is identical to that for  $\bar{\mathcal{C}}_n$ .

**Lemma 5.2** For a given capacity function  $(g_k, k \geq 0)$ , and initial state

$$(q(0); (l_k(0), 1 \leq k \leq |q(0)|)),$$

the cost under OPS  $c_p(\cdot)$  is given by,

$$c_p(\mathbf{l}(0); |q(0)|) = \sum_{k=1}^{|q(0)|} \pi_k l_{(k)}(0), \quad (5.8)$$

where, for any  $k > 0$ ,

$$\pi_k \equiv \frac{k^2}{g_k} - \frac{(k-1)^2}{g_{k-1}}. \quad (5.9)$$

See appendix at the end of the chapter for proof.

### 5.5.3 Competitive ratio of OPS

For a given transient system – *i.e.*, a capacity function  $(g_k, k \geq 0)$  along with the *actual* capacity regions  $\mathcal{C}$  and the *outer bound*  $\bar{\mathcal{C}}$  – we will refer to the ratio  $\frac{c_p(\cdot)}{c_s(\cdot)}$  as the competitive ratio of OPS when starting in state  $(\cdot)$ . When the starting state or an element of it is not specified, the competitive ratio is intended to mean the supremum over unspecified elements.

**Remark 5.5** *Competitive ratio has the following interpretation. Consider a transient system starting with  $n$  files of sizes  $\mathbf{l} = (l_1, \dots, l_n)$ . In the interest of clarity, we reiterate that, given only  $(g_k, 0 \leq k \leq n)$ , the largest possible capacity region that the server can have is the polymatroid  $\bar{\mathcal{C}}_n$ . Therefore, the smallest possible cost a scheduler can have is  $c_s(n; \mathbf{l})$ . Then, the competitive ratio  $\frac{c_p(n; \mathbf{l})}{c_s(n; \mathbf{l})}$  gives us a bound on the sub-optimality of OPS when starting in state  $(\{1, \dots, n\}; \mathbf{l})$ . Moreover, this bound is the tightest possible if only  $(g_k, 0 \leq k \leq n)$  is specified instead of the complete actual capacity region  $\mathcal{C}_n$ .*

It is easy to show that if  $g_k = k$  for all  $k \geq 0$  (regions  $\mathcal{C}_k$  and  $\bar{\mathcal{C}}_k$  are  $k$ -cubes), then  $\theta_k = \pi_k = 1$  and we simply have  $\frac{c_p(\cdot)}{c_s(\cdot)} = 1$ . In the following, we investigate this ratio for capacity functions of the type  $g_k = \frac{1-a^k}{1-a}$  (equivalently,  $\Delta_k = a^k$ ) for any fixed  $a \in [0, 1)$ . Prior to doing so, we offer three comments on this choice of capacity function.

First, it corresponds to the capacity function of a system where  $R$  is an on-off random variable, with  $\mathbb{P}(R = 0) = a$  and  $\mathbb{P}(R = \frac{1}{1-a}) = 1 - a$ .

Second, parameter  $a$  provides a measure for the opportunistic gain present in the system: for any fixed  $k$ , the higher the value of  $a$ , the greater the marginal opportunistic capacity  $\Delta_k$ . Setting  $a = 0$  corresponds to a non-opportunistic work-conserving system with a single unit speed server.

Third, if  $R$  takes values in a bounded set  $[0, r^{max}]$ , then  $R$  is stochastically dominated by an on-off random variable  $\tilde{R} \in \{0, r^{max}\}$  with  $\mathbb{P}(\tilde{R} = r^{max}) \equiv \mathbb{E}[R]/r^{max}$ . Therefore, the capacity function associated with  $R$  can be bounded above by  $(g_k, k \geq 0)$  with  $a = 1 - \mathbb{E}[R]/r^{max}$ . The capacity function associated with  $R$  can also be lower bounded by a scaled version of  $(g_k, k \geq 0)$ . The significance of the results presented below is either a monotonicity in  $a$  or independence from  $a$ , therefore, this simple choice of capacity function still provides significant insight.

**Fact 5.1** *With  $g_k = \frac{1-a^k}{1-a}$  for some fixed  $a \in [0, 1)$ , we can take the  $z$ -transform of both sides of (5.7) and show that the corresponding  $(\theta_k, k > 0)$  are given by,*

$$\theta_k = \frac{k}{g_\infty} + a, \quad (5.10)$$

where  $g_\infty \equiv \frac{1}{1-a}$ .

To indicate the dependence of cost on the parameter  $a$ , in this subsection we will explicitly write  $c_s(\cdot; \cdot; a)$  and  $c_p(\cdot; \cdot; a)$ . Moreover,  $\mathbf{1}$  in  $c(\mathbf{1}; n; \cdot)$  will denote an  $n$  dimensional vector of all ones. We have the following result; see appendix at the end of the chapter for proof.

**Theorem 5.2** *For any  $a \in [0, 1)$ , let  $g_k = \frac{1-a^k}{1-a}$  for all  $k \geq 0$ . Let  $c_s(\mathbf{l}; n; a)$  and  $c_p(\mathbf{l}; n; a)$  be the costs under SRPT-HPR and OPS respectively, for capacity function parameter  $a$  and when starting with  $n$  files of sizes given in  $\mathbf{l} \equiv (l_1, \dots, l_n)$ . Then for any fixed  $a$ ,*

$$\sup_{n \geq 1, \mathbf{l} \in (0, \infty)^n} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} = \sup_{n \geq 1} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = 2. \quad (5.11)$$

More precisely, as  $n \rightarrow \infty$ , we have,

$$\frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} \uparrow 2.$$

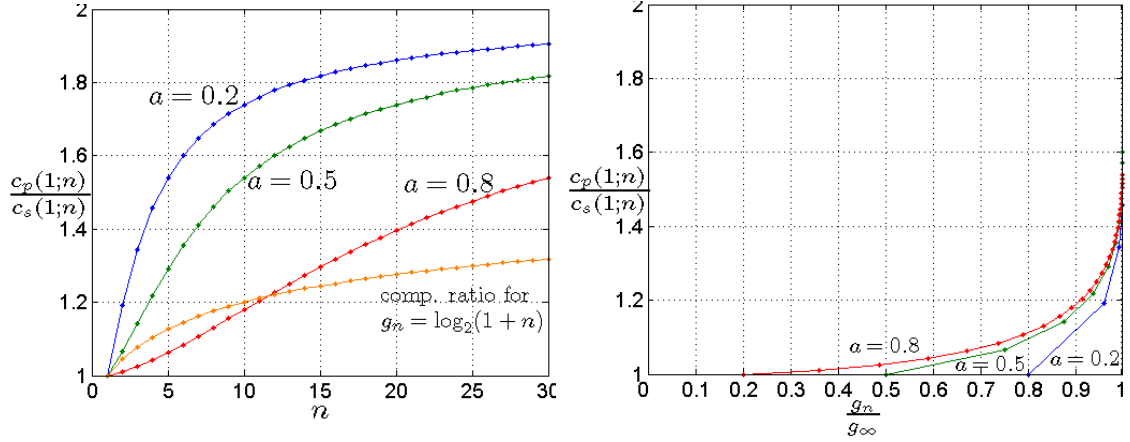


Figure 5.2: Competitive ratio of OPS plotted versus (left)  $n$ , (right)  $\frac{g_n}{g_\infty}$ .

Also, for any fixed  $n > 0$ ,

$$\sup_{a \in [0,1], \mathbf{l} \in (0, \infty)^n} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} = \sup_{a \in [0,1]} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = \frac{2}{1 + 1/n}.$$

More precisely, as  $a \rightarrow 0$ , we have,

$$\frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} \uparrow \frac{2}{1 + 1/n}.$$

#### 5.5.4 Discussion

The following useful observations can be made based on Theorem 5.2.

- (i) For any fixed  $n$ , the competitive ratio of OPS *decreases* as the parameter  $a$  – a measure of opportunistic gain – increases; see Fig. 5.2 (left).
- (ii) However, *irrespective of*  $a$ , the competitive ratio of OPS is exactly 2, which is achieved only as a monotonic limit in  $n$ . An intuitive explanation for this is as follows: the opportunistic capacity  $g_k$  is bounded above by the limit  $g_\infty \equiv \frac{1}{1-a}$ , therefore for any fixed  $a$  and correspondingly large  $n$  (where  $n$  is the number of files), the opportunistic system *appears* non-opportunistic with constant capacity  $g_\infty$ ; thus the competitive ratio 2 which is the same as in the case of non-opportunistic systems ( $a = 0$ ).



(iii) See Fig. 5.2 (right) for a plot of competitive ratio as a function of  $n$  versus the capacity ratio  $\frac{g_n}{g_\infty}$ . We note that the competitive ratio is under 1.2, for all  $n$  such that  $\frac{g_n}{g_\infty} \leq 0.9$ . But as  $n$  becomes larger *beyond the point where most of the opportunistic capacity has already been harvested*, the sub-optimality of OPS over SRPT starts to emerge and the ratio quickly deteriorates.

To summarize: *the presence of opportunistic capacity gains mitigates the sub-optimality of OPS, however, if load increases to a point where opportunistic capacity function saturates, OPS can become significantly sub-optimal.*

This observation naturally leads us to investigate a threshold-based or regulated-admission scheduler described in the next section – for all sufficiently large thresholds, the scheduler’s competitive ratio is better than that of OPS.

### 5.5.5 SRPT-OPS scheduler for capacity regions $\mathcal{C}$ and $\bar{\mathcal{C}}$

The SRPT-OPS( $n^*$ ) scheduler admits to service at most a fixed number  $n^*$  of users according to SRPT, and serves the admitted users according to OPS. The details are as follows.

**Definition 5.3** *Consider a system with a concave increasing opportunistic capacity function ( $g_k, k \geq 0$ ) and the associated capacity regions ( $\mathcal{C}_k, k \geq 0$ ) or ( $\bar{\mathcal{C}}_k, k \geq 0$ ). Set a threshold  $n^* > 0$ . At any time  $t$  when there are  $|q(t)| = n$  files in the system, SRPT-OPS( $n^*$ ) serves each of the shortest  $\min(n, n^*)$  files at rate  $\frac{g_{\min(n, n^*)}}{\min(n, n^*)}$ , while the remaining  $(n - n^*)^+$  files wait in a queue.*

**Lemma 5.3** *For any concave increasing capacity function ( $g_k, k \geq 0$ ), and initial state ( $q(0); (l_k(0), 1 \leq k \leq |q(0)|)$ ), the cost under SRPT-OPS( $n^*$ ),  $c_{sp}(\cdot)$ , with threshold parameter  $n^* \geq 1$  is given by,*

$$c_{sp}(\mathbf{l}(0); |q(0)|) = \sum_{k=1}^{|q(0)|} \hat{\pi}_k l_{(k)}(0) , \quad (5.12)$$

where, for any  $k > 0$ ,

$$\hat{\pi}_k \equiv \pi_{[k]} + \left\lfloor \frac{k-1}{n^*} \right\rfloor \frac{n^*}{g_{n^*}}, \quad (5.13)$$

where  $[k] \equiv k \pmod{n^*}$  (with  $\pi_0 \equiv \pi_{n^*}$ , see (5.9)) and  $\left\lfloor \frac{k-1}{n^*} \right\rfloor$  is the integer part of  $\frac{k-1}{n^*}$ .

In particular, we note that for multiples for  $n^*$ , we have that,

$$\hat{\pi}_{kn^*} = \pi_{n^*} + \frac{(k-1)n^*}{g_{n^*}},$$

which is affine in  $k$  with slope  $\frac{n^*}{g_{n^*}}$ . From (5.10), we have that for  $g_k = \frac{1-a^k}{1-a}$ , the coefficient  $\theta_{kn^*}$  is affine in  $k$  with slope  $\frac{n^*}{g_\infty}$ . Therefore, the ratio  $\frac{\hat{\pi}_{kn^*}}{\theta_{kn^*}}$  is monotone in  $k$  and  $\sup_{k \geq 1} \frac{\hat{\pi}_{kn^*}}{\theta_{kn^*}} = \max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right)$ . This leads to the following result.

**Theorem 5.3** *For some fixed  $a \in [0, 1)$ , let  $g_k = \frac{1-a^k}{1-a}$  for all  $k \geq 0$ . Let  $c_s(\mathbf{l}; n)$  and  $c_{sp}(\mathbf{l}; n)$  be the costs under SRPT-HPR and SRPT-OPS( $n^*$ ) respectively when starting with  $n$  files of sizes given in  $\mathbf{l} \equiv (l_1, \dots, l_n)$ . Then we have that,*

$$\sup_{n \geq 1, \mathbf{l} \in (0, \infty)^n} \frac{c_{sp}(\mathbf{l}; n)}{c_s(\mathbf{l}; n)} \leq \max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right). \quad (5.14)$$

Moreover, for any fixed  $\epsilon > 0$ ,

$$\lim_{n \rightarrow \infty} \sup_{\mathbf{l} \in (\epsilon, \infty)^n} \frac{c_{sp}(\mathbf{l}; n)}{c_s(\mathbf{l}; n)} = \frac{g_\infty}{g_{n^*}}. \quad (5.15)$$

See Fig. 5.2 (right) and recall the discussion from Section 5.5.4: we noted that for  $n$  such that  $\frac{g_n}{g_\infty} \lesssim 0.9$  (or  $\frac{g_\infty}{g_n} \gtrsim 1.1$ ), the competitive ratio of OPS (and implicitly the ratio  $\frac{\pi_n}{\theta_n}$ ) was fairly small, *i.e.*, under 1.25.

Therefore, in light of the bound “ $\max\left(\frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}}\right)$ ” given in (5.14), if the admission threshold  $n^*$  is set at a point where most of the opportunistic capacity, say 90%, is useable under SRPT-OPS( $n^*$ ), then SRPT-OPS( $n^*$ ) will have fairly small competitive ratio. But unlike SRPT-HPR which allocates service from a fictitious larger capacity region  $\bar{\mathcal{C}}_n$ , the SRPT-OPS scheduler allocates service from the actual region  $\mathcal{C}_n$  and is therefore feasible in a real system.

In fact, since  $\frac{\pi_k}{\theta_k} \leq 2$ , for any  $n^*$  such that  $\frac{g_\infty}{g_{n^*}} \leq 2$ , *i.e.*, at least half the opportunistic capacity is useable under SRPT-OPS( $n^*$ ), we have that the competitive ratio of SRPT-OPS( $n^*$ ) is always better than that of OPS. Moreover, by (5.15), as  $n \rightarrow \infty$ , the competitive ratio of SRPT-OPS becomes  $\frac{g_\infty}{g_{n^*}}$ .

In light of the bound given in (5.14), Fig. 5.2 (right) captures the tradeoff mentioned in Section 5.1, and gives the cost of compromising opportunistic gain,  $\frac{g_\infty}{g_{n^*}}$ , versus that of ignoring file-size information,  $\frac{\pi_{n^*}}{\theta_{n^*}}$ .

In the next sections we discuss simulation results for a dynamic system.

## 5.6 Dynamic system in steady state

In this section, we compare through simulations, the *mean* sojourn time under OPS, SRPT-OPS, and SRPT-HPR for a dynamic system in steady state, and under various fading scenarios. For a system that implements admission control, we also compare the probability of blocking. We will also comment on the validity of modeling the dynamic system with time varying channels – introduced in Section 5.2.2 – as an instance of  $M/GI/\mathbf{C}$  queue; recall that an  $M/GI/\mathbf{C}$  queue is obtained from the dynamic system with time varying channels by using a separation-of-time-scales argument.

We begin by presenting “simple” simulation results for the  $M/GI/\bar{\mathbf{C}}$  queue to highlight the key insights, followed by the “detailed” simulations of the dynamic system with time varying channels.

We proceed by simulating the  $M/GI/\bar{\mathbf{C}}$  OPS, SRPT-HPR, and SRPT-OPS( $n^*$ ) queues for lognormal file-size distribution with mean and variance of 1 and 1.7 respectively, and various arrival rates. Recall that  $M/GI/\bar{\mathbf{C}}$  OPS and  $M/GI/\mathbf{C}$  OPS are indeed identical, and so are  $M/GI/\bar{\mathbf{C}}$  SRPT-OPS( $n^*$ ) and  $M/GI/\mathbf{C}$  SRPT-OPS( $n^*$ ). Moreover, for the  $M/GI/\bar{\mathbf{C}}$  OPS queue, various quantities of interest including the

mean sojourn time can also be analytically computed, see, *e.g.*, [22, Proposition 3.1].

Fig. 5.3 shows the simulation results for polymatroid capacity regions  $\bar{\mathcal{C}}^{(1)}$  and  $\bar{\mathcal{C}}^{(2)}$  corresponding to two unbounded capacity functions  $\mathbf{g}^{(1)} \equiv (g_k^{(1)}, k \geq 0)$  and  $\mathbf{g}^{(2)} \equiv (g_k^{(2)}, k \geq 0)$ , given by

$$\begin{aligned} g_k^{(1)} &= \log_2(1 + k), \\ g_k^{(2)} &= \log_2(1 + \log_2(1 + k)). \end{aligned}$$

Fig. 5.4 shows the simulation results for capacity regions  $\bar{\mathcal{C}}^{(3)}$  corresponding to a bounded capacity function  $\mathbf{g}^{(3)} \equiv (g_k^{(3)}, k \geq 0)$ , given by

$$g_k^{(3)} = 2 \left( 1 - \left( \frac{1}{2} \right)^k \right).$$

We make the following observations.

- For the two unbounded capacity functions, as the load increases, the ratio of *mean* sojourn time under OPS to that under SRPT-HPR stays around 1.05 and 1.2 respectively for  $\mathbf{g}^{(1)}$  and  $\mathbf{g}^{(2)}$ .
- For the bounded capacity function, as the load increases, the ratio of *mean* sojourn time under OPS to that under SRPT-HPR increases sharply from 1.2 to 2. Whereas, the ratio of *mean* sojourn time under SRPT-OPS(5) to that under SRPT-HPR stays under 1.25. Also, see Fig. 5.5 for a plot of the mean sojourn time versus the threshold  $n^*$  for other values of threshold besides 5, indicating that the performance of SRPT-OPS is better than OPS for a wide range of thresholds.

After a caveat, we state our conclusions from these simulation results. As per Theorem 5.1, SRPT-HPR is optimal only for the *transient* system with polymatroid capacity regions  $\bar{\mathcal{C}}$ . The optimal scheduler for the dynamic system  $M/GI/\bar{\mathcal{C}}$  is not

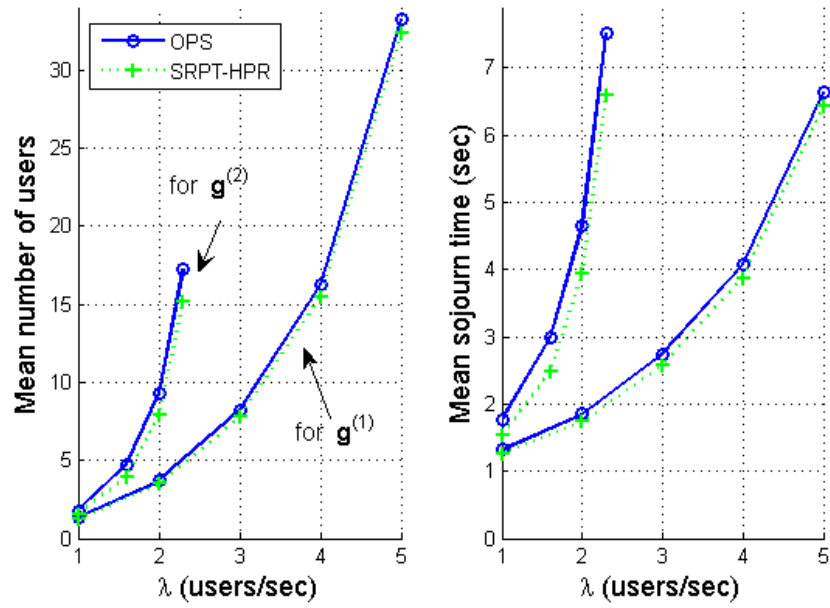


Figure 5.3: Mean number of users in the system and mean sojourn time for unbounded capacity functions  $g^{(1)}$  and  $g^{(2)}$ .

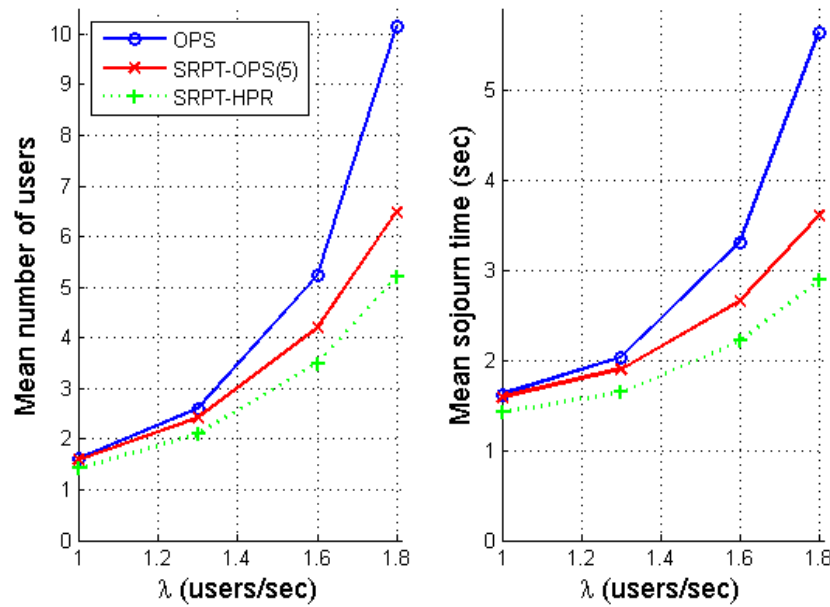


Figure 5.4: Mean number of users in the system and mean sojourn time for bounded capacity function  $g^{(3)}$ .

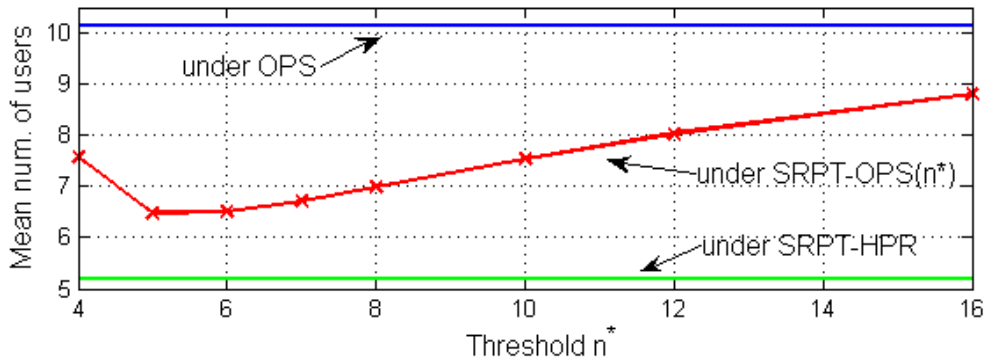


Figure 5.5: Mean sojourn times under SRPT-OPS( $n^*$ ), OPS, and SRPT-HPR for capacity function  $g^{(3)}$  and user arrival rate  $\lambda = 1.8$  user/sec.

known, however, one can expect SRPT-HPR to be close to optimal. Therefore, formally speaking, the mean sojourn time under SRPT-HPR cannot be claimed as the minimum achievable mean sojourn time in  $M/GI/\bar{\mathcal{C}}$  system, or a lower bound for achievable mean sojourn time in the  $M/GI/\mathcal{C}$  system. Nevertheless, subsequently we will compare the performance of various schedulers against that of SRPT-HPR, as done for the transient system.

Now we are ready to state our conclusions from the results in Figs. 5.3–5.5. We conclude that for servers/channels with *high* marginal opportunistic gain (*e.g.*,  $\log^2(n)$ ), a file-size aware scheduler will not offer a significant reduction in mean sojourn time over that of the file-size oblivious OPS (or SRPT-OPS with large threshold). Therefore, the only case of interest which offers room for improvement is where the channels exhibit only moderate marginal opportunistic gains (*e.g.*,  $1 - a^n$ ). In this case, SRPT-OPS *with an appropriate threshold* can offer significant improvement over OPS, *e.g.*, for the shown results, SRPT-OPS is not more than 25% worse than SRPT-HPR.

Next, we present a detailed simulation of a dynamic system with time varying channels, as well as the corresponding  $M/GI/\bar{\mathcal{C}}$  system.

### 5.6.1 Simulation of a dynamic wireless system with time varying channels and the corresponding $M/GI/\bar{\mathcal{C}}$ queue

We consider a slotted time system with a slot duration  $t_s = 1$  ms, where only a single user can be scheduled over a time slot.

**Model for time varying channels:** The time varying wireless channel of user  $i$ , denoted by  $(R_i(t), t \in \mathbb{Z})$ , has the same distribution as that of random  $R$ , which in turn is given by,

$$R = 10^6 \times \log_2(1 + H) \text{ bps}, \quad (5.16)$$

where  $H$  denotes the random SNR. If user  $i$  is scheduled over the time slot  $[t, t + 1)$  (of duration 1 ms), then  $R_i(t)t_s$  bits can be served from its queue. We will consider the following two scenarios for the distribution of SNR  $H$ .

- (A) *Rayleigh fading:*  $\sqrt{H}$  has a Rayleigh distribution with a mean of 1 dB.
- (B) *Rician fading with quantization:*  $H$  is a quantized version of  $\hat{H}$ , where  $\sqrt{\hat{H}}$  has a Rician distribution with a mean of 1 dB and  $K$  factor of 10, and the quantization levels are as prescribed for the modulation and coding schemes available in an HDR system; see [32, Table 2].

Recall the definition of opportunistic capacity function from (5.3), and let  $\left(g_k^{(A)}, k \geq 0\right)$  and  $\left(g_k^{(B)}, k \geq 0\right)$  denote the capacity functions corresponding to the fading scenarios (A) and (B) respectively. See Fig. 5.6 for a plot of the normalized capacity functions, *i.e.*,  $\left(\frac{g_k^{(A)}}{g_1^{(A)}}, k \geq 0\right)$  and  $\left(\frac{g_k^{(B)}}{g_1^{(B)}}, k \geq 0\right)$  where  $g_1^{(A)} \approx 10^6$  and  $g_1^{(B)} \approx 1.06 \times 10^6$ . We note that the normalized capacity function associated with Rayleigh fading matches  $\left(g_k^{(2)}, k \geq 0\right)$ , and the normalized capacity function associated with quantized Rician fading matches  $\left(\frac{1-a^k}{1-a}, k \geq 0\right)$  for  $a = 0.33$ . For any  $n > 0$ , let  $\bar{\mathcal{C}}_n^{(A)}$  denote the polymatroid capacity region defined by  $\left(g_k^{(A)}, k \geq 0\right)$  and let  $\bar{\mathcal{C}}^{(A)} \equiv \left(\bar{\mathcal{C}}_n^{(A)}, n > 0\right)$ . The regions  $\bar{\mathcal{C}}^{(B)}$  are defined similarly.

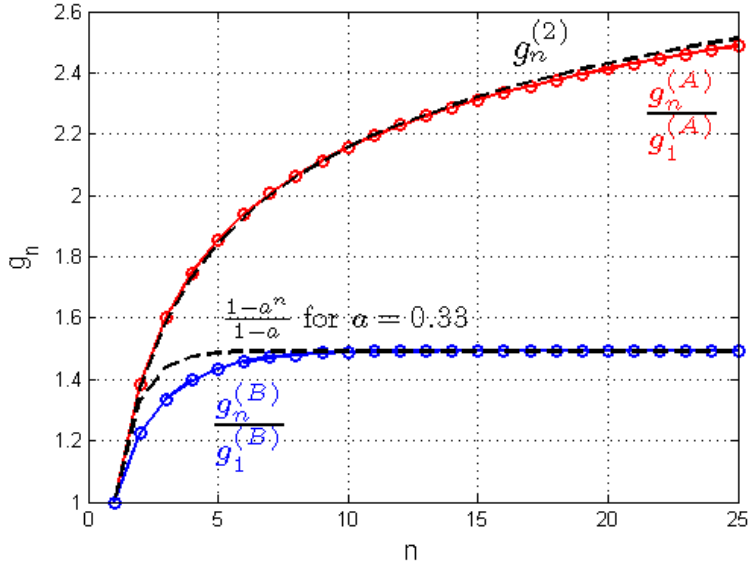


Figure 5.6: Normalized capacity functions associated with various fading scenarios, and their approximate matches.

**Model for user/file arrivals:** New users/files arrive in the beginning of a time slot as a Bernoulli process of rate  $\lambda$ ; simulated values of  $\lambda$  will be specified later. File sizes are lognormally distributed with mean  $\beta = 10^6$  bits (122 KB) and variance  $1.72\beta^2$ . We have that  $\frac{\beta}{g_1^{(\cdot)}}$  is approximately 1 second.

**Model for admission control:** Most systems handling best-effort traffic implement admission control due to constraints on memory or computational resource, and/or to manage overheads and user-perceived performance [59, 60]. We model admission control by limiting the system to at most 25 users, and any users that arrive when there are already 25 users in the system are blocked and rejected.

**Schedulers:** In this slotted time system with time varying channels, the OPS scheduler corresponds to scheduling in each time slot a file/user with the best channel, with ties broken uniformly at random. Similarly, SRPT-OPS( $n^*$ ) corresponds to scheduling according to OPS from amongst (at most)  $n^*$  files with the shorter residual sizes.



**Simulated system loads:** We simulate the dynamic system with time varying channels under fading scenario (A) (*DSwTC-A* for short) and the corresponding  $M/GI/\bar{\mathcal{C}}^{(A)}$  system for the following system loads:

$$\lambda\beta \in \left\{ g_1^{(A)}, g_2^{(A)}, g_5^{(A)}, g_{10}^{(A)}, g_{15}^{(A)} \right\}, \quad (5.17)$$

see Fig. 5.6 for reference. We simulate *DSwTC-B* and the corresponding  $M/GI/\bar{\mathcal{C}}^{(B)}$  system for the following system loads:

$$\lambda\beta \in \left\{ g_1^{(B)}, g_2^{(B)}, g_3^{(B)}, g_5^{(B)} \right\}. \quad (5.18)$$

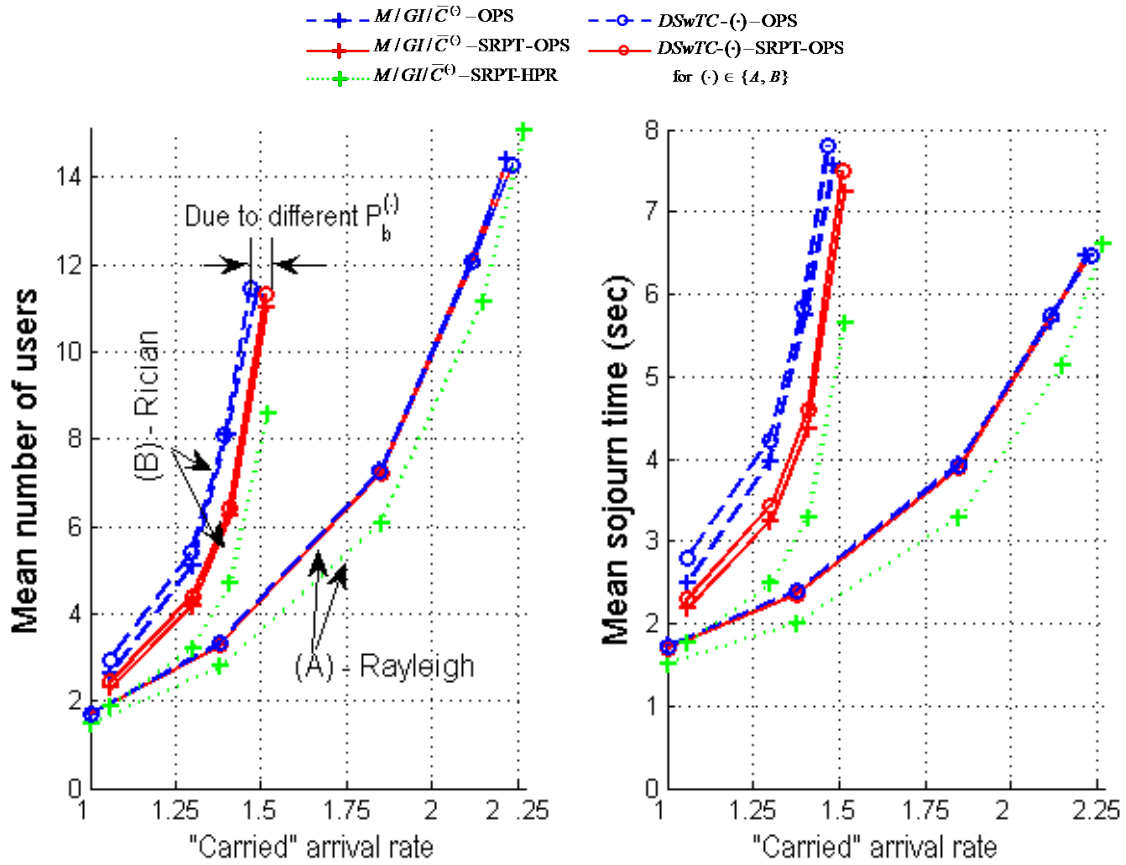


Figure 5.7: Mean number of users in the system and mean sojourn time under various schedulers, plotted versus *carried* load.

Fig. 5.7 gives the plots for mean sojourn time and mean number of users in the system under the following schedulers:

- $DSwTC-A$  and  $DSwTC-B$  under schedulers OPS and SRPT-OPS( $n^*$ ) with various load-dependent thresholds  $n^*$  (specified later).
- $M/GI/\bar{\mathcal{C}}^{(A)}$  and  $M/GI/\bar{\mathcal{C}}^{(B)}$  under schedulers OPS, SRPT-HPR, and SRPT-OPS( $n^*$ ).

Note that the  $x$ -axis in the plots is the *carried* rate  $(1 - P_b^{(\cdot)})\lambda$  (instead of the *offered* rate  $\lambda$ ), where  $P_b^{(\cdot)}$  for  $(\cdot) \in \{\text{OPS, SRPT-HPR, SRPT-OPS}(n^*)\}$  denotes the probability that an arriving user is blocked under scheduler  $(\cdot)$ . For a fixed arrival rate  $\lambda$ , the probability of blocking and thus the carried rate is different for different schedulers. We make the following observations.

### Validity of modeling $DSwTC-(\cdot)$ as queue $M/GI/\bar{\mathcal{C}}^{(\cdot)}$

We note that the simulation results for the  $M/GI/\bar{\mathcal{C}}^{(\cdot)}$  OPS system are nearly identical to those for the corresponding  $DSwTC-(\cdot)$  OPS system.<sup>4</sup> The same also holds for SRPT-OPS. This further validates the reduction of a wireless system with time varying channels to an instance of the queue  $M/GI/\mathcal{C}$  for appropriately defined capacity regions  $\mathcal{C}$ . The latter system offers certain analytical simplifications which were discussed in Remark 5.3.

### Opportunistic gain mitigates sub-optimality of OPS

For the fading scenario (A) (Rayleigh fading), we note that the sub-optimality of OPS versus SRPT-HPR – measured from the “vertical” separation between the curves corresponding to OPS and SRPT-HPR – is under 20%. Whereas, for the fading scenario (B), the sub-optimality of OPS versus SRPT-HPR is 40% to 75%, depending on the load. We conclude that for a system where the opportunistic capacity increases

---

<sup>4</sup>This is also shown in [22] for the case of Rayleigh fading.

rapidly with the number of users, no significant reduction in mean sojourn time is achievable over the file-size-oblivious and purely-opportunistic OPS scheduler.

We would like to add that in a heterogeneous wireless system where the assumption of identically distributed channels (modulo scaling) is generally not true, OPS is perhaps better implemented as MaxQuantile scheduler rather than Proportional Fair. Moreover, [31] shows through extensive simulations that the performance under MaxQuantile is better than that under Proportional Fair.

### **SRPT-OPS( $n^*$ ) reduces sojourn time as well as blocking probability**

For fading scenario (A), we observed that under SRPT-OPS( $n^*$ ) for any threshold  $n^*$  such that  $\frac{\lambda\beta}{g_{n^*}} \geq 0.85$ , the performance of SRPT-OPS was marginally better or identical to that of OPS. The results plotted in Fig. 5.7 are for the following *load-dependent* thresholds: for the five system loads given in (5.17), the threshold  $n^*$  is set to 5, 5, 12, 20 and 25 respectively. For these thresholds, the means under SRPT-OPS are consistently about 1–3% lower than those under OPS, which is a negligible improvement.

For fading scenario (B), the results plotted in Fig. 5.7 are for the following *load-dependent* thresholds: for the four system loads given in (5.18), the threshold  $n^*$  is set to 3, 5, 5 and 8 respectively. The means under SRPT-OPS are roughly half-way between those under OPS and SRPT-HPR, and about 10% to 30% worse than those under SRPT-HPR. Moreover, we note that the probability of blocking is about ten times lower under SRPT-OPS than that under OPS. For example, when  $\lambda\beta = g_5^{(B)}$ , about 2.3% users are blocked under OPS versus 0.23% under SRPT-OPS(8).

Lastly, for the *DSwTC-B* system with an offered rate of  $\lambda = 1.41$  files/sec (or  $\lambda\beta = g_3^{(B)}$ ), Fig. 5.8 gives the plots of mean sojourn time under OPS and SRPT-OPS( $n^*$ ) for various values of threshold  $n^*$ , indicating that for a wide range of thresh-

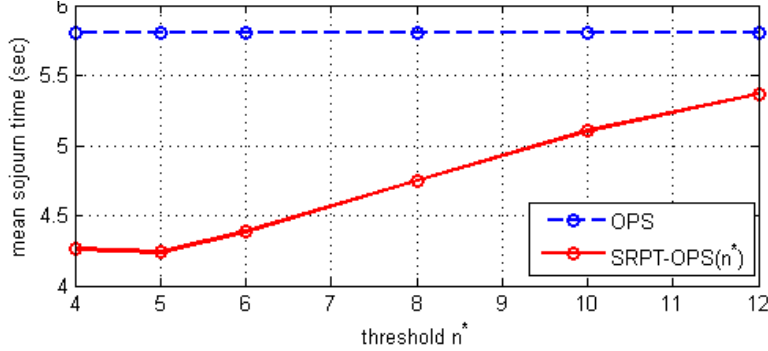


Figure 5.8: Mean sojourn time for  $DSwTC-B$  under OPS and SRPT-OPS( $n^*$ ).

olds, the performance under SRPT-OPS is better than that under OPS. For a fair comparison, the following method was used to maintain the same probability of blocking under all schedulers: whenever an arriving user is blocked in any one system, it is also blocked from all other systems even if those systems had fewer than 25 users at that time. The carried rate thus achieved is about 1.39 files/sec.

## 5.7 Extensions and comments

Consider, for example, the queue  $M/GI/\mathcal{C}^{(A)}$  without any admission control/blocking of users. Then for this queue, the SRPT-OPS( $n^*$ ) scheduler for any *fixed* threshold is not throughput-optimal. In particular, if the load  $\lambda\beta > g_{n^*}^{(A)}$ , then the queue will be unstable under SRPT-OPS( $n^*$ ). While throughput-optimality of a scheduler may not be of much relevance in a wireless system like the one considered in this work, we nevertheless can ensure throughput-optimality of SRPT-OPS by introducing the following modification. Instead of a fixed threshold  $n^*$ , one can make the threshold a function of the number of users in the system. For example, when there are  $n$  users in the system, at most  $n^*(n) \equiv 7 + \frac{n}{4}$  users (with the shorter residual file sizes) are admitted for service, and the remaining are made to wait in a queue. That is, we need to have  $n^*(n) \uparrow \infty$  with  $n$ . Moreover, along the lines of the throughput-optimality result in the previous chapter, one can show that for any

file-size distribution with finite support, the modified SRPT-OPS scheduler will be throughput-optimal also for the corresponding  $DSwTC$ .

## 5.8 Conclusion

We presented new results and insights regarding the key tradeoff involved in scheduling best effort flows over time varying channels. Just as importantly, we reduced the complex scheduling problem to a simpler system, namely queue  $M/GI/\mathcal{C}$ , which explicitly captures the tradeoff and other salient features of the original scheduling problem. For a given system load (number of users in the system), if the opportunistic capacity does not saturate, then the sub-optimality of file-size oblivious OPS (or SRPT-OPS with large threshold) is only minimal. If however the opportunistic capacity saturates at the given load, then OPS can be significantly sub-optimal, whereas, SRPT-OPS, which makes use of file-size information, can still offer near optimal performance. SRPT-OPS is simple to implement, reduces the amount of channel state feedback from users to the base station, and offers service at a nearly constant rate to the *admitted* users.

## Appendix

### Proof of Lemma 5.1

For  $n = 1$  file of size  $l_1$ , we trivially have  $c_s((l_1); 1) = \frac{l_1}{g_1}$ . For  $n = 2$ , let  $l_1 \geq l_2 > 0$  be the initial sizes of the two files. SRPT-HPR serves according to the rate vector  $(g_2 - g_1, g_1)$  until file 2 (the smaller file) departs, leaving behind file 1 with residual size  $l_1 - (g_2 - g_1)\frac{l_2}{g_1}$ . We have,

$$c_s((l_1, l_2); 2) = 2\frac{l_2}{g_1} + \frac{1}{g_1} \left( l_1 - (g_2 - g_1)\frac{l_2}{g_1} \right)$$

Re-arranging the above, we get,

$$\begin{aligned}
c_s((l_1, l_2); 2) &= \frac{l_2}{g_1} \left( 3 - \frac{g_2}{g_1} \right) + \frac{l_1}{g_1}, \\
&= \frac{l_2}{g_1} \left( 2 - \left( \frac{g_2}{g_1} - \frac{g_1}{g_1} \right) \right) + \frac{l_1}{g_1}, \\
&\equiv \theta_2 l_2 + \theta_1 l_1 g_1,
\end{aligned} \tag{5.19}$$

where coefficients  $\theta_k$  are given as follows. Set  $\theta_0 = 0$  and for any  $n > 0$ , let

$$\theta_n \equiv \frac{1}{g_1} \left( n - \sum_{l=1}^{n-1} \theta_l (g_{n+1-l} - g_{n-l}) \right), \tag{5.20}$$

which is equivalent (5.7) with  $g_0 \equiv 0$ . So, for 2 files, the cost under SRPT is given by (5.19). Suppose for some  $n \geq 2$  files  $l_1 \geq \dots \geq l_n$ , the cost under SRPT-HPR is given by

$$c_s((l_1, \dots, l_n); n) = \sum_{k=1}^n \theta_k l_k,$$

then we will show that the cost under SRPT-HPR for  $n + 1$  files  $l_1 \geq \dots \geq l_{n+1}$  is given by  $\frac{1}{g_1} \sum_{k=1}^{n+1} \theta_k l_k$ . We proceed with the induction.

With  $n + 1$  files, the SRPT-HPR scheduler will allocate rates

$$(g_{n+1} - g_n) \leq (g_n - g_{n-1}) \leq \dots \leq (g_2 - g_1) \leq g_1$$

to files with original sizes  $l_1 \geq l_2 \geq \dots \geq l_{n+1}$  respectively. The file  $n + 1$  will be the first to finish (after taking  $l_{n+1}/g_1$  units of time) and leave behind  $n$  files of size,

$$l_1 - (g_{n+1} - g_n) \frac{l_{n+1}}{g_1} \geq l_2 - (g_n - g_{n-1}) \frac{l_{n+1}}{g_1} \geq \dots \geq l_n - (g_2 - g_1) \frac{l_{n+1}}{g_1}.$$

Therefore, the cost under SRPT-HPR is,

$$\begin{aligned}
& c_s((l_1, \dots, l_{n+1}); n+1) \\
&= (n+1) \frac{l_{n+1}}{g_1} + \\
&\quad \sum_{k=1}^n \theta_k \left( l_k - (g_{n+2-k} - g_{n+1-k}) \frac{l_{n+1}}{g_1} \right), \\
&= \sum_{k=1}^n \theta_k l_k + \\
&\quad \frac{l_{n+1}}{g_1} \left( n+1 - \sum_{k=1}^n \theta_k (g_{n+2-k} - g_{n+1-k}) \right), \\
&= \sum_{k=1}^{n+1} \theta_k l_k.
\end{aligned}$$

■

### Proof of Theorem 5.1

To prove Theorem 5.1, first we will need to show that  $\theta_k$  given by (5.7) is increasing in  $k$ . We proceed as follows.

**$\theta_k$  is increasing in  $k$**  It is clear that  $\theta_1 = 1 > \theta_0 = 0$ . Suppose for some  $n \geq 1$ , we have that  $\theta_k - \theta_{k-1} \geq 0$  for all  $k \leq n$ . Then we will show that  $\theta_{n+1} - \theta_n \geq 0$ . By (5.7), we have,

$$\begin{aligned}
1 &= (\boldsymbol{\theta} * \boldsymbol{\Delta})(n) - (\boldsymbol{\theta} * \boldsymbol{\Delta})(n-1), \\
&= \sum_{k=0}^{n-1} \Delta_k (\theta_{n-k} - \theta_{n-1-k}),
\end{aligned} \tag{5.21}$$

and,

$$\begin{aligned}
1 &= (\boldsymbol{\theta} * \boldsymbol{\Delta})(n+1) - (\boldsymbol{\theta} * \boldsymbol{\Delta})(n), \\
&= \sum_{k=0}^n \Delta_k (\theta_{n+1-k} - \theta_{n-k}).
\end{aligned} \tag{5.22}$$

Substituting  $l = k - 1$  in (5.22) and rearranging,

$$\begin{aligned}
(\theta_{n+1} - \theta_n)\Delta_0 &= 1 - \sum_{l=0}^{n-1} \Delta_{l+1} (\theta_{n-l} - \theta_{n-1-l}), \\
&\geq 1 - \sum_{l=0}^{n-1} \Delta_l (\theta_{n-l} - \theta_{n-1-l}), \\
&= 0
\end{aligned}$$

where the inequality above follows from the induction assumption that  $\theta_{k+1} - \theta_k \geq 0$  for  $0 \leq k < n$  and the fact that  $\Delta_l \geq \Delta_{l+1}$ , and the last equality from (5.21).

Now we can continue with the proof of the optimality of SRPT-HPR (Theorem 5.1).

**Proof of Theorem 5.1 (cont.)** Due to convexity of  $\bar{\mathcal{C}}_n$  for all  $n > 0$ , it suffices to consider schedulers  $\psi(s(t))$  that do *not* vary over an epoch. That is, given  $|q(t)| = k > 0$  over some interval  $[t_1, t_2]$ , we only need consider schedulers under which the service rate allocation  $\psi(s(t)) \in \bar{\mathcal{C}}_k$  is constant over  $[t_1, t_2]$ .

Let  $l_1 \geq l_2 \geq \dots \geq l_n$  be initial sizes of the  $n$  given files. For  $n = 1$ , the minimum cost  $c^*$  is simply,

$$c^*((l_1); 1) = \frac{l_1}{g_1} = c_s((l_1); 1).$$

Next we consider the case for  $n = 2$ . Consider a schedule that picks rate vector  $\psi(s(0)) \equiv (r_1, r_2) \in \bar{\mathcal{C}}_2$  – *i.e.*, serves files 1 and 2 at rates  $r_1$  and  $r_2$  respectively – until the first departure, then serves the remaining file (if any) at optimal rate  $g_1$ . Let  $(i, j) = (1, 2)$  if file 1 completes service before (or at the same time) as file 2, otherwise  $(i, j) = (2, 1)$ , *i.e.*,

$$(i, j) = \begin{cases} (1, 2) & \text{if } \frac{l_1}{r_1} \leq \frac{l_2}{r_2}, \\ (2, 1) & \text{otherwise.} \end{cases}$$



Therefore we have  $\frac{l_i}{r_i} \leq \frac{l_j}{r_j}$  and  $r_i \neq 0$ . Then we have,

$$\begin{aligned} c_\psi((l_1, l_2); 2) &= 2\frac{l_i}{r_i} + \frac{1}{g_1} \left( l_j - r_j \frac{l_i}{r_i} \right), \\ &\geq 2\frac{l_i}{r_i} + \frac{1}{g_1} \left( l_j - (g_2 - r_i) \frac{l_i}{r_i} \right), \end{aligned}$$

where the inequality is obtained by noting  $r_j \leq g_2 - r_i$ . Re-arranging above, we get,

$$\begin{aligned} c_\psi((l_1, l_2); 2) &\geq \left( 2 - \frac{g_2}{g_1} \right) \frac{l_i}{r_i} + \frac{l_i}{g_1} + \frac{l_j}{g_1}, \\ &\stackrel{(i)}{\geq} \left( 2 - \frac{g_2}{g_1} \right) \frac{l_i}{g_1} + \frac{l_i}{g_1} + \frac{l_j}{g_1}, \\ &= \left( 3 - \frac{g_2}{g_1} \right) \frac{l_i}{g_1} + \frac{l_j}{g_1}, \\ &\stackrel{(ii)}{\geq} \left( 3 - \frac{g_2}{g_1} \right) \frac{l_2}{g_1} + \frac{l_1}{g_1}, \\ &= c_s((l_1, l_2); 2), \end{aligned}$$

where inequality (i) is obtained by noting that  $2 - \frac{g_2}{g_1} \geq 0$  and  $r_i \leq g_1$ , and replacing  $r_i$  with  $g_1$ ; and inequality (ii) by noting that  $l_1 \geq l_2$  and

$$3 - \frac{g_2}{g_1} \geq 1.$$

Since rate vector  $(r_1, r_2) \in \bar{\mathcal{C}}_2$  was arbitrary, it follows that SRPT-HPR is optimal when starting with  $n \leq 2$  files. Next, suppose SRPT-HPR is optimal for  $n \geq 2$  files, *i.e.*

$$c^*((l_1, \dots, l_n); n) = c_s((l_1, \dots, l_n); n),$$

we will show that SRPT-HPR is also optimal when starting with  $n + 1$  files.

We start with  $n + 1$  files  $l_1 \geq \dots \geq l_{n+1}$ . Consider a scheduler  $\psi(\cdot)$  which picks a rate vector  $\mathbf{r} \in \bar{\mathcal{C}}_{n+1}$  until the first departure. Let  $i \in \{1, \dots, n + 1\}$  be the first to finish file under vector  $\mathbf{r}$ , *i.e.*, for all  $k \in \{1, \dots, n + 1\}$ ,

$$\tau_i \equiv \frac{l_i}{r_i} \leq \frac{l_k}{r_k},$$

(and therefore,  $r_i \neq 0$ ). Then at time  $\tau_i$ , each file  $k$  has size  $l_k - r_k \tau_i$ . For  $k \leq n$ , let  $o(k)$  denote the index of the  $k^{\text{th}}$  largest file in the system at time  $\tau_i$  (ties broken arbitrarily), *i.e.*,

$$l_{o(1)} - r_{o(1)} \frac{l_i}{r_i} \geq l_{o(2)} - r_{o(2)} \frac{l_i}{r_i} \geq \cdots \geq l_{o(n)} - r_{o(n)} \frac{l_i}{r_i} \geq 0.$$

Also, for  $k < i$ , let  $i(k) = k$ , and for  $k \geq i$ , let  $i(k) = k + 1$ . Then the cost  $c_\psi(\cdot)$  under the scheduler  $\psi$  satisfies,

$$\begin{aligned} & c_\psi((l_1, \dots, l_{n+1}); n+1) \\ & \stackrel{(i)}{\geq} (n+1) \frac{l_i}{r_i} + c^* \left( l_{o(1)} - r_{o(1)} \frac{l_i}{r_i}, \dots, l_{o(n)} - r_{o(n)} \frac{l_i}{r_i} \right), \\ & = (n+1) \frac{l_i}{r_i} + \sum_{k=1}^n \theta_k \left( l_{o(k)} - r_{o(k)} \frac{l_i}{r_i} \right), \\ & = \sum_{k=1}^n \theta_k l_{o(k)} + \frac{l_i}{r_i} \left( n+1 - \sum_{k=1}^n \theta_k r_{o(k)} \right), \\ & \stackrel{(ii)}{\geq} \sum_{k=1}^n \theta_k l_{i(k)} + \frac{l_i}{r_i} \left( n+1 - \sum_{k=1}^n \theta_k r_{o(k)} \right), \\ & \stackrel{(iii)}{\geq} \sum_{k=1}^n \theta_k l_{i(k)} + \frac{l_i}{r_i} \left( n+1 - \sum_{k=1}^{n-1} \theta_k (g_{n+2-k} - g_{n+1-k}) - \theta_n (g_2 - r_i) \right), \\ & = \sum_{k=1}^n \theta_k l_{i(k)} + \frac{l_i}{r_i} \left( n+1 - \sum_{k=1}^n \theta_k (g_{n+2-k} - g_{n+1-k}) - \theta_n g_1 \right) + \theta_n l_i, \\ & = \sum_{k=1}^n \theta_k l_{i(k)} + \frac{l_i}{r_i} g_1 (\theta_{n+1} - \theta_n) + \theta_n l_i, \\ & \geq \sum_{k=1}^n \theta_k l_{i(k)} + \theta_{k+1} l_i, \\ & \stackrel{(iv)}{\geq} \sum_{k=1}^{n+1} \theta_k l_k, \\ & = c_s((l_1, \dots, l_{n+1}); n+1), \end{aligned}$$

where the inequalities follow respectively from,

(i) replacing the cost over time  $(\tau_i, \infty)$  with the optimal cost;

(ii) noting that  $\theta_k$  is increasing while  $l_{i(k)}$  is decreasing in  $k$ .

(iii) the greedy maximization of linear functions on a polymatroid region [54, Theorem 44.3], and noting that the service vector  $(r_{o(1)}, \dots, r_{o(n)})$  lies in a  $n$ -polymatroid defined by  $(g_{k+1} - r_i, 1 \leq k \leq n)$ .

(iv) noting that  $\theta_k$  is increasing and  $l_k$  is decreasing in  $k$ .

■

### Proof of Lemma 5.2

For  $n = 1$  file with initial size  $l_1$ , we simply have  $c_p((l_1); 1) = \frac{l_1}{g_1}$ . For  $n = 2$  files with initial sizes  $l_1 \geq l_2 > 0$ , we have,

$$\begin{aligned} c_p((l_1, l_2); 2) &= 2 \frac{l_2}{g_2/2} + \frac{1}{g_1} (l_1 - l_2), \\ &= \left( \frac{2^2}{g_2} - \frac{1}{g_1} \right) l_2 + \frac{1}{g_1} l_1, \\ &\equiv \pi_2 l_2 + \pi_1 l_1, \end{aligned}$$

where coefficients  $\pi_k$  are given as follows:  $\pi_1 = \frac{1}{g_1}$ , and for any  $k > 1$ ,

$$\pi_k \equiv \frac{k^2}{g_k} - \frac{(k-1)^2}{g_{k-1}}.$$

It will be useful to note that for any  $n > 0$ , we have  $\sum_{k=1}^n \pi_k = \frac{n^2}{g_n}$ . Now suppose the cost under OPS when starting in state  $(\{1, \dots, n\}; \mathbf{l} \equiv (l_1, \dots, l_n))$  for some  $n \geq 2$  and  $l_1 \geq \dots, l_n$  is given by,

$$c_p(\mathbf{l}; n) = \sum_{k=1}^n \pi_k l_k,$$

then we will show that the cost when starting in state  $(\{1, \dots, n+1\}; \mathbf{l} \equiv (l_1, \dots, l_{n+1}))$  with initial sizes  $l_1 \geq \dots \geq l_{n+1}$  is given by  $\sum_{k=1}^{n+1} \pi_k l_k$ .

When starting with  $n + 1$  files, each file is served at rate  $\frac{g_{n+1}}{n+1}$  until the first

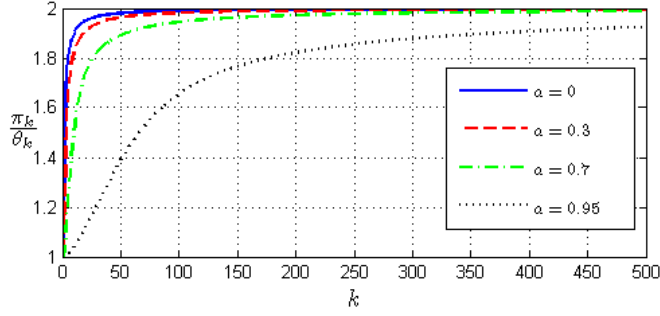


Figure 5.9: Plot of  $\left(\frac{\pi_k}{\theta_k}, k > 0\right)$  for various values of  $a \in [0, 1)$ .

departure, leaving behind  $n$  files of sizes  $l_1 - l_{n+1}, \dots, l_n - l_{n+1}$ . We have,

$$\begin{aligned}
 c_p(\mathbf{l}; n+1) &= (n+1)^2 \frac{l_{n+1}}{g_{n+1}} + \sum_{k=1}^n \pi_k (l_1 - l_{n+1}), \\
 &= \left( \frac{(n+1)^2}{g_{n+1}} - \sum_{k=1}^n \pi_k \right) l_{n+1} + \sum_{k=1}^n \pi_k l_k, \\
 &= \sum_{k=1}^{n+1} \pi_k l_k.
 \end{aligned}$$

■

### Proof of Theorem 5.2

With  $\mathbf{g} = \left(\frac{1-a^k}{1-a}, k \geq 0\right)$  for some fixed  $a \in [0, 1)$ , we have  $\theta_1 = \pi_1 = 1$  and for all  $k > 1$ ,

$$\theta_k = (1-a)k + a \quad \text{and} \quad \pi_k = (1-a) \left( \frac{k^2}{1-a^k} - \frac{(k-1)^2}{1-a^{k-1}} \right).$$

It is easy to show that,

$$\lim_{k \rightarrow \infty} \frac{\pi_k}{\theta_k} = 2. \tag{5.23}$$

We will also need that  $\frac{\pi_k}{\theta_k}$  is increasing in  $k$  for all  $k > 0$ . It can be shown that, for any  $k \geq 0$ , we have

$$\begin{aligned}
 &\frac{g_{k+1}g_{k+2}(k+g_\infty)(k+g_\infty+1)}{g_\infty} \left( \frac{\pi_{k+2}}{\theta_{k+2}} - \frac{\pi_{k+1}}{\theta_{k+1}} \right) \\
 &= (2g_\infty - 1)g_k g_{k+1} + a^k k^2 (k+g_\infty+1)g_{k+2} - \\
 &\quad a^{k+1} \left( (k+1)^2 (k+g_\infty) + (2k+1)(k+g_\infty+1) \right) g_k,
 \end{aligned}$$

where the second and third terms of the summation on the right side of the equality are  $a^k(g_{k+2}k^3 + O(k^2)) - a^{k+1}(g_k k^3 + O(k^2))$ . Therefore, for large  $k$ , we have that  $\frac{\pi_k}{\theta_k}$  is increasing in  $k$ . To show that  $\frac{\pi_k}{\theta_k}$  is increasing in  $k$  for all  $k > 0$ , we resort to numerical computation and refer to Fig. 5.9 which gives the plots of  $\left(\frac{\pi_k}{\theta_k}, k > 0\right)$  for various values of  $a \in [0, 1)$ . Monotonicity of  $\frac{\pi_k}{\theta_k}$  gives the following result,

$$\begin{aligned} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} &= \frac{\sum_{k=1}^n \pi_k l_k}{\sum_{k=1}^n \theta_k l_k}, \\ &= \frac{\pi_n}{\theta_n} \frac{\sum_{k=1}^n \frac{\pi_k l_k}{\pi_n l_n}}{\sum_{k=1}^n \frac{\theta_k l_k}{\theta_n l_n}}. \end{aligned} \quad (5.24)$$

But for any  $k < n$ , since  $\frac{\pi_k}{\pi_n} \leq \frac{\theta_k}{\theta_n}$ , therefore,

$$\frac{\sum_{k=1}^n \frac{\pi_k l_k}{\pi_n l_n}}{\sum_{k=1}^n \frac{\theta_k l_k}{\theta_n l_n}} \leq 1.$$

Using this along with (5.24) and (5.23) we get, for any  $n \geq 1$ ,

$$\frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} \leq \frac{\pi_n}{\theta_n} \leq 2. \quad (5.25)$$

We will also need the following fact which can be verified using simple calculus.

**Fact 5.2** *For any positive constants  $a, b, c, d$ ,*

$$\frac{ax + b}{cx + d}$$

*is increasing in  $x$  if and only if  $\frac{a}{c} \geq \frac{b}{d}$ .*

Recall the assumption  $l_1 \geq \dots \geq l_n$ . For any fixed  $l_1, \dots, l_{n-1}$ , let us examine the ratio  $\frac{c_p(\mathbf{l}; n)}{c_s(\mathbf{l}; n)}$  as a function of  $l_n$  (which can take values in  $[0, l_{n-1}]$ ). Then using Fact 5.2, along with (5.25) (for  $n - 1$ ) and monotonicity of  $\frac{\pi_k}{\theta_k}$ , we have that,  $\frac{c_p(\mathbf{l}; n)}{c_s(\mathbf{l}; n)}$  is maximized at  $l_n = l_{n-1}$ . Proceeding in this way, we can show that,

$$\begin{aligned} \sup_{\mathbf{l} \in \mathbb{R}_+^n} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} &= \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = \frac{\sum_{k=1}^n \pi_k}{\sum_{k=1}^n \theta_k} \\ &= \frac{\frac{n^2}{g_n}}{\frac{1-a}{2}n^2 + \frac{1+a}{2}n} = \frac{2}{(1-a)g_n + (1+a)\frac{g_n}{n}}. \end{aligned} \quad (5.26)$$

Moreover, it can be verified that right side of (5.26),

- monotonically increases to 2 as  $n \uparrow \infty$ . Therefore, for any fixed  $a \in [0, 1)$ ,

$$\sup_{n>0, \mathbf{l} \in \mathbb{R}_+^n} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} = \lim_{n \rightarrow \infty} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = 2. \quad (5.27)$$

- monotonically increases to  $\frac{2}{1+n^{-1}}$  as  $a \downarrow 0$ . That is, for any fixed  $n > 0$ ,

$$\sup_{a \in [0, 1), \mathbf{l} \in \mathbb{R}_+^n} \frac{c_p(\mathbf{l}; n; a)}{c_s(\mathbf{l}; n; a)} = \lim_{a \rightarrow 0} \frac{c_p(\mathbf{1}; n; a)}{c_s(\mathbf{1}; n; a)} = \frac{2}{1+n^{-1}}. \quad (5.28)$$

■

### Proof of Lemma 5.3

The expression for cost under SRPT-OPS,  $c_{sp}(\cdot)$ , *i.e.*,

$$c_{sp}(\mathbf{l}; n) = \sum_{k=1}^n \hat{\pi}_k l_k$$

can be verified using induction, in the same way as that for SRPT-HPR in the proof of Lemma 5.1 and OPS in the proof of Lemma 5.2. ■

### Proof of Theorem 5.3

It can be shown that,

$$\lim_{k \rightarrow \infty} \frac{\hat{\pi}_k}{\theta_k} = \frac{g_\infty}{g_{n^*}}, \quad \text{and} \quad (5.29)$$

$$\sup_{k>0} \frac{\hat{\pi}_k}{\theta_k} = \max \left( \frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}} \right). \quad (5.30)$$

Then (5.15) follows directly from (5.29), whereas (5.14) follows by an application of Fact 5.2 and (5.30), *i.e.*,

$$\sup_{n \geq 1, \mathbf{l} \in (0, \infty)^n} \frac{c_{sp}(\mathbf{l}; n)}{c_s(\mathbf{l}; n)} \leq \sup_{k>0} \frac{\hat{\pi}_k}{\theta_k} = \max \left( \frac{\pi_{n^*}}{\theta_{n^*}}, \frac{g_\infty}{g_{n^*}} \right).$$

■

## Conclusion

To conclude, we review the main results of the thesis and discuss some of the remaining challenges.

First, we review our results on QoS scheduling and in doing so, present an alternative view of the thesis. The complexity in this problem has two aspects to it:

- (1) complexity due to the fact that the opportunistic capacity region (see (2.4)) is typically not a simplex or a polymatroid, but instead a general convex, coordinate convex set;
- (2) complexity due to the fact that the opportunistic capacity region is realizable only on sufficiently long time scales, as an average of the time-varying instantaneous capacity regions.

So, Chapter 2 and Appendix A can be viewed as focusing on the first aspect of complexity and ignoring the second by using a time-scale separation argument. That is, we investigated the control of parallel queues, where the queues can be *served simultaneously at any rate vector from a fixed convex capacity region*. If the capacity region is not a simplex or a polymatroid, then there is a tradeoff between *maximizing total service rate* and *balancing unequal queues*. Numerical computation of the mean delay optimal policies for various capacity regions, as well as an investigation of the corresponding fluid models, led to the discovery of a key property we named radial sum-rate monotonicity (RSM). RSM pertains to the above-mentioned tradeoff, and an RSM policy, such as the Log rule, can be viewed as *perturbing* a greedy policy – greedy with respect to the service rate – just enough to make it throughput optimal.<sup>5</sup>

As mentioned above, both Chapter 2 and Appendix A ignore the second aspect of complexity, *i.e.*, the fact that only the *long-run* average service rate offered to

---

<sup>5</sup>By contrast, Exp rule can be viewed as perturbing the “serve the longest queue” policy just enough to make it throughput optimal.



the contending queues/users lies in the opportunistic capacity region. Nevertheless, a functional law of large numbers suggests that under an appropriate scaling of time and queue state-space, the second source of complexity should vanish and the scaled queues should appear as if they can be served at any rate vector from a fixed capacity region. This observation was formalized in Chapter 3, where we investigated the scheduling problem in all its complexity, and showed that a candidate RSM policy minimizes the asymptotic probability of sum-queue overflow. Moreover, in conjunction with the results of [19] and [47], we note that the mode of overflow under an RSM policy translates to a more graceful degradation of service under transient overloads.

Second, we review our results on best-effort scheduling. The main result of Chapter 4 was motivated by [28], and stated that a MaxWeight-type scheduler which prioritizes flows that are experiencing long delays is throughput optimal. However, the relationship between the *residual* flow size and its *current* delay is unclear, and the performance of delay-driven MaxWeight may typically be worse than a purely opportunistic schedulers. Therefore, in Chapter 5 we explicitly investigated the impact of using residual flow size information for scheduling. We found that in fading scenarios where the opportunistic capacity grows sharply with the number of users – Rayleigh fading for example – the use of residual flow size information does not result in a significant decrease in mean flow transfer delay. In such fading scenarios, purely opportunistic schedulers like MaxQuantile or Proportional Fair perform very well. By contrast, in fading scenarios where the opportunistic capacity grows slowly with the number of users, the use of residual flow size information may offer a significant reduction in mean flow transfer delay. We proposed a threshold-based scheduler that performs well in all fading scenarios, would be “simple” to implement, and exhibits robustness in that its performance is better than purely opportunistic schedulers for all sufficiently large thresholds.

Finally, we comment on some of the remaining challenges. In the first half

of the thesis, as well as in other works [13, 18, 19, 47, 61, 62], the performance improvement offered by the proposed schedulers – which are optimal in their respective settings – over previously known heuristics is often determined through simulations. The analytical sub-optimality bounds for various schedulers largely remain an open problem, except for some limited results given in [8]. It would be interesting to extend the Lyapunov function techniques of [8] to more general systems, or investigate alternative techniques such as those introduced in [63] to obtain performance bounds for some representative, insightful examples.

This leads us to the next open problem: a good measure of robustness for schedulers in wireless systems. In this thesis, we used a somewhat informal notion of robustness, namely, sensitivity of performance to scheduler parameters and graceful degradation of service (which we subsequently related to the mode of queue overflow). The large deviations or asymptotic optimality results in this thesis as well as those in [18, 19, 47, 62] are insensitive to the choice of the respective scheduler’s parameters – this can be viewed as a strength and a weakness of these results, at the same time. It is a strength because the asymptotic optimality holds irrespective of the choice of the scheduler parameters, and a weakness because the performance over the regime of interest – non-asymptotic performance – typically depends significantly on the scheduler parameters.

In the second half of the thesis when investigating the scheduling of best-effort flows, we explicitly focused on obtaining sub-optimality bounds. However, the analytical bounds developed were applicable only to a transient system. For the dynamic system in steady state, a straight-forward lower bound on the achievable mean delay is given in [64], however, this bound perhaps is not sufficiently tight for most cases of interest in our setting. Therefore, good sub-optimality bounds for the dynamic system in steady state still remain an open problem.

## Appendices

# Appendix A

## Fluid-scale asymptotic optimality and RSM policies

In Section 2.4.1, we observed that optimal policies  $\boldsymbol{\mu}^*$  for the MDP, computed through relative value iteration for different capacity regions and arrival vectors, satisfied condition (a) in the definition of RSM (see Definition 2.1). In this appendix, we will consider if and when RSM policies are *fluid-scale asymptotic optimal* (FSAO), which is formally defined in Section A.0.2. Roughly speaking, the asymptotic slopes of the switching curves under  $\boldsymbol{\mu}^*$  and FSAO policies are identical.

For the MDP, we have already defined an optimal policy  $\boldsymbol{\mu}^*$  by (2.12) and (2.13), and a representative RSM policy, namely, the Log rule  $\boldsymbol{\mu}^L$  by (2.26). Paralleling this, next, for a deterministic fluid model, we will introduce an optimal fluid policy  $\boldsymbol{\mu}^{F^*}$  and a greedy fluid policy  $\boldsymbol{\mu}^{Fg}$ . The two policies for the MDP are related to the two policies for the fluid model, as shown in Fig. A.1. Using these relationships, one can show the following.

- (1) If  $\boldsymbol{\mu}^{Fg}$  is not an optimal policy for the fluid model, then  $\boldsymbol{\mu}^*$  does not satisfy condition (b) in the definition of RSM.
- (2) Otherwise, RSM policies like the Log rule are FSAO. Furthermore, if  $\boldsymbol{\mu}^{Fg}$  is the *unique* optimal policy for the fluid model, then it follows that  $\boldsymbol{\mu}^*$  and  $\boldsymbol{\mu}^L$  have similar switching curves on the state-space of appropriately scaled queue process.

The formal description of the above follows. Subsequently, we will also consider examples of fluid models where the greedy policy may or may not be optimal.

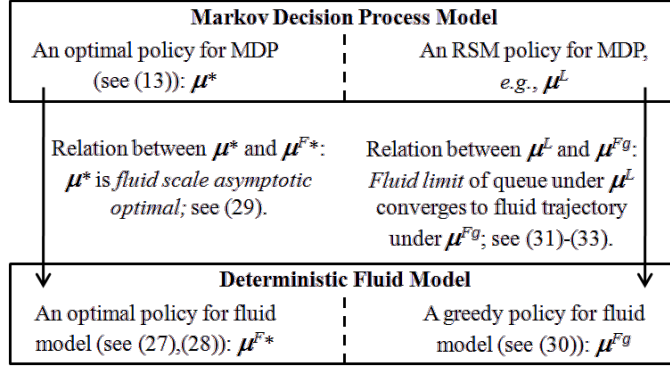


Figure A.1: The Relation between policy  $\boldsymbol{\mu}^*$  for MDP and policy  $\boldsymbol{\mu}^{F*}$  for fluid model, and between policy  $\boldsymbol{\mu}^L$  for MDP and policy  $\boldsymbol{\mu}^{Fg}$  for fluid model.

### A.0.1 Deterministic Fluid Model

One can associate a fluid model (see, *e.g.*, [38, 65]) with the MDP defined in Section 2.3, as follows. Let  $(\boldsymbol{x}(t), t \geq 0)$  be a deterministic continuous trajectory starting at point  $\boldsymbol{x}(0) = \boldsymbol{y} \in \mathbb{R}_+^{\bar{n}}$  and evolving as,

$$\boldsymbol{x}(t) = \boldsymbol{y} + \boldsymbol{\lambda}t - \int_0^t \boldsymbol{u}(\tau) d\tau ,$$

where the *control*  $\boldsymbol{u}(\cdot) \in \mathcal{C}$  and the control policy  $(\boldsymbol{u}(t), t \geq 0)$  is measurable and keeps  $\boldsymbol{x}(\cdot)$  in  $\mathbb{R}_+^{\bar{n}}$ . To indicate the dependence of the trajectory on the control policy and initial condition, subsequently we will write  $\boldsymbol{x}^u(t; \boldsymbol{y})$  instead of  $\boldsymbol{x}(t)$ . The cost of trajectory  $(\boldsymbol{x}^u(t; \boldsymbol{y}), t \geq 0)$  in turn is given by,

$$J_u^F(\boldsymbol{y}) \equiv \int_0^\infty \langle \boldsymbol{w}, \boldsymbol{x}^u(t; \boldsymbol{y}) \rangle dt . \tag{A.1}$$

Since  $\boldsymbol{\lambda}$  lies in the interior of  $\mathcal{C}$  (see (2.7)), there exists at least one control policy  $(\boldsymbol{u}(t), t \geq 0)$  under which the cost is finite for all finite starting points  $\boldsymbol{x}(0)$ .

If  $\boldsymbol{u}(t) = \boldsymbol{\mu}^F(\boldsymbol{x}(t))$ , then we will refer to  $\boldsymbol{\mu}^F : \mathbb{R}_+^{\bar{n}} \rightarrow \mathcal{C}$  as a state-feedback control policy for the fluid model. It is shown in [65] that there exists a state-feedback policy  $\boldsymbol{\mu}^{F*}$  with the following properties:

- For any starting point  $\mathbf{y} \in \mathbb{R}_+^{\bar{n}}$ , the trajectory  $(\mathbf{x}^{\mu^{F^*}}(t; \mathbf{y}), t \geq 0)$  is absolutely continuous (and thus differentiable a.e.) and satisfies for all regular  $t$ ,

$$\frac{d}{dt} \mathbf{x}(t) = \boldsymbol{\lambda} - \boldsymbol{\mu}^{F^*}(\mathbf{x}(t)).$$

- The policy  $\boldsymbol{\mu}^{F^*}$  is optimal for the fluid model. That is, for any admissible policy  $(\mathbf{u}(t), t \geq 0)$ , we have,

$$J_{\boldsymbol{\mu}^{F^*}}^F(\mathbf{y}) = \int_0^\infty \langle \mathbf{w}, \mathbf{x}^{\mu^{F^*}}(t; \mathbf{y}) \rangle dt \leq J_{\mathbf{u}}^F(\mathbf{y}). \quad (\text{A.2})$$

Moreover, *any* state-feedback optimal fluid policy  $\boldsymbol{\mu}^{F^*}$  must satisfy the Hamilton-Jacobi-Bellman equation (see [38, Proposition 4.3.2]), *i.e.*, for all  $\mathbf{x} \in \mathbb{R}_+^{\bar{n}}$ ,

$$\boldsymbol{\mu}^{F^*}(\mathbf{x}) \in \arg \max_{\mathbf{u} \in \mathcal{C}_{\mathbf{x}}} \langle \nabla J_{\boldsymbol{\mu}^{F^*}}^F(\mathbf{x}), \mathbf{u} \rangle, \quad (\text{A.3})$$

where  $\mathcal{C}_{\mathbf{x}} = \{\mathbf{u} \in \mathcal{C} : \forall n \in \mathcal{N}, \text{ if } x_n = 0 \Rightarrow u_n \leq \lambda_n\}$  is the set of admissible controls in state  $\mathbf{x}$ . Subsequently we will use (A.3) to test a candidate fluid policy for optimality.

### A.0.2 Fluid-scale asymptotic optimality

For each integer  $\theta > 0$ , consider an independent Markov chain,  $(\mathbf{Q}_k^{(\theta)}, k \geq 0)$ , starting in state  $\mathbf{q}^{(\theta)}(0) = (\lfloor \theta y_n \rfloor, n \in \mathcal{N}) \in \mathbb{Z}_+^{\bar{n}}$  and evolving under a scheduling policy  $\boldsymbol{\mu}$ . Let  $\bar{\mathbf{Q}}^{(\theta)}(t)$  denote the *fluid-scaled* version of the Markov chain, *i.e.*,

$$\bar{\mathbf{Q}}^{(\theta)}(t) \equiv \frac{1}{\theta} \mathbf{Q}_{\lfloor \theta t \rfloor}^{(\theta)}, \quad t \in \mathbb{R}_+.$$

Then the policy  $\boldsymbol{\mu}$  is said to be FSAO if it satisfies the following [65]:

$$\lim_{t \rightarrow \infty} \lim_{\theta \rightarrow \infty} \mathbb{E}_{\bar{\mathbf{q}}^{(\theta)}(0)}^\mu \left[ \int_0^t \langle \mathbf{w}, \bar{\mathbf{Q}}^{(\theta)}(\tau) \rangle d\tau \right] = J_{\boldsymbol{\mu}^{F^*}}^F(\mathbf{y}). \quad (\text{A.4})$$

### A.0.3 Relation between $\boldsymbol{\mu}^*$ and $\boldsymbol{\mu}^{F^*}$

The optimal policies  $\boldsymbol{\mu}^*$  for the MDP are FSAO; see [38, Theorem 10.0.5 and Section 10.6.2].

#### A.0.4 Relation between RSM Policy for MDP and Greedy Policy for Fluid Model

For ease of exposition, we assume that the capacity region  $\mathcal{C}$  and vectors  $\boldsymbol{\lambda}$  and  $\boldsymbol{w}$  are such that the  $\arg \max_{\boldsymbol{u} \in \mathcal{C}_x} \langle \boldsymbol{w}, \boldsymbol{u} \rangle$  is unique for all  $\boldsymbol{x} \in \mathbb{R}_+^{\bar{n}}$ . Let us define a policy  $\boldsymbol{\mu}^{Fg}$  which is *greedy* with respect to the weight vector  $\boldsymbol{w}$ . More precisely,  $\boldsymbol{\mu}^{Fg}$  is given as follows: for any  $\boldsymbol{x} \in \mathbb{R}_+^{\bar{n}}$ , we have,

$$\boldsymbol{\mu}^{Fg}(\boldsymbol{x}) \equiv \arg \max_{\boldsymbol{u} \in \mathcal{C}_x} \langle \boldsymbol{w}, \boldsymbol{u} \rangle. \quad (\text{A.5})$$

For example, for any  $\boldsymbol{x} > \mathbf{0}$ , the control  $\boldsymbol{\mu}^{Fg}(\boldsymbol{x})$  is equal to the (weighted) max-sum-rate vertex  $\arg \max_{\boldsymbol{u} \in \mathcal{C}} \langle \boldsymbol{w}, \boldsymbol{u} \rangle$ .

Next, we describe the relation between the RSM policy and the greedy policy. For each integer  $\theta > 0$ , consider an independent Markov chain,  $(\boldsymbol{Q}_k^{(\theta)}, k \geq 0)$ , starting in state  $\boldsymbol{q}^{(\theta)}(0) = (\lfloor \theta y_n \rfloor, n \in \mathcal{N}) \in \mathbb{Z}_+^{\bar{n}}$  and evolving under an RSM policy. The analysis in Lemmas 3.1–2.2 and [47] shows that under an RSM policy like the Log rule,  $\boldsymbol{\mu}^L$ , we have the following uniform convergence over compact sets along some subsequence  $\{\theta_j\}$ ,

$$(\bar{\boldsymbol{Q}}^{(\theta_j)}(t), t \geq 0) \rightarrow (\bar{\boldsymbol{q}}(t), t \geq 0),$$

where the *fluid-limit*  $(\bar{\boldsymbol{q}}(t), t \geq 0)$  with  $\bar{\boldsymbol{q}}(0) = \boldsymbol{y}$  satisfies,

$$\frac{d}{dt} \bar{\boldsymbol{q}}(t) = \boldsymbol{\lambda} - \arg \max_{\boldsymbol{u} \in \mathcal{C}_{\bar{\boldsymbol{q}}(t)}} \langle \boldsymbol{w}, \boldsymbol{u} \rangle. \quad (\text{A.6})$$

Subsequently we will write  $(\bar{\boldsymbol{q}}(t; \boldsymbol{y}), t \geq 0)$  to explicitly indicate the starting point of the fluid-limit trajectory. By comparing (A.5) and (A.6), we have that the fluid limit under the RSM policy and the fluid trajectory under the greedy policy are identical, *i.e.*,

$$\boldsymbol{x}^{\boldsymbol{\mu}^{Fg}}(t; \boldsymbol{y}) = \bar{\boldsymbol{q}}(t; \boldsymbol{y}), \quad t \geq 0. \quad (\text{A.7})$$

The following observation captures the intuition: recall the weighted max-rate horn described in Section 2.4.1; for any point of the fluid-limit trajectory in the interior

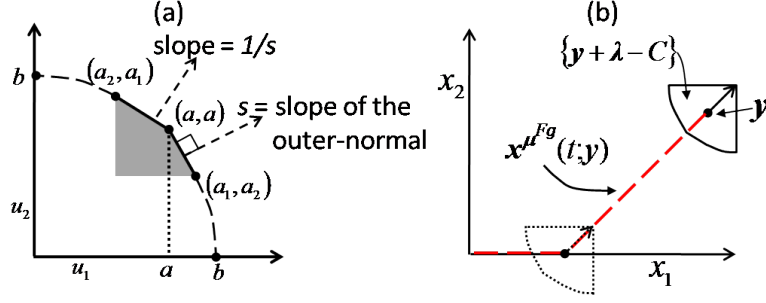


Figure A.2: Fluid model: (a) capacity region  $\mathcal{C}$ , (b) fluid trajectory  $\mathbf{x}^{\mu^{Fg}}(\cdot; \mathbf{y})$  starting at point  $\mathbf{y}$  with  $y_1 > y_2$ .

of  $\mathbb{R}_+^{\bar{n}}$ , *i.e.*,  $\bar{\mathbf{q}}(t) > \mathbf{0}$ , the unscaled queue,  $\theta_j \bar{\mathbf{Q}}^{(\theta_j)}(t)$ , for large enough  $\theta_j$  lies in the weighted max-rate horn.

Lastly, we have the following convergence in the mean,

$$\lim_{t \rightarrow \infty} \lim_{\theta \rightarrow \infty} \mathbb{E}_{\bar{\mathbf{q}}^{(\theta)}(0)}^{\mu^L} \left[ \int_0^t \langle \mathbf{w}, \bar{\mathbf{Q}}^{(\theta)}(\tau) \rangle d\tau \right] = \int_0^\infty \langle \mathbf{w}, \bar{\mathbf{q}}(\tau; \mathbf{y}) \rangle d\tau = J_{\mu^{Fg}}^F(\mathbf{y}). \quad (\text{A.8})$$

### A.0.5 Main Result

See Fig. A.1 and recall the relation between  $\mu^*$  and  $\mu^{F*}$ , and between  $\mu^L$  and  $\mu^{Fg}$ . Suppose that for a given fluid model, the greedy policy  $\mu^{Fg}$  is optimal, *i.e.*,  $J_{\mu^{F*}}^F(\cdot) = J_{\mu^{Fg}}^F(\cdot)$ . Then it follows from (A.4) and (A.8) that  $\mu^L$  is FSAO. Furthermore, suppose  $\mu^{Fg}$  is the *unique* (a.e.) optimal policy. Then, since  $\mu^*$  is also FSAO, the fluid limit ( $\mathbf{q}(t; \cdot)$ ,  $t \geq 0$ ) of the queue process under  $\mu^*$  (if the limit exists) and under  $\mu^L$  will be identical, and in turn, identical to the deterministic fluid trajectory ( $\mathbf{x}^{\mu^{Fg}}(t; \cdot)$ ,  $t \geq 0$ ). In other words, the switching curves under  $\mu^*$  and  $\mu^L$  on the state-space of fluid-scaled queue will be identical and, in turn, identical to the switching curves under the greedy policy  $\mu^{Fg}$ . Next, we show through a representative example that, if  $\lambda$  is *not too small*, then  $\mu^{Fg}$  is indeed the unique optimal policy for the fluid model.

To simplify the exposition, we restrict the system to  $\bar{n} = 2$  users, weight vector



$\mathbf{w} = (1, 1)$ , and a capacity region  $\mathcal{C} \in \mathbb{R}_+^{\bar{n}}$  depicted in Fig. A.2(a). That is,  $\mathcal{C}$  has a unique max-sum-rate vertex  $(a, a)$ , with the two adjacent vertices  $(a_1, a_2)$  and  $(a_2, a_1)$  satisfying  $a_1 + a_2 < 2a$ . Let the point  $(0, \hat{b})$  be such that the line segment joining points  $(a, a)$  and  $(0, \hat{b})$  passes through vertex  $(a_1, a_2)$ . Then, the region  $\mathcal{C}$  intercepts with the two axes at points  $(b, 0)$  and  $(0, b)$  for some  $b \in (a, \hat{b})$ . The remaining shape of  $\mathcal{C}$  is unspecified and can be anything (as long as  $\mathcal{C}$  remains convex).

Consider any symmetric vector  $\boldsymbol{\lambda}$  in the shaded region of  $\mathcal{C}$  in Fig. A.2, *i.e.*,

$$\lambda_1 = \lambda_2 \in [a_2, a).$$

Fig. A.2 depicts the trajectory  $(\mathbf{x}^{\mu^{Fg}}(t; \mathbf{y}), t \geq 0)$  starting from some point  $\mathbf{y} > \mathbf{0}$  such  $y_1 > y_2$ . The trajectory evolves as follows: if  $\mathbf{x}^{\mu^{Fg}}(t; \mathbf{y}) > \mathbf{0}$ , then

$$\frac{d}{dt} \mathbf{x}^{\mu^{Fg}}(t; \mathbf{y}) = \boldsymbol{\lambda} - (a, a);$$

and if  $x_1^{\mu^{Fg}}(t; \mathbf{y}) > 0, x_2^{\mu^{Fg}}(t; \mathbf{y}) = 0$ , then

$$\frac{d}{dt} \mathbf{x}^{\mu^{Fg}}(t; \mathbf{y}) = \boldsymbol{\lambda} - (c(\lambda_2), \lambda_2),$$

where  $c(\lambda_2)$  is the first coordinate of the point  $(c(\lambda_2), \lambda_2)$  on the boundary of  $\mathcal{C}$ . Then, for any starting point  $\mathbf{y}$  such that  $y_1 \geq y_2$ , it is easy to show that,

$$J_{\mu^{Fg}}^F(\mathbf{y}) = \frac{(y_1 - y_2)^2}{2(c(\lambda_2) - \lambda_1)} + \frac{y_1 y_2}{a - \lambda_2}.$$

Note that  $J_{\mu^{Fg}}^F(\cdot)$  is homogenous, *i.e.*,  $J_{\mu^{Fg}}^F(\theta \mathbf{y}) = \theta^2 J_{\mu^{Fg}}^F(\mathbf{y})$ , and so

$$\nabla J_{\mu^{Fg}}^F(\mathbf{y}) \propto \nabla J_{\mu^{Fg}}^F(\theta \mathbf{y}).$$

Now we are ready to test the policy  $\mu^{Fg}$  for optimality using (A.3). Let  $s$  denote the slope of the outer-normal to the facet joining the vertex  $(a, a)$  and  $(a_1, a_2)$  of  $\mathcal{C}$ , *i.e.*,  $s = \frac{a_1 - a}{a - a_2}$ ; see Fig. A.2. It can be shown that for any  $\mathbf{y} > \mathbf{0}$  such that  $y_1 \geq y_2$ , the slope of gradient  $\nabla J_{\mu^{Fg}}^F(\mathbf{y})$  lies in  $(s, 1]$ , *i.e.*,

$$\frac{\nabla_2 J_{\mu^{Fg}}^F(\mathbf{y})}{\nabla_1 J_{\mu^{Fg}}^F(\mathbf{y})} \in (s, 1],$$

whereas, on the  $x_1$ -axis, the slope is equal to  $s$ . Then by (A.3), we have that  $\boldsymbol{\mu}^{Fg}$  is the unique optimal policy. In fact, along the lines of [66], one can show that  $\boldsymbol{\mu}^{Fg}$  is the optimal policy for any  $\boldsymbol{\lambda}$ , not necessarily symmetric, that lies in the shaded region of  $\mathcal{C}$ . Therefore, for the corresponding MDP with  $\boldsymbol{\lambda}, \mathcal{C}$ , and  $\boldsymbol{w}$  as described above,  $\boldsymbol{\mu}^L$  is FSAO and has same switching curves on the state-space of fluid-scaled queue as those of  $\boldsymbol{\mu}^*$ .

However, it can also be shown that for any  $\boldsymbol{\lambda} < (a_2, a_2)$ , the policy  $\boldsymbol{\mu}^{Fg}$  is *not* optimal for the fluid model. That is, there exist states  $\boldsymbol{x} \in \mathbb{R}_+^{\bar{n}}$  for which  $J_{\boldsymbol{\mu}^{Fg}}^F(\boldsymbol{x}) > J_{\boldsymbol{\mu}^{F*}}^F(\boldsymbol{x})$ . It follows from (A.4) and (A.8) that RSM policies like  $\boldsymbol{\mu}^L$  will *not* be FSAO. Since the optimal policies  $\boldsymbol{\mu}^*$  are FSAO, they cannot be RSM.

## Appendix B

### A note on delay-optimal opportunistic schedulers for multichannel systems

To analyze the queueing system embedded in a centralized, OFDMA-based, multiuser wireless system, a few recent works [42, 61, 62] have introduced a model with multiple parallel servers, where the speed/capacity of a server for each user varies over time, independently of that of other users.<sup>1</sup>Using the classical terminology (see, *e.g.*, [4]), such servers can perhaps be termed “*time-varying*” *unrelated parallel processors*, “unrelated” refers to the fact that the speed at which the server can process one user can be different from that of another user. This model provides a method of abstracting out the Physical layer and focusing on structural properties of delay-optimal control/scheduling of the parallel contending queues. A closely related model is used in [67] for a multibeam satellite communication systems, the difference being that the parallel servers are *uniform* rather than *unrelated*.

Various analytical and optimality results in [42, 61, 62] are all restricted to servers with *on-off* capacities and under certain symmetry assumptions. Moreover, all proposed schedulers are described as solutions of certain bipartite matching problems (*e.g.*, finding the maximum-weighted matching). Arguably, such a presentation of a scheduler offers little hint at generalizing the result to more realistic heterogeneous systems with *multi-speed* servers (equivalently, systems that support adaptive

---

<sup>1</sup>In this model, each server is often viewed as an integral component of bandwidth, *e.g.*, a resource block (RB) in LTE, and therefore, can only be allocated to at most one user over a transmission time interval (TTI). A server can also be viewed as a subband, *i.e.*, a collect of few consecutive RBs, in which case the server can be allocated to multiple users over a TTI.

modulation and coding), instead of just on-off servers.

In this appendix, we give an alternative description of the scheduling problem discussed in [42, 61, 62, 67], this description can be related to the dual of the bipartite matching problem. The alternative description offers the following.

- i. Certain proofs can be drastically shortened.
- ii. The fact that even though mean-delay optimal schedulers for single-server systems (see [12]) can be described as a *linear* convex program, such schedulers become a nonlinear convex program for multi-server systems.
- iii. Immediate extension of the schedulers to more realistic systems with multi-speed servers and heterogeneity.
- iv. Methods of implementation in real systems, taking into account practical limits on the spectral granularity of channel feedback, and leveraging vast literature on optimization and resource allocation in OFDMA.

## B.1 System Model

In the following we describe a slotted-time multi-queue multi-server system. Let the positive integer  $\bar{n}$  denote the total number of users, or equivalently, the number of parallel queues, and let  $\mathcal{N} \equiv \{1, 2, \dots, \bar{n}\}$ . Let  $\mathcal{P}(\mathcal{N})$  denote the set of probability mass functions on  $\mathcal{N}$  and  $\mathcal{P}_0(\mathcal{N}) \subset \mathcal{P}(\mathcal{N})$  the set of Dirac measures. For any  $t \in \mathbb{Z}_+$  and  $n \in \mathcal{N}$ , let the random variable  $A_n(t)$  denote the number of packets arriving in the  $n^{\text{th}}$  queue in the beginning of time slot  $[t, t + 1)$ . We assume that for each  $n$  and  $t$ ,  $A_n(t)$  is drawn independently from the same distribution as that of random  $A$ , and takes values in a finite subset of  $\mathbb{Z}_+$ . Let  $\mathbf{A}(t) \equiv (A_n(t), n \in \mathcal{N})$ . Also, let  $Q_n(t)$  denote the number of packets in the  $n^{\text{th}}$  queue at the end of time slot  $[t - 1, t)$  and let  $\mathbf{Q}(t) \equiv (Q_n(t), n \in \mathcal{N})$ .

Let a positive integer  $\bar{m}$  denote the number parallel servers and let the set  $\mathcal{M} \equiv \{1, \dots, \bar{m}\}$ . At any time  $t \in \mathbb{Z}_+$ , let the random variable  $R_{mn}(t) \in \mathbb{Z}_+$  denote the state (or speed) of the  $m^{\text{th}}$  server for  $n^{\text{th}}$  user over time slot  $[t, t + 1)$ . That is, if the  $m^{\text{th}}$  server is allocated to the  $n^{\text{th}}$  user at time  $t$ , then  $R_{mn}(t)$  number of packets can be served from the  $n^{\text{th}}$  user's queue. We assume that for each  $m, n$  and  $t$ , the state  $R_{mn}(t)$  is independently drawn from the same distribution as that of random  $R$ . Let the matrix  $\mathbf{R}(t) \equiv (R_{mn}(t), (m, n) \in \mathcal{M} \times \mathcal{N})$ ; we will refer to  $\mathbf{R}(t)$  as the servers' state at time  $t$ . We will enumerate some assumptions regarding the servers, in the sequel we will investigate the system under various combination of these assumptions:

- **Assumption B.1** *On-off server – the random variable  $R$  takes values in set  $\{0, 1\}$ .*
- **Assumption B.2** *Multi-rate server – the random variable  $R$  takes values in a finite subset of  $\mathbb{Z}_+$ .*
- **Assumption B.3** *Integral server – a server can only be allocated to at most one user over a time slot.*
- **Assumption B.4** *Divisible server – a server can be shared by multiple users over a time slot.*

Let  $S(t) \equiv (\mathbf{Q}(t), \mathbf{A}(t), \mathbf{R}(t))$  denote the random system state at time  $t \in \mathbb{Z}_+$ , that takes values in space  $\mathcal{S}$ . The scheduling problem in this context is to allocate the  $\bar{m}$  servers to  $\bar{n}$  users based on the current system state, such that a given optimality criterion is achieved. More precisely, a scheduling policy or scheduler is described as a mapping

$$s(t) \rightarrow \boldsymbol{\pi}(s(t)) \equiv (\pi_{mn}(s(t)), (m, n) \in \mathcal{M} \times \mathcal{N}),$$

where, for all  $m \in \mathcal{M}$ ,

- under Assumption (B.3), we have that  $(\pi_{mn}(s(t)), n \in \mathcal{N}) \in \mathcal{P}_0(\mathcal{N})$ , and  $\pi_{mn}(s(t)) \in \{0, 1\}$  denotes whether or not the  $m^{\text{th}}$  server is allocated to the  $n^{\text{th}}$  queue;
- under Assumption (B.4), we have that  $(\pi_{mn}(s(t)), n \in \mathcal{N}) \in \mathcal{P}(\mathcal{N})$  and  $\pi_{mn}(s(t)) \in [0, 1]$  denotes the fraction of the time slot  $[t, t + 1)$  over which the  $m^{\text{th}}$  server is allocated to the  $n^{\text{th}}$  user.

Under a policy  $\boldsymbol{\pi}(\cdot)$ , the total service offered to the  $n^{\text{th}}$  queue, conditional on the system being in state  $s \equiv (\mathbf{q}, \mathbf{a}, \mathbf{r})$ , is given by,

$$\mu_n(s) = \sum_{m \in \mathcal{M}} r_{mn} \pi_{mn}(s), \quad n \in \mathcal{N}. \quad (\text{B.1})$$

Let  $\boldsymbol{\mu}(s) \equiv (\mu_n(s), n \in \mathcal{N})$ , and let  $\Gamma$  denote the functional defined by (B.1), *i.e.*,  $\boldsymbol{\mu}(\cdot) \equiv \Gamma \boldsymbol{\pi}(\cdot)$ . Under the policy  $(\boldsymbol{\pi}(s), s \in \mathcal{S})$ , the queue process evolves as follows,

$$Q_n(t+1) = (Q_n(t) + A_n(t) - \mu_n(s))^+, \quad n \in \mathcal{N}.$$

Let  $\Pi$  be the set of all admissible scheduling policies under Assumption (B.4) and  $\Pi_0 \subset \Pi$  that of under Assumption (B.3). Note that under Assumption (B.3), the queue process is restricted to  $\mathbb{Z}_+^{\bar{n}}$ , whereas under Assumption (B.4), it can lie in  $\mathbb{R}_+^{\bar{n}}$ .

This concludes the description of system model which, under Assumption (B.1), captures the models in [42, 61, 62]. The next section is the departing point from these works.

## B.2 Alternative view of the system

Assumption (B.1) holds throughout this section.

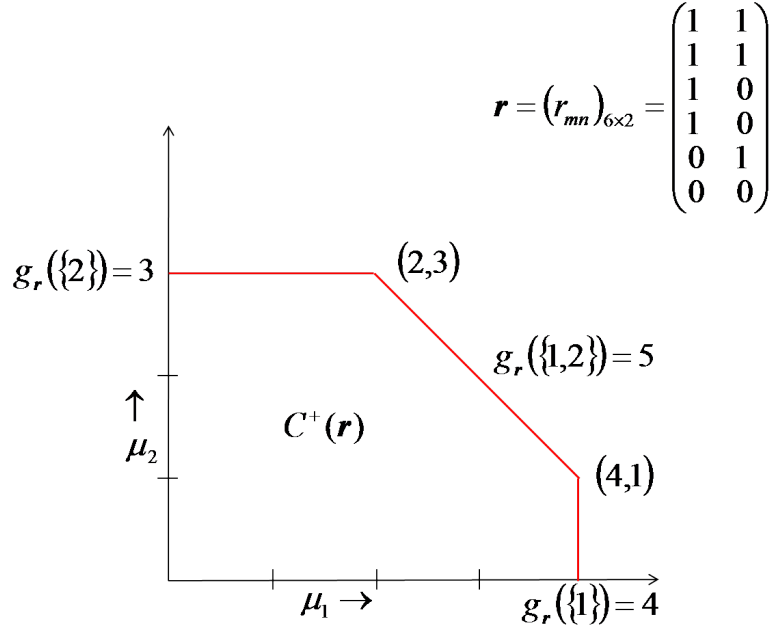


Figure B.1: Illustration of  $\mathcal{C}(\mathbf{r})$  for a  $\bar{n} = 2$  user  $\bar{m} = 6$  server system.

### B.2.1 From “On-Off servers” to “polymatroid service regions”

For any servers’ state  $\mathbf{r} = (r_{mn}, (m, n) \in \mathcal{M} \times \mathcal{N})$ , define a set function  $g_{\mathbf{r}} : 2^{\mathcal{N}} \rightarrow \mathbb{Z}_+$  as follows: for any  $\mathcal{N}' \subseteq \mathcal{N}$ ,

$$g_{\mathbf{r}}(\mathcal{N}') = \sum_{m \in \mathcal{M}} \mathbf{1}_{\{r_{mn}=1 \text{ for some } n \in \mathcal{N}'\}},$$

(with  $g_{\mathbf{r}}(\{\}) \equiv 0$ .) In words,  $g_{\mathbf{r}}(\mathcal{N}')$  denotes the number of server that are “On” for at least one user in set  $\mathcal{N}'$ . The total service rate (in number of packets per slot) that can be offered to the users in set  $\mathcal{N}'$  is bounded above by  $g_{\mathbf{r}}(\mathcal{N}')$ . Moreover, it is easy to verify that for any  $\mathbf{r}$ , the function  $g_{\mathbf{r}}(\cdot)$  is submodular and increasing. Therefore, the region  $\mathcal{C}(\mathbf{r}) \in \mathbb{R}^{\bar{n}}$  defined as,

$$\mathcal{C}(\mathbf{r}) \equiv \left\{ \mathbf{x} \in \mathbb{R}^{\bar{n}} : \forall \mathcal{N}' \subseteq \mathcal{N}, \sum_{n \in \mathcal{N}'} x_n \leq g_{\mathbf{r}}(\mathcal{N}') \right\}$$

and the region

$$\mathcal{C}^+(\mathbf{r}) \equiv \mathcal{C}(\mathbf{r}) \cap \mathbb{R}_+^{\bar{n}} \tag{B.2}$$

form an extended polymatroid and a polymatroid, respectively; (see, *e.g.*, [54, p. 767]). See Fig. B.1 for an illustrative example. Also, let

$$\mathcal{C}_0(\mathbf{r}) \equiv \mathcal{C}(\mathbf{r}) \cap \mathbb{Z}^{\bar{n}} \quad \text{and} \quad \mathcal{C}_0^+(\mathbf{r}) \equiv \mathcal{C}(\mathbf{r}) \cap \mathbb{Z}_+^{\bar{n}}, \quad (\text{B.3})$$

*i.e.*, the subset of integer points of the extended polymatroid and the polymatroid respectively.

The region  $\mathcal{C}^+(\mathbf{r})$  has the following interpretation: *conditional on the servers being in state  $\mathbf{r}$ , the region  $\mathcal{C}^+(\mathbf{r})$  denotes the set of service rates that can be jointly offered to the  $\bar{n}$  queues by server allocations that are admissible under Assumption (B.4) (divisible server).* That is to say, under any scheduling policy  $\boldsymbol{\pi}(\cdot) \in \Pi$  and for any system state  $s \equiv (\mathbf{q}, \mathbf{a}, \mathbf{r})$ , we have that  $\boldsymbol{\mu}(s) \equiv \Gamma \boldsymbol{\pi}(s)$  (see (B.1)) lies<sup>2</sup> in  $\mathcal{C}^+(\mathbf{r})$ .

Similarly, the region  $\mathcal{C}_0^+(\mathbf{r})$  denotes the set of service rates that can be jointly offered to the  $\bar{n}$  queues by server allocations that are admissible under Assumption (B.3) (integral servers). That is, under any scheduling policy  $\boldsymbol{\pi}(\cdot) \in \Pi_0$  and for any system state  $s \equiv (\mathbf{q}, \mathbf{a}, \mathbf{r})$ , we have that  $\boldsymbol{\mu}(s) \equiv \Gamma \boldsymbol{\pi}(s) \in \mathcal{C}_0^+(\mathbf{r})$ .

Therefore, we will refer to  $\mathcal{C}^+(\mathbf{r})$  and  $\mathcal{C}_0^+(\mathbf{r})$  as service regions and to  $\mathcal{C}(\mathbf{r})$  and  $\mathcal{C}_0(\mathbf{r})$  as extended service regions, associated with servers' state  $\mathbf{r}$ . Polymatroids are well-studied objects in the context of combinatorial optimization, and all results presented in subsequent sections follow from this polymatroid structure of service regions, by leveraging the results obtained/summarized in, *e.g.*, [68].

## B.2.2 From “server allocation through bipartite matching” to “service allocation”

In light of the above subsection, instead of defining a scheduling policy as picking a *server* allocation  $\boldsymbol{\pi}(s)$  based on the current system state  $s$ , one can define

---

<sup>2</sup>The mapping from  $\boldsymbol{\pi}(\cdot)$  to  $\boldsymbol{\mu}(\cdot)$  defined by (B.1) is surjective.



it as picking a *service* allocation  $\boldsymbol{\mu}(s) \in \mathcal{C}(\mathbf{r})$  based on the current system state  $s \equiv (\mathbf{q}, \mathbf{a}, \mathbf{r})$ . That is, a scheduling policy can be redefined as a mapping

$$(\mathbf{q}(t), \mathbf{a}(t), \mathbf{r}(t)) \rightarrow \boldsymbol{\mu}((\mathbf{q}(t), \mathbf{a}(t), \mathbf{r}(t))) \in \mathcal{C}^+(\mathbf{r})$$

or  $\mathcal{C}_0^+(\mathbf{r})$ , as appropriate. We will denote the set of service allocation policies that are admissible under Assumption (B.3) as  $\bar{\Pi}_0$  and that of those under Assumption (B.4) as  $\bar{\Pi}$ .

### B.2.3 Main Result

Let us formally define the notion of “least weakly submajorized element” of a subset of  $\mathbb{R}_{\bar{n}}$ .

**Definition B.1** For any  $\mathbf{x} \equiv (x_n, n \in \mathcal{N}) \in \mathbb{R}_{\bar{n}}$ , let  $x_{[n]}$  denote its  $n^{\text{th}}$  largest component with ties broken arbitrarily, i.e.,  $x_{[1]} \geq x_{[2]} \geq \dots \geq x_{[\bar{n}]}$ . Then for any  $\mathbf{x}$  and  $\mathbf{y} \equiv (y_n, n \in \mathcal{N})$  in  $\mathbb{R}_n$ , we say that  $\mathbf{x}$  is weakly submajorized by  $\mathbf{y}$ ,  $\mathbf{x} \ll \mathbf{y}$ , if for all  $n \in \mathcal{N}$ ,

$$\sum_{k=1}^n x_{[k]} \leq \sum_{k=1}^n y_{[k]}.$$

Moreover, for any subset  $\mathcal{R}$  of  $\mathbb{R}_{\bar{n}}$ , we say a point  $\mathbf{x} \in \mathcal{R}$  is a least weakly submajorized element of  $\mathcal{R}$  if  $\mathbf{x} \ll \mathbf{y}$  for all  $\mathbf{y} \in \mathcal{R}$ .

Let  $s(t) = (\mathbf{q}(t), \mathbf{a}(t), \mathbf{r}(t))$  denote the system state at time  $t$  and let  $\mathcal{Q}(s(t))$  denote the set of all possible queue states at the end of time slot  $[t, t+1)$  (i.e.  $\mathbf{q}(t+1)$ ) that are achievable under scheduling policies in  $\bar{\Pi}$ , i.e.,

$$\begin{aligned} \mathcal{Q}(s(t)) &= \{\mathbf{q}(t) + \mathbf{a}(t) - \mathcal{C}^+(\mathbf{r}(t))\} \cap \mathbb{R}_+^{\bar{n}}, \\ &= \{\mathbf{q}(t) + \mathbf{a}(t) - \mathcal{C}(\mathbf{r}(t))\} \cap \{\mathbf{y} \in \mathbb{R}^{\bar{n}} : \mathbf{0} \leq \mathbf{y} \leq \mathbf{q}(t) + \mathbf{a}(t)\}. \end{aligned} \quad (\text{B.4})$$

The set

$$\{\mathbf{q}(t) + \mathbf{a}(t) - \mathcal{C}(\mathbf{r}(t))\}$$

in (B.4) is a contra-polymatroid and the set

$$\{\mathbf{y} \in \mathbb{R}^{\bar{n}} : \mathbf{0} \leq \mathbf{y} \leq \mathbf{q}(t) + \mathbf{a}(t)\}$$

is simply a cuboid, and their intersection, that is the set  $\mathcal{Q}(s(t))$ , is a generalized polymatroid (see, *e.g.*, [68]). Similarly, the set  $\mathcal{Q}_0(s(t)) \equiv \mathcal{Q}(s(t)) \cap \mathbb{Z}_+^{\bar{n}}$  (with  $\mathbf{q}(t) \in \mathbb{Z}_+^{\bar{n}}$ ) denotes the set of queue states achievable under scheduling policies in  $\bar{\Pi}_0$ . Then, by Theorem 2.3, 3.2, and Section 4 of [68], we have the following results.

**Theorem B.1** *For any  $s \equiv (\mathbf{q}, \mathbf{a}, \mathbf{r})$ , the following hold.*

(i) *There exists a service allocation  $\boldsymbol{\mu}^*(s) \in \mathcal{C}^+(\mathbf{r})$  such that the next queue state,*

$$\mathbf{q}^* = (\mathbf{q} + \mathbf{a} - \boldsymbol{\mu}^*(s))^+ \in \mathcal{Q}(s)$$

*is the unique least weakly submajorized element of  $\mathcal{Q}(s)$ .*

(ii) *For all nondecreasing convex functions  $f : \mathbb{R} \rightarrow \mathbb{R}$ , we have that,*

$$\sum_{n \in \mathcal{N}} f(q_n^*) = \min_{\mathbf{y} \in \mathcal{Q}(s)} \sum_{n \in \mathcal{N}} f(y_n). \quad (\text{B.5})$$

(iii) *In the case of integral servers and assuming  $\mathbf{q} \in \mathbb{Z}_+^{\bar{n}}$ , there exists a service allocation  $\boldsymbol{\mu}^*(s) \in \mathcal{C}_0^+(\mathbf{r})$  such that the next queue state,*

$$\mathbf{q}^* = (\mathbf{q} + \mathbf{a} - \boldsymbol{\mu}^*(s))^+ \in \mathcal{Q}_0(s)$$

*is the a least weakly submajorized element of  $\mathcal{Q}_0(s)$ ; moreover, we have that,*

$$\sum_{n \in \mathcal{N}} f(q_n^*) = \min_{\mathbf{y} \in \mathcal{Q}_0(s)} \sum_{n \in \mathcal{N}} f(y_n). \quad (\text{B.6})$$

See Fig. B.2 for an illustration of the weakly least majorized point of set  $\mathcal{Q}(\cdot)$ , and that the set  $\mathcal{Q}_0(\cdot)$  may not have a unique such point (see the region that intersects with the line  $q_1 = q_2$ ).

Theorem B.1 has following important implications.

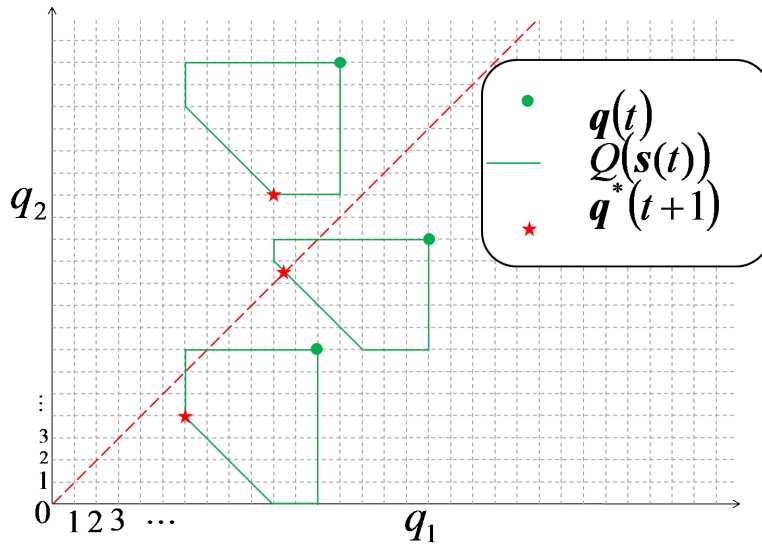


Figure B.2: Illustration of  $\mathcal{Q}(s(t))$  for various states  $s$  and the corresponding least weakly submajorized element  $\mathbf{q}^*(t+1)$ .

### B.2.3.1 Absence of tradeoff between *maximizing total service rate* and *balancing unequal queues*

In general, opportunistic scheduling involves a tradeoff between *maximizing total current service* and *balancing unequal queues* (see Chapter 1 or [61]). However, Theorem B.1(i) immediately shows that in the case of on-off servers, the above mentioned tradeoff is absent. That is, in any system state, there exists a server allocation such that the next queue state (queue state after receiving service) is weakly submajorized by that under any other server allocation. This was originally shown in [61] using the existence and certain properties of a bipartite matching.

Moreover, the above-mentioned tradeoff is clearly absent in the case of a single ( $\bar{m} = 1$ ) integral on-off server, because the objective of balancing queues and maximizing total current service is simultaneously achieved by allocating the server to a longest “On” queue; this is the Longest Connected Queue (LCQ) policy from [12]. [12] shows through a coupling argument that under LCQ policy, the queue process is stochastically weakly submajorized. It is noteworthy that the coupling argument used in [12]

to prove this strong optimality of LCQ policy has no known extension to the case of multiple on-off servers (whether integral or divisible). That is, for  $\bar{m} > 1$ , it is not known if the policy  $\boldsymbol{\mu}^*(\cdot)$  – which can be viewed as an extension of LCQ for the case of multiple parallel servers – leads to stochastic weak submajorization of  $\mathbf{Q}(\cdot)$ . However, for the case of divisible servers,  $\boldsymbol{\mu}^*(\cdot)$  does minimize various functionals of  $(\mathbf{Q}(t), t \geq 0)$  – we discuss this next.

### B.2.3.2 Restatement of the delay-optimal policy MTLB for divisible servers

Through this subsection, we assume a stabilizable system with *divisible* servers (Assumption (B.4)). The policy “maximizing throughput (while simultaneously) load balancing” (MTLB), introduced in [61], is indeed the policy  $\boldsymbol{\mu} * (\cdot) \in \Pi$  of Theorem B.1(i). Using dynamic programming, [61] shows that for all convex increasing functions  $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ ,

$$\mathbb{E}^{\boldsymbol{\mu}^*} \left( \sum_{n \in \mathcal{N}} f(Q_n) \right) = \min_{\boldsymbol{\mu}(\cdot) \in \Pi} \mathbb{E}^{\boldsymbol{\mu}} \left( \sum_{n \in \mathcal{N}} f(Q_n) \right),$$

where the expectation  $\mathbb{E}^{\boldsymbol{\mu}} (\sum_{n \in \mathcal{N}} f(Q_n))$  is with respect to random  $\mathbf{Q}$  drawn from the stationary distribution of queue process under a stable policy  $\boldsymbol{\mu}(\cdot)$ .

In [61], the optimal policy  $\boldsymbol{\mu}^*(s)$  is described as a maximum weighted matching on a bipartite graph constructed from the elements of state  $s = (\mathbf{q}, \mathbf{a}, \mathbf{r})$ . However, Theorem B.1(ii) gives us an alternative method of solving for the optimal queue state  $\mathbf{q}^* \in \mathcal{Q}(s)$  (and implicitly  $\boldsymbol{\mu}^*(s)$ ). That is, the queue state  $\mathbf{q}^*$  is the unique solution of the following convex program for any *strictly* convex  $f : \mathbb{R} \rightarrow \mathbb{R}$ :

$$\begin{aligned} & \text{minimize}_{\mathbf{y}} \quad \sum_{n \in \mathcal{N}} f(y_n) \\ & \text{subject to} \quad \mathbf{y} \in \mathcal{Q}(s). \end{aligned} \tag{B.7}$$

For example,  $\sum_{n \in \mathcal{N}} f(y_n) = \|\mathbf{y}\|_p^p$  for any  $p \in (1, \infty)$ . Equivalently, for any strictly convex  $f$  such that  $f(0) \leq f(y)$  for all  $y \in \mathbb{R}$ , we have,

$$\boldsymbol{\mu}^*(s) = \underset{\boldsymbol{\mu} \in \mathcal{C}^+(\mathbf{r})}{\text{argmin}} \quad \sum_{n \in \mathcal{N}} f(q_n + a_n - \mu_n). \tag{B.8}$$

Unlike the construction of bipartite matching problem in [61], this convex program representation is readily extendable to a system with multi-rate servers (Assumption B.2) by appropriately redefining the convex region  $\mathcal{C}(\mathbf{r})$  or  $\mathcal{Q}(s)$ ; this will be further explored in Section B.3.

### B.2.3.3 Nonlinearity of delay-optimal policy for *multiserver*

For  $\bar{m} > 1$ , the optimal service allocation policy  $\boldsymbol{\mu}^*(\cdot)$  (see (B.8)) is a *non-linear* convex program (recall that  $f(\cdot)$  is strictly convex). Whereas, the (even more strongly) optimal scheduler in the case of a single, integral, on-off server, namely LCQ, is a *linear* program given by replacing the objective function in (B.8) with  $\langle \mathbf{q}, \boldsymbol{\mu} \rangle$ .

### B.2.3.4 Restatement of the delay-optimal policy for integral servers

[61] also shows the optimality of policy  $\boldsymbol{\mu}^*(\cdot) \in \bar{\Pi}_0$  given in Theorem B.1(iii) for a  $\bar{n} = 2$  user stabilizable system with *integral* servers (Assumption (B.3)). As noted in Theorem B.1(iii), in particular (B.6), the optimal next queue state  $\mathbf{q}^* \in \mathcal{Q}(s)$  is still a solution of the convex program (B.7) *with the set  $\mathcal{Q}(s)$  replaced by the set  $\mathcal{Q}_0(s)$* . However, the solution of this integer program may not be unique, nevertheless, all solutions are optimal.

### B.2.3.5 “Server-side greedy” policy

The server-side greedy (SSG) policy was introduced in [43] as an extension of MaxWeight policy to multiserver systems, and shown in [42] to minimize the asymptotic probability of max-queue overflow in the more relevant *small buffer regime*, as the number of queues and servers increase. The SSG policy can be viewed as a (sub-optimal) greedy algorithm for solving (B.8). That is, SSG policy iterates over servers in an arbitrary order and allocates the next available server to a queue so as to bring

about the greatest decrease in the objective function.

### B.3 Extending the schedulers for on-off servers to multi-rate servers

When the divisible servers are in state  $\mathbf{r} = (r_{mn}, (m, n) \in \mathcal{M} \times \mathcal{N})$  where  $r_{mn} \in \{0, 1\}$ , the set of all feasible service allocations is given by the polymatroid  $\mathcal{C}^+(\mathbf{r})$  (see B.2). Next, we define the set of feasible service allocations for the case of multi-rate servers, *i.e.*, where  $r_{mn}$  can be any nonnegative integer.

#### B.3.1 Service region associated with multi-rate servers

More generally, when  $r_{mn} \in \mathbb{Z}_+$ , the set of all feasible service allocations, still denoted by  $\mathcal{C}^+(\mathbf{r})$ , can be given as follows. Recall that  $(r_{m,n}, n \in \mathcal{N}) \in \mathbb{Z}_+^{\bar{n}}$  denotes the state of the  $m^{\text{th}}$  server. Let  $\text{conv}(r_{mn}, n \in \mathcal{N})$  denote the convex hull of origin and the following  $\bar{n}$  points in  $\mathbb{R}_+^{\bar{n}}$ :

$$r_{m1}\mathbf{e}_1, r_{m2}\mathbf{e}_2, \dots, r_{m\bar{n}}\mathbf{e}_{\bar{n}},$$

where  $\mathbf{e}_n$  is the  $n^{\text{th}}$  standard basis vector. Then  $\mathcal{C}^+(\mathbf{r})$  is given by,

$$\mathcal{C}^+(\mathbf{r}) \equiv \text{conv}(r_{1n}, n \in \mathcal{N}) \oplus \text{conv}(r_{2n}, n \in \mathcal{N}) \oplus \dots \oplus \text{conv}(r_{\bar{m}n}, n \in \mathcal{N}),$$

where  $\oplus$  denotes Minkowski sum. The region  $\mathcal{C}^+(\mathbf{r})$  is a compact, convex, coordinate convex polytope in  $\mathbb{R}_+^{\bar{n}}$ . It can be verified that if  $r_{mn} \in \{0, 1\}$ , the above definition of  $\mathcal{C}^+(\mathbf{r})$  is equivalent to the one in (B.2).

#### B.3.2 Scheduling policies for systems with multi-rate servers

For any convex  $f : \mathbb{R} \rightarrow \mathbb{R}$  with  $f(0) \leq f(y)$  for all  $y \in \mathbb{R}$ , let  $\boldsymbol{\mu}^f(\cdot)$  be any service allocation policy that satisfies the following: for any  $s = (\mathbf{q}, \mathbf{a}, \mathbf{r})$ ,

$$\boldsymbol{\mu}^f(s) \in \underset{\boldsymbol{\mu} \in \mathcal{C}^+(\mathbf{r})}{\text{argmin}} \sum_{n \in \mathcal{N}} f(q_n + a_n - \mu_n). \quad (\text{B.9})$$

That is, the queue state after service,  $(\mathbf{q} + \mathbf{a} - \boldsymbol{\mu}^f(s))^+ \in \operatorname{argmin}_{\mathbf{y} \in \mathcal{Q}(s)} \sum_{n \in \mathcal{N}} f(y_n)$ .

In general, since the region  $\mathcal{Q}(s)$  is not a generalized polymatroid, it may not have a least weakly majorized element. Therefore, different choices for  $f$  will lead to different scheduling policies and different tradeoffs between maximizing total service rate,  $\|\boldsymbol{\mu}^f(\cdot)\|_1$ , and balancing unequal queues.

In fact, the objective function in (B.9),  $\sum_{n \in \mathcal{N}} f(q_n + a_n - \mu_n)$ , can be seen as an estimate for the relative cost function  $h(\cdot)$  on the state space of queue process (see (2.12) and Lemma (2.2)). For any  $\alpha > 0$ , by setting

$$\sum_{n \in \mathcal{N}} f(q_n + a_n - \mu_n) = \|\mathbf{q} + \mathbf{a} - \boldsymbol{\mu}\|_{(\alpha+1)}^{(\alpha+1)}$$

in (B.9), one obtains a class of policies which can be seen as an extension of single-channel MaxWeight policy with exponent  $\alpha$  (see (2.17)). Similarly, by replacing the objective function in (B.9) by  $h^L(\mathbf{y})$  (see (2.29)), one obtains the multi-channel version of the Log rule, which was proposed in Section 2.7. Recall that the multi-channel Log rule is RSM whereas multi-channel MaxWeight is not.

## Bibliography

- [1] M. Wemersson, S. Wanstedt, and P. Synnergren, “Effects of QoS scheduling strategies on performance of mixed services over LTE,” in *Proc. IEEE Int. Symp. on Pers., Indoor and Mobile Radio Comm.*, 3-7 2007, pp. 1–5.
- [2] S. Sesia, I. Toufik, and M. Baker, Eds., *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley, April 2009.
- [3] S. Shakkottai, “Effective capacity and QoS for wireless scheduling,” *IEEE Trans. Automatic Control*, vol. 53, no. 3, pp. 749–761, April 2008.
- [4] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*, 3rd ed. Prentice-Hall, 2008.
- [5] R. Knopp and P. Humblet, “Information capacity and power control in single-cell multiuser communications,” in *Proc. IEEE Int. Conf. on Comm.*, vol. 1, Jun 1995, pp. 331–335.
- [6] P. Viswanath, D. Tse, and R. Laroia, “Opportunistic beamforming using dumb antennas,” *IEEE Trans. Information Theory*, vol. 48, no. 6, pp. 1277–1294, Jun 2002.
- [7] X. Liu, E. K. P. Chong, and N. B. Shroff, “A framework for opportunistic scheduling in wireless networks,” *Comput. Networks*, vol. 41, no. 4, pp. 451–474, 2003.
- [8] M. J. Neely, “Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks,” *IEEE/ACM Trans. Networking*, 2008.



- [9] M. Andrews, “Instability of the proportional fair scheduling algorithm for HDR,” *IEEE Trans. Wireless Comm.*, vol. 3, no. 5, pp. 1422–1426, Sept. 2004.
- [10] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queuing system with asynchronously varying service rates,” *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, 2004.
- [11] S. Shakkottai and A. Stolyar, “Scheduling for multiple flows sharing a time-varying channel: The Exponential Rule,” *American Mathematical Society Transactions, Series 2*, vol. 207, 2002.
- [12] L. Tassiulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Information Theory*, vol. 39, no. 2, pp. 466–478, Mar 1993.
- [13] E. M. Yeh and A. S. Cohen, “Delay optimal rate allocation in multiaccess fading communications,” in *Proc. of Allerton Conf. on Comm. Control and Comp.*, 2004, pp. 140–149.
- [14] B. Hajek, “Optimal control of two interacting service stations,” *IEEE Trans. Automatic Control*, vol. 29, no. 6, pp. 491–499, Jun 1984.
- [15] V. Subramanian, “Large deviations of max-weight scheduling policies on convex rate regions,” in *Proc. of IEEE Information Theory Workshop*, Feb. 2008, pp. 535–544.
- [16] V. Subramanian, T. Javidi, and S. Kittipiyakul, “Many sources large deviations of max-weight scheduling.” *submitted*.
- [17] D. Bertsimas, I. Paschalidis, and J. Tsitsiklis, “Asymptotic buffer overflow probabilities in multiclass multiplexers: an optimal control approach,” *IEEE Trans. Automatic Control*, vol. 43, no. 3, pp. 315–335, Mar 1998.

- [18] V. J. Venkataramanan and X. Lin, “On wireless scheduling algorithms for minimizing the queue-overflow probability,” *submitted*.
- [19] A. L. Stolyar, “Large deviations of queues sharing a randomly time-varying server,” *Queueing Syst. Theory Appl.*, vol. 59, no. 1, pp. 1–35, 2008.
- [20] D. Shah and D. Wischik, “Optimal scheduling algorithms for input-queued switches,” *Proc. IEEE INFOCOM*, April 2006.
- [21] L. Schrage, “A proof of the optimality of the Shortest Remaining Processing Time discipline,” *Opns. Res.*, vol. 16, pp. 687–690, 1968.
- [22] S. Borst, “User-level performance of channel-aware scheduling algorithms in wireless data networks,” *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 636–647, 2005.
- [23] T. Bonald, S. C. Borst, and A. Proutière, “How mobility impacts the flow-level performance of wireless data systems,” in *Proc. IEEE INFOCOM*, vol. 3, 2004, pp. 1872–1881.
- [24] B. Tsybakov, “File transmission over wireless fast fading downlink,” *IEEE Trans. Information Theory*, vol. 48, no. 8, pp. 2323–2337, 2002.
- [25] M. Hu, J. Zhang, and J. Sadowsky, “Traffic aided opportunistic scheduling for wireless networks: algorithms and performance bounds,” *Comput. Netw.*, vol. 46, no. 4, pp. 505–518, 2004.
- [26] S. Aalto and P. Lassila, “Impact of size-based scheduling on flow-level performance in wireless downlink data channels,” in *Proc. of ITC*, 2007.
- [27] P. Lassila and S. Aalto, “Combining opportunistic and size-based scheduling in wireless systems,” in *Proc. of MSWiM*, 2008.

- [28] P. van de Ven, S. Borst, and S. Shneer, “Instability of maxweight scheduling algorithms,” in *Proc. IEEE INFOCOM*, April 2009.
- [29] S. Liu, L. Ying, and R. Srikant, “Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics,” in *Proc. IEEE INFOCOM*, 2010.
- [30] D. Park, H. Seo, H. Kwon, and B. G. Lee, “A new wireless packet scheduling algorithm based on the CDF of user transmission rates,” in *Proc. IEEE Glob. Telecom. Conf.*, vol. 1, Dec. 2003, pp. 528–532.
- [31] S. Patil and G. de Veciana, “Managing resources and quality of service in heterogeneous wireless systems exploiting opportunism,” *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1046–1058, 2007.
- [32] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi, “CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users,” *IEEE Comm. Mag.*, vol. 38, no. 7, pp. 70–77, jul 2000.
- [33] R. Prakash and V. V. Veeravalli, “Centralized wireless data networks with user arrivals and departures,” *IEEE Trans. Information Theory*, vol. 53, no. 2, pp. 695–713, Feb 2007.
- [34] V. S. Borkar, “Controlled markov chains and stochastic networks,” *SIAM Journal on Control and Optimization*, vol. 21, no. 4, pp. 652–666, 1983.
- [35] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995, vol. 2.
- [36] S. Stidham and R. Weber, “A survey of Markov decision models for control of networks of queues,” *Queueing Systems*, vol. 13, no. 1, pp. 291–314, Mar 1993.
- [37] C. Zhou and G. Wunder, “General stability conditions in wireless broadcast channels,” in *Proc. of Allerton Conf. on Comm. Control and Comp.*, Sep 2008.

- [38] S. Meyn, *Control techniques for complex networks*. Cambridge, 2008.
- [39] S. Shakkottai and A. Stolyar, “Scheduling algorithms for a mixture of real-time and non-real-time data in HDR,” in *Proc. of ITC*, Sep 2001.
- [40] R. Madan, S. P. Boyd, and S. Lall, “Fast algorithms for resource allocation in cellular networks,” *IEEE/ACM Trans. Networking*, submitted.
- [41] J. Huang, V. G. Subramanian, R. Berry, and R. Agrawal, *Orthogonal Frequency Division Multiple Access*. Auerbach Publications, CRC Press, to appear, ch. Scheduling and Resource Allocation in OFDMA Wireless Systems.
- [42] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, “Low-complexity scheduling algorithms for multi-channel downlink wireless networks,” in *Proc. IEEE INFOCOM*, 2010.
- [43] B. Sadiq, R. Madan, and A. Sampath, “Downlink scheduling for multi-class traffic in LTE,” *EURASIP J. Wirel. Commun. Netw.*, Aug 2009.
- [44] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, 2nd ed. Springer, 1998.
- [45] M. I. Freidlin and A. D. Wentzell, *Random Perturbations of Dynamical Systems*, 2nd ed. Springer, 1998.
- [46] S. Salas, E. Hille, and G. Etgen, *Calculus: One and Several Variables*, 8th ed. Wiley, 1998.
- [47] V. J. Venkataramanan and X. Lin, “On the large-deviations optimality of scheduling policies minimizing the drift of a Lyapunov function,” in *Proc. of Allerton Conf. on Comm. Control and Comp.*, Oct 2009.

- [48] A. L. Stolyar, “Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic,” *The Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.
- [49] M. Airy, “Multiple antenna wireless systems: Capacity and user performance limits,” Ph.D. dissertation, The University of Texas at Austin, May 2006.
- [50] M. Andrews, L. Qian, and A. Stolyar, “Optimal utility based multi-user throughput allocation subject to throughput constraints,” in *Proc. IEEE INFOCOM*, vol. 4, no. 13-17, Mar 2005, pp. 2415–2424.
- [51] V. A. Malyshev and M. V. Menshikov, “Ergodicity, continuity, and analyticity of countable markov chains,” *Transactions of Moscow Mathematical Society*, 1979.
- [52] J. G. Dai, “On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models,” *Annals of Applied Probability*, vol. 5, pp. 49–77, 1995.
- [53] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford, 2001.
- [54] A. Schrijver, *Combinatorial Optimization – Polyhedra and Efficiency*. Springer, 2003, vol. B.
- [55] V. Gupta and M. Harchol-Balter, “Self-adaptive admission control policies for resource-sharing systems,” in *Proc. of ACM SIGMETRICS Conf.*, 2009, pp. 311–322.
- [56] N. Bansal, H.-L. Chan, and K. Pruhs, “Speed scaling with an arbitrary power function,” in *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, 2009, pp. 693–701.

- [57] L. L. Andrew, A. Wierman, and A. Tang, “Optimal speed scaling under arbitrary power functions,” *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, pp. 39–41, 2009.
- [58] T. Gonzalez, “Minimizing the mean and maximum finishing time on uniform processors,” *Technical Report CS-78-22, The Pennsylvania State University*, 1978.
- [59] L. Massoulié and J. Roberts, “Arguments in favour of admission control for tcp flows,” in *Proc. of ITC*, 1999.
- [60] —, “Bandwidth sharing and admission control for elastic traffic,” *Telecommunication Systems*, vol. 15, no. 1-2, pp. 185–201, Nov. 2000.
- [61] S. Kittipiyakul and T. Javidi, “Delay-optimal server allocation in multi-queue multi-server systems with time-varying connectivities,” *IEEE Trans. Information Theory*, vol. 55, no. 5, pp. 2319–2333, May 2009.
- [62] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, “Scheduling in multi-channel wireless networks: rate function optimality in the small-buffer regime,” in *Proc. of ACM SIGMETRICS Conf.*, 2009, pp. 121–132.
- [63] R. Cogill and S. Lall, “Suboptimality bounds in stochastic control: A queueing example,” in *American Control Conference*, 14-16 2006, pp. 1642–1647.
- [64] S. Raj, E. Telatar, and D. Tse, “Job scheduling and multiple access,” in *Advances in Network Information Theory*, ser. DIMACS Series in Discrete Mathematics and Computer Science. American Mathematical Society, 2004.
- [65] C. Maglaras, “Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality,” *The Annals of Applied Probability*, vol. 10, no. 3, pp. 897–929, Aug. 2000.

- [66] I. M. Verloop and R. Nez-Queija, “Asymptotically optimal parallel resource assignment with interference,” *Queueing Systems*, vol. 65, no. 1, pp. 43–92, 2010.
- [67] A. Ganti, E. Modiano, and J. N. Tsitsiklis, “Optimal transmission scheduling in symmetric communication models with intermittent connectivity,” *IEEE Trans. Information Theory*, vol. 53, no. 3, pp. 998–1008, March 2007.
- [68] A. Tamir, “Least majorized elements and generalized polymatroids,” *Mathematics of Operations Research*, vol. 20, no. 3, pp. 583–589, 1995.
- [69] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. Cambridge, 2009.

## Vita

Bilal Sadiq received his degree of Bachelor of Science in Electronic Engineering from Ghulam Ishaq Khan Institute in May 2000, graduating at the top of his class. He was the recipient of Faculty Medal for Best Academic Performance in Electronic Engineering, and Ghulam Ishaq Khan Medal for Overall Best Academic Performance in All Faculties. From June through November of that year, he worked in Algorithms groups at Avaz Networks, Islamabad. He subsequently joined Reservoir Evaluation–Wireline group at Schlumberger and worked as a Field Engineer until 2003. In 2003, he entered graduate school at The University of Texas at Austin. In 2006, he started working under the supervision of Professor Gustavo de Veciana and became a member of the Wireless Networking and Communications Group. From July through December of 2008, he was a student intern at Qualcomm Flarion Technologies. He was admitted to PhD candidacy in 2009. He received the greatest joy of his life on June 3rd, 2008.

Permanent address: F-178, Street 8, Officers Colony  
Wah Cantt, 47010  
Pakistan

This dissertation was typeset with  $\text{\LaTeX}^\dagger$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{\TeX}$  Program.