Prajwal Osti

Optimal flow-level performance of opportunistic scheduling with size information

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 12.4.2011

Thesis supervisor:

Prof. Samuli Aalto

Thesis instructor:

D.Sc. (Tech.) Pasi Lassila

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Prajwal Osti

Title: Optimal flow-level performance of opportunistic scheduling with size information

Date: 12.4.2011 Language: English Number of pages:6+49

Department of Communication and Networking

Professorship: Teletraffic Theory Code: S-38

Supervisor: Prof. Samuli Aalto

Instructor: D.Sc. (Tech.) Pasi Lassila

Proportional fair (PF) is well known as the algorithm of choice for scheduling data flow in a wireless setting. Although considered as having good performance characteristics with nice fairness properties, PF is far from optimal when it comes to minimizing the flow-level delays. In this thesis, we study the optimal scheduling policy in an opportunistic wireless environment. The analysis can be applied to any setting where some kind of rate region can be constructed. As the main result we prove that when a fixed number of flows can be served at an average rate taken from a compact and symmetric rate region, the optimal policy is consistent with the SRPT-FM scheduling policy used in queueing systems with multiple servers. Moreover, we see that for some specially crafted rate region we can also get the explicit analytic expressions of the optimal long term service rates that minimize the cumulative delay of the flows. Through simulation, we observe that in a dynamic setting the performance of this static-case optimal policy is no worse than the PF policy in this kind of rate region. For a case with asymmetric service requirements, we consider a case with only two flows where we observe that if the rate region has certain characteristics met in practice, there are only two possible optimal rate pairs. The choice between them can be made based on the size of flows at hand. We also study a numerical example where the optimal policy still follows the spirit of the SRPT-FM policy even in the asymmetric case.

Keywords: Proportional fairness, optimality, opportunistic scheduling, SRPT, SRPT-FM, sized-based scheduling, flow-level performance

Preface

This thesis comes as the culmination of the work I have done since June 2010 at the Department of Communication and Networking, School of Electrical Engineering, Aalto university. I would like to thank the people who have helped me during this time to accomplish the work.

First of all I would like to thank my supervisor, Prof. Samuli Aalto for granting me the opportunity work under him. I owe him and my instructor, Dr. Pasi Lassila, a sincere debt of gratitude for their valuable guidance during the whole time. I am utterly grateful for their endless patience while discussing the ideas and for the scrutiny they provided while reviewing the drafts of the thesis.

I would also like to thank my colleagues at the department—Viktor Nässi for for first putting together and then helping me move my work space, Hanna Ropponen for taking care of all the official matters and Kimmo Pitkäniemi and Joni Laaksonen for the all important IT support they provided.

Although they are thousands of miles away, my family back home in Kathmandu, Nepal deserve my heartfelt gratitude for supporting and guiding me in every phase of my life. Lastly I would like to thank my friends Rajendra, Gautam, Manoj, Saurav, Deepak and Ujjwal for the valuable support and suggestions they have provided for the past one and a half years.

Otaniemi, 29.3.2011

Prajwal Osti

Contents

Abstract ii

Preface iii

Contents iv

Symbols and abbreviations vi

1 Introduction 1
- 1.1 Research problem formulation 2
- 1.2 Organization of thesis 3

2 Technological background 4
- 2.1 CDMA/EVDO 4
- 2.2 Beyond CDMA/EVDO 5

3 Theoretical background 7
- 3.1 Flow-level modeling 7
- 3.2 Elastic and streaming flows 7
- 3.3 Time scale separation 8
- 3.4 Opportunistic scheduling and rate region 8
- 3.5 $M/G/1$ queueing model and SRPT 9
- 3.6 Proportional fairness 10

4 Model 12
- 4.1 Static setting 13
  - 4.1.1 Asymmetric case 13
  - 4.1.2 Symmetric case 14
- 4.2 Dynamic setting 14

5 Asymmetric case with two jobs 16
- 5.1 Numerical example 24
- 5.2 More than two jobs in asymmetric case 26

6 Symmetric case with any number of jobs 27
- 6.1 One job 27
- 6.2 More than one job 27

7 Alpha-ball rate region 32

8 Simulation for the dynamic setting 34
- 8.1 Description of the simulator 34
- 8.2 Numerical results 35

9 Conclusions **39**

9.1 Summary of work accomplished 39

9.2 Future works 40

References **41**

Appendices **43**

A Mathematica Source Code **44**

B Two-user rate region **46**

Symbols and abbreviations

Symbols

$\nabla_n A$	n-dimensional gradient of A.
$\mathbf{P}\{A\}$	Probability of the event A.
$\mathbf{E}[X]$	Expected value of the random variable X.
$A \circ B$	Hadamard product of the vectors A and B.

Abbreviations

3GPP	Third Generation Partnership Project
AMC	Adaptive Modulation and Coding
AP	All Pole
ARQ	Automatic Repeat Request
CDMA	Code Division Multiple Access
CQI	Channel Quality Indicator
DRC	Data rate request channel
EVDO	Evolution Data Optimized
HDR	High Data Rate
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
IR	Incremental Redundancy
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
PF	Proportional Fairness
PS	Processor Sharing
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SC-FDE	Single Carrier- Frequency Domain Equalization
SNR	Signal-to-Noise Ratio
SRPT	Shortest Remaining Processing Time
SRPT-FM	Shortest Remaining Processing Time with the Fastest Machine
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access

1 Introduction

Although modern technological innovations have brought about drastic changes in almost all aspects of modern life, there are few areas where the impact has been as conspicuous as the field of personal communication. The "wireless revolution" that has occurred in the last decade or so has greatly impacted how we live our lives. The "always on" nature of this technology has connected people and devices to an extent that was previously unfathomable and we are just beginning to understand its true impact in a wider context.

Historically, the wireless medium was the first to be harnessed for the purpose of communication. However, for a very long time its use for personal communication had been very limited, mostly being favored for broadcast communication like the television and the radio. For personal day-to-day use by the general population the wired medium had been preferred—mainly because the wireless medium is noisier than the wired channel and therefore difficult to utilize effectively at a cost suitable for general use. But over the years the discoveries in different fields such as material science, information theory, signal processing, queueing theory, *etc.*, coupled with their implementation in modern digital circuitry have made the modern broadband wireless communication systems possible.

The wireless medium comes with its own set of challenges. Besides being more vulnerable to random degradations, its availability also is inherently limited. This means that arbitrarily increasing the spectrum when the demand is too high is not a viable option. It is imperative that the spectrum at hand be used in the best way possible to meet the ever-growing demand for capacity. Moreover, some minimum service requirement for all the users should *always* be provided to maintain its position as the ubiquitous medium. So, efficient management of radio resources at hand is one of the primary goals of the wireless networks in operation today.

Modern cellular wireless systems extensively reuse the spectrum they have, when ever they can, to provide service to as many users as possible. In these systems, a base station serves users in a fixed area with a fixed set of radio resources (frequency band, codes, *etc.*). The base stations in the nearby areas use a different set of resources to avoid interference. Different power control schemes are also used to further mitigate the effects of interference among the users in a cell.

Furthermore, there are different adaptive modulation and coding (AMC) schemes that are used to provide the best service to users with different channel conditions. This implies that the base station should 'know' the state of the channel of the users it is serving to. The base station generally gets this information through the feedback information (channel quality indicator) it receives from the users being served. In a fast fading environment, the channel state of a user is changing at a very high rate. The feedback should be provided at a rate faster than the rate the channel is changing for this strategy to work.

This gives rise to the very interesting issue of scheduling. If the base station has the perfect information of the channel states of the users, then it can cater to the one(s) who can best use it to make the optimal use of spectrum at hand. This approach is called opportunistic scheduling and it takes the advantage of the randomly varying nature of the wireless channel when there are many users trying to use it. The main drawback of this approach is that it may stifle the users that have bad channels (*e.g.*, the ones who are far away from the base station). Consequently, some fairer schemes like the proportionally fair (PF) scheduling are used to maintain the quality of service.

Moreover, from the classic results of queueing theory, we know that in many cases, scheduling the user with the smallest service requirement generally minimizes the average delay of all the users [18]. This approach alone, however, fails to exploit the opportunism present in the wireless channel. It is possible to combine both these previous approaches to get a good result [17]. In any case, all these methods imply that properly scheduling the users that are concurrently using or trying to use the channel can have a profoundly positive impact in the overall performance of the system.

1.1 Research problem formulation

Multiuser diversity presents a unique opportunity to harness the capacity of the wireless spectrum. Borst [4] introduces the opportunistic rate region for a static population, and for a dynamic population he provides a useful multi-class queue abstraction for the PF policy under the time-scale separation assumption. By switching very fast among the various users, a wireless scheduler can increase the aggregate rate in which it can provide service. Any consistent rule to allocate the channel, which is based on the instantaneous service rate supported by the user, gives a long term service rate to each user. This can be thought of as a point in the opportunistic capacity region [4]. Such rate regions also arise when there are multiple senders and receivers using the same channel [1] like a cellular mobile system with different base stations and mobile terminals.

Thus, we can essentially formulate the problem of most efficiently using a wireless channel as finding some rate vector in these rate regions that minimizes the average delay experienced by the users. Sadiq and de Veciana [17] introduce a new queueing model to analyze the scheduling in such opportunistic rate regions. They also give the results for optimal scheduling in the static environment for a size-aware scheduler when the opportunistic rate region is symmetric and approximated by a polymatroid. Simulation results when this principle is applied in a dynamic setting are also provided.

Size-oblivious policies such as the PF give a good performance in practical opportunistic environments [7]. However, the strong optimality results of the SRPT policy in the $M/G/1$ queues suggest that policies that exploit the size information of flows can give better results than the ones that ignore the size information. Moreover, when optimality is desired, the actual geometry of the rate region also plays a crucial role in deciding the best scheduling policy.

With these ideas in our mind we study the optimal scheduling policies in different kinds of rate regions and attempt to generalize the nature of these policies. Wherever possible, we also try to find the explicit expressions of such optimal policies. Moreover, since a case with random arrival of flows is different than the case which has a fixed number of flows, we also attempt to observe the effect of the application of the optimal policies in the static case to a setting the flows arrive dynamically.

To this end, we derive results for some cases with a static user population. As the main result we prove that the optimal scheduling policy in a symmetric rate region follows the SRPT-FM principle under certain nonrestrictive conditions. We then illustrate it with a numerical example. Furthermore, the optimal scheduling policy for an asymmetric rate region for two users is also analytically determined. For the case with dynamic flow arrivals, which is not mathematically tractable, we use simulation to study the effect of the application of the scheduling policies that are optimal in the static case. We see that these static-case optimal policies are no worse than the PF policy in the dynamic setting.

1.2 Organization of thesis

The thesis is organized in the following way: Section 2 provides some technological background for the work presented in this thesis. This is followed by Section 3 where we elaborate on the theoretical aspects of the ideas presented here. The general model we use for further analysis and simulation is described in Section 4. In Section 5, we prove a result when there are only two jobs present in an asymmetric setting. We investigate the case of a symmetric rate region and its properties in Section 6, and then we illustrate the ideas presented in this section with a numerical example in Section 7. Then, we simulate for the dynamically arriving flows in Section 8 by using the policy that is optimal for the static case. Finally we present the conclusions of our work in Section 9.

2 Technological background

The services like CDMA/EVDO provide very high data rate to its users. The main feature of this technology is that it tracks the channels of all the users in the system and based on the rates they can support, the scheduler sends data to them at an appropriate rate. We will discuss some parts of this technology that are pertinent to the work presented in this thesis.

2.1 CDMA/EVDO

CDMA/EVDO is based on the HDR system proposed in [2]. It uses a separate RF carrier of 1.25 MHz bandwidth to deliver data to its users. Coexisting with the CDMA2000 carrier and occupying a frequency band just beside it, CDMA/EVDO promises data rate from 38.4 kbps all the way up to 2.45 Mbps. The actual data rate mainly depends on the channel conditions of the users.

The forward link is a time division multiple access (TDMA) system (that is used for downlink by the user) which is divided into time slots of 1.67 ms (600 time slots in 1 second). The users are dynamically allocated one or more of these slots depending on what data rate they request. Although the standard does not specifically mention any particular scheme for the allocation of these time slots, Jalali *et al.* [7] suggest the PF [9] method as a good choice for a scheduling algorithm. PF scheduling gives acceptable performance and at the same time also has some nice fairness properties. It also allows the scheduler to take the advantage of the rapid temporal variation of the data rate supported by any user which has been termed *opportunistic scheduling* in the literature [13].

The variation of the supported data rate is the result of the various forms of degradation, like attenuation, multipath fading, shadowing, *etc*, suffered by the RF signal during its journey from the sender to the receiver. These random degradations affect the signal-to-noise ratio (SNR) of the channel and influence the supported data rate [20].

The forward link, in addition to the user data, contains the pilot signal and occasionally some other control information. Its simplified structure is shown in Figure 1. The pilot bursts (that occur twice every 1.67 ms even when there is no

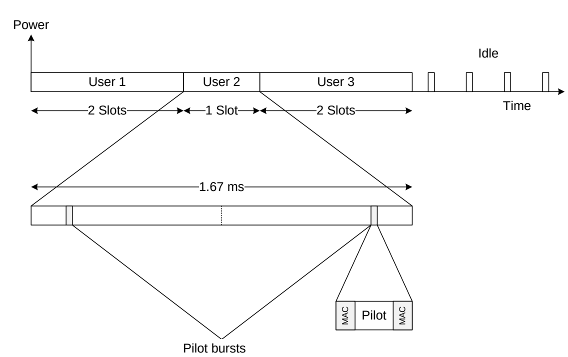

Figure 1: *Structure of an EVDO frame.* (Adapted from [2])

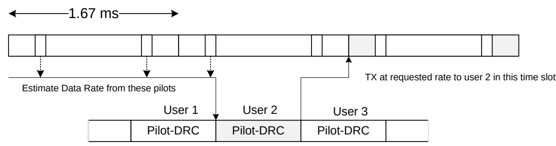

Figure 2: *An illustration of how DRC in reverse channel works. The pilot is tracked by all the users who estimate the channel data rate based on it. Then they request the data rate to the base station in the DRC.* (Adapted from [2])

data to send) are tracked by all the users of the system. These are sent at full power and are deterministic in nature. The mobile stations that track these pilot signals can estimate the SNR of the channel they wish to use by comparing what signal they actually receive with what they expect to receive. Again, the standard does not specify any particular method the mobile stations should use for this estimation. Methods based on All-Pole Autoregressive (AP) model of the channel and Linear prediction have been suggested [3].

The supported data rate is determined based on the estimated signal-to-noise ratio. The mobile stations send this information to the base station on the reverse link using the data rate request channel (DRC) using a 4 bit value (thus a total of 16 data rates can be supported). The DRC channel is code multiplexed among all the users in the system so that they can simultaneously update the base station with the data rate they can support. This is depicted in Figure 2.

If the PF algorithm is used, the user who gets the channel is chosen in the following way. The n users wishing to use the channel are uniquely indexed by the elements of the set $\mathcal{I}_n = \{1, \ldots, n\}$ and the user $k = \arg\max_{i \in \mathcal{I}_n} \frac{R_i(t)}{\bar{\theta}_i(t)}$ is chosen where $R_i(t)$ is the data rate requested by user i in the time slot beginning at t and $\bar{\theta}_i(t)$ is the average throughput of user i in some time window ending in t. The different data rates are achieved by using the various modulation and coding schemes that allocate variable time slots. A list of data rates the different modulation and coding schemes can achieve is shown in Table 1.

All the foregoing discussion assumes that the base station has the perfect knowledge of the data rate it should use to serve any particular user. Due to the random nature of channel this is rarely the case and therefore the CDMA/EVDO standard also uses Incremental Redundancy (IR) hybrid ARQ that ensures the correct data is received if the channel estimation process fails due to, *e.g.*, high interference from other base stations, or high speed of the user.

2.2 Beyond CDMA/EVDO

There are other technologies as well that offer high speed data transfer in wireless channel some of which are HSDPA, HSPA+, *etc.* They share some similarities with the EVDO system. The most noticeable difference, however, is unlike CDMA/EVDO they can serve multiple users at a time. The newest addition to these standards that has been deployed [19] is the Long Term Evolution (LTE) of the 3-rd Generation Partnership Project (3GPP) which, among other things, boasts data

DRC Index	No. of Slots	Data Rate [kbps]	Payload Size [bits]	Modulation
0x00	-	-	-	-
0x01	16	38.4	1024	QPSK
0x02	8	76.8	1024	QPSK
0x03	4	153.6	1024	QPSK
0x04	2	307.2	1024	QPSK
0x05	4	307.2	2048	QPSK
0x06	1	614.4	1024	QPSK
0x07	2	614.4	2048	QPSK
0x08	2	921.6	3072	8-PSK
0x09	1	1228.8	2048	QPSK
0x0a	2	1228.8	4096	16-QAM
0x0b	1	1843.2	3072	8-PSK
0x0c	1	2457.6	4096	16-QAM

Table 1: *Bit rates*

rates up to 100 Mbps for downlink and 50 Mbps for uplink. The next standard, named LTE Advanced promises data rates that are up to ten times greater than that of LTE for uplink as well as downlink and presents itself as the prime candidate of the 4-th generation technology. These new technologies are generally based on all-IP core (*i.e.*, no circuit switching even for voice) at the network layer and methods like wide channel OFDM, SC-FDMA, MIMO, *etc.* at the physical layer [6].

3 Theoretical background

In this section we will briefly introduce the theoretical aspects of the various concepts that are discussed in this thesis.

3.1 Flow-level modeling

In modern communication system like the Internet the data from the source terminal is transmitted to the destination terminal through a series of nodes in a network. The source data is broken down into a number of packets before being transmitted to the network. The network then essentially sees a series of independent data packets, which it delivers from one node to another until they reach the destination where they are finally reassembled. One way to evaluate the performance of such networks is to analyze the dynamics of these packets (*e.g.*, their size, arrival rate) at every node in them. The complexity of such analysis grows as the number of nodes in the network increases.

An alternative approach of analysis when the end-to-end data transmission is considered is to employ the flow-level models where only the dynamics of the arriving flows at the network (as opposed to the packets at nodes) is considered. These flows, then carry the whole file that has to be downloaded,*e.g.*, to a personal computer from a server. The network allocates its resources to these flows during the time of transmission of the files. The flow-level abstraction provides a good workaround for the complexity that may arise when the packet level models are used with a large number of nodes. In addition, this approach is also more conducive to analyzing the end user experience.

A flow-level model is said to be *static* if there are initially some flows to be transmitted but no new flows arrive once the transmission has begun. By contrast, in the *dynamic* setting new flows can arrive when the transmission has started. The arrivals are usually modeled by some random processes (*e.g.*, the Poisson process).

A number of flows may exist in a network that take data from the various sources to some destination. As more than one of them can exist simultaneously, the network resources (*e.g.*, bandwidth) may have to be shared among them in some way. This gives rise to the issue of queueing of the flows. Various queueing policies may be used by the network so that the network is used optimally while providing the best possible service. The choice of such queueing policies has a profound impact on the performance of the system and some of them perform better than others if some particular performance requirement is to be met.

3.2 Elastic and streaming flows

The flows generally encountered during data transmission are generally of two types — elastic and streaming [14]. The streaming flows carry the data such as voice and video that have a strict service rate requirement which is at least equal to the rate at which they are generated. Elastic flows, on the other hand, carry data that do not have any deadline for the arrival at the destination and therefore can be transmitted at any rate supported by the system. Data transfer through the TCP protocol correspond to such flows. Moreover, TCP also enables the bandwidth available in the network to be shared among the existing flows.

For elastic flows in the static case with n flows in the beginning, a useful metric for the measurement of performance is the cumulative delay, T, which is the sum of the time spent by all the flows in the system, *i.e.*,

$$T = \sum_{i=1}^{n} t_i, \tag{1}$$

where t_i is the time spent by the i-th flow in the system. In a dynamic setting, where the flows of random sizes arrive at random times for service, the delay of the i-th user, t_i, can be thought of as a realization of a random variable D. At steady state, assuming D has some stationary distribution, $\mathbf{E}[D]$ gives the average time a flow spends in the system.

3.3 Time scale separation

A scheduler in a time slotted wireless environment has to deal with the various time scales associated with the different times that are related to the various events (*e.g.*, arrival, departure, *etc.*) tied with the flows [12]. The largest time scale, τ_1 is that of arrivals and departures of flows. Typically a few flows arrive every second and after a certain delay they leave the system. Then there is the time scale of the rate process variation which corresponds to the time, τ_2, during which the rate process remains constant at some value. This is typically in some milliseconds in practical systems. Finally there is the time scale for the allocation of time slot, τ_3, which is the duration the time slot allocation algorithm takes to determine the appropriate user to serve. Its value is at most equal to the smallest allocable time slot. The time scale separation assumption means that $\tau_1 \gg \tau_2 \geq \tau_3$. That is, the allocation of the time slots to any user is done at a very high rate compared to the rate of arrival and departure of flows in the system. This assumption enables the application of the very useful processor sharing abstraction in some systems such as the one mentioned in Subsection 3.4.

3.4 Opportunistic scheduling and rate region

By serving a user[1] with the best channel conditions (and the highest supported data rate) a wireless base station can exploit multiuser diversity to provide more aggregate throughput. This process is called *opportunistic scheduling.* For example, if $R_i(t)$ is the stationary data rate process of the i-th user, the system can work at an aggregate rate of $\mathbf{E}[\max_i\{R_i(t)\}]$ when it caters to the user with the best channel conditions as opposed to $\mathbf{E}[R_i(t)]$ when it serves only the i-th user. Clearly,

$$\mathbf{E}\left[\max_i\{R_i(t)\}\right] \geq \mathbf{E}[R_i(t)].$$

[1]the words ‘user’ and ‘flow’ are used interchangeably

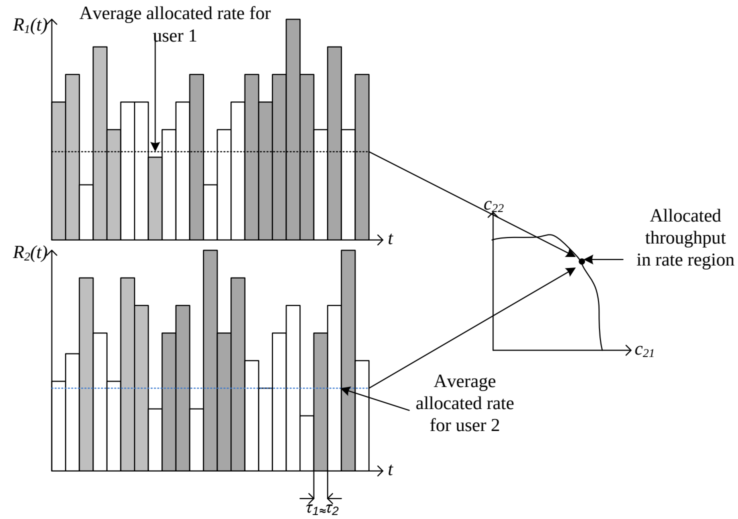

Figure 3: *Time scale separation and rate region. The shaded portion represents the time when that particular user is served by the base station.*

So the overall throughput of the base station is increased. When there are n users and the base station serves a flow based on its supported data rate, the long term throughputs received by each user (represented by a vector) can only come from some *rate region* $\mathcal{R} \subset \mathbb{R}_+^n$ [4]. Moreover, any non dominated rate vector from this rate region can be achieved by some *weight based strategy* augmented with a suitable tie-breaking rule [5]. For example, in Figure 3 we consider a case with two users, where the one with the highest supported rate is served at every time. This allows the two to be served at a long-term rate belonging to the opportunistic rate region. If some other weight based strategy is used (with a tie breaking rule), other non dominated points in the rate region can be achieved. The interior points of the rate region are achieved when the base station remains idle in some time slot without catering to any flow.

When a weight based strategy is implemented, a weight w_i is assigned to a user with rate process $R_i(t)$ and the user $i^*(t) = \arg\max_{i \in \{1,\ldots,n\}} w_i R_i(t)$ is chosen to transmit at time t. PF scheduling (see Subsection 3.6) then emerges as a special case of this weight based strategy that assigns $w_i = \frac{1}{c_i(t)}$ where $c_i(t)$ is the throughput of user i until time t. If we assume that the time scales are separated, it follows from the law of large numbers that $c_i(t)$ quickly converges to the average value $\bar{c}_i$ and a constant weight of $w_i = \frac{1}{\bar{c}_i}$ can be assigned to the user i.

3.5 $M/G/1$ queueing model and SRPT

The $M/G/1$ queue has Poisson arrival of flows at the rate λ. The sizes of these jobs, represented by the random variable X, are independent and distributed with some distribution function $F_X(x) = \mathbf{P}\{X \leq x\}$. The jobs are served by one server at a constant rate. It is a very well understood queueing model with multitude of applications [10, 11]. Besides the arrival rate and the job size distribution, the parameters related with the behavior of $M/G/1$ queue depend on the service policy of the server. The service policy determines the order in which the jobs are processed. The first in first out (FIFO) policy processes the jobs in the order of their arrival. The processor sharing (PS) policy divides the service rate equally among all the jobs present, serving all of them simultaneously. With the shortest remaining processing time (SRPT) policy, the server serves only the job in the queue that needs the least time to be processed. SRPT policy is optimal in the sense that it minimizes the average delays of the flows and by Little's law the mean queue length [18].

The $M/G/n$ queue is a system identical to $M/G/1$ except that it has more than one (*i.e.*, n) identical servers. The optimal policy for this case is unknown. However, SRPT (*i.e.*, the smaller jobs served before the larger ones) is still the optimal policy for the static case (*i.e.*, there no new arrivals after service begins) in the sense that it minimizes the cumulative delay. For heterogeneous servers (which serve at different rates) the policy that assigns the fastest machine to the shortest jobs initially present (SRPT-FM) achieves the least cumulative delay in the static case [16].

In opportunistic scheduling within some service rate region, some modified form of the $M/G/1$ can be used for the analysis of the system. Sadiq and de Veciana [17] propose such an $M/G/\mathcal{C}_n$ queue which allocates the service rate to jobs from some point on the opportunistic capacity region $\mathcal{C}_n$. Furthermore, when the rate can be chosen from the tightest polymatroid bounding the opportunistic capacity region they also prove that a policy that adheres to the SRPT-FM policy in principle minimizes the cumulative delay of jobs in the static case.

3.6 Proportional fairness

The issue of allocating the available bandwidth to the users in a *fair* manner is one of the primary design considerations of any modern wireless communication system. Fairness can be measured based on the utility of bandwidth to a user. Consider a system with n users indexed by the elements of $\mathcal{I}_n = \{1, \ldots, n\}$ to whom rates in form of a vector $(c_{n1}, \ldots, c_{nn}) \in \mathcal{C}_n \subset \mathbb{R}^n_+$ can be allocated. The fairest rate allocation, $(c^*_{n1}, \ldots, c^*_{nn})$, will maximize the aggregate utility to the users, *i.e.*,

$$(c^*_{n1}, \ldots, c^*_{nn}) = \underset{(c_{n1},\ldots,c_{nn})\in\mathcal{C}_n}{\arg\max} \sum_{i=1}^{n} U(c_{ni}), \tag{2}$$

where $U(c_{ni})$ is the utility of c_{ni} to user i. Utility functions are generally increasing, strictly concave and generally differentiable functions of $c_{ni} \geq 0$. Moreover they can be added together to get the aggregate utility. One such utility function that is applicable in a wide range of settings is the α-fair utility function [15] defined by

$$U(x) = \begin{cases} \frac{x^{1-\alpha}}{1-\alpha} & \text{if } \alpha \neq 1, \\ \log x & \text{if } \alpha = 1. \end{cases}$$

The different values of α corresponds to different fairness criteria, *e.g.*, $\alpha = 0$ maximizes the bandwidth used, $\alpha = \infty$ corresponds to the max-min fairness scheme, *etc.* In CDMA/EVDO system the PF criterion is generally used which corresponds to $\alpha = 1$. That is, the proportionally fair rate vector is achieved by maximizing the objective function

$$\sum_{i=1}^{n} \log c_{ni}.$$

It favors high throughput without stifling the flows with bad channels and can also be interpreted as follows [8]: For n flows, the rate vector $\boldsymbol{c} = (c_{ni}, i \in \mathcal{I}_n) \in \mathcal{C}_n$ is proportionally fair if for any other rate vector $\boldsymbol{c}' = (c'_{ni}, i \in \mathcal{I}_n) \in \mathcal{C}_n$, the aggregate proportional change is either zero or negative, *i.e.*,

$$\sum_{i=1}^{n} \frac{c'_{ni} - c_{ni}}{c_{ni}} \leq 0.$$

For the dynamic case, Borst [4] also points out that under the assumption of time scale separation (Section 3.3), the proportionally fair scheduler can be abstracted as a multi-class processor sharing system. In a time-slotted system with n flows, where only one user is served at a time, Stolyar [21] proves that the utility function described in (2) (the type II utility function) can be maximized (asymptotically) by using a gradient based rule which, in every time t, chooses the user $i^*(t)$ given by

$$i^*(t) = \arg\max_{i \in \mathcal{I}_n} [\nabla_n U(\hat{\boldsymbol{c}}(t))] \circ \boldsymbol{R}(t),$$

where $\circ$ denotes the *Hadamard product* of the vectors, the 'arg max' is taken over the components of this product and $\hat{\boldsymbol{c}}(t)$ is the estimate of the average service rate calculated by exponentially smoothing the data rates provided to the users, which is updated as

$$\hat{c}_i(t) = (1 - \beta)\hat{c}_i(t-1) + \beta R_i(t) \mathbb{1}_{\{i^*(t)\}}(i).$$

Here $\mathbb{1}_A(\cdot)$ is the indicator function of the set A, $R_i(t)$ is the data rate supported by user i at time t and vector $\boldsymbol{R}(t) := (R_i(t))$ for all $i \in \mathcal{I}_n$. Stolyar [21] also proves that the optimal solution is achieved regardless of the choice of initial $\hat{\boldsymbol{c}}(t)$. For the proportional fair case, the condition for optimality can be simplified as,

$$i^*(t) = \arg\max_{i \in \mathcal{I}_n} \frac{R_i(t)}{\hat{c}_i(t)},$$

where $c_i(t)$ is the i-th element of the vector $\hat{\boldsymbol{c}}(t)$ which is the exponentially smoothed service (*i.e.*, average) rate of user i.

This essentially means that in a time slotted system proportional fairness can be achieved by allocating the time slot to the user with the highest supported rate compared to the achieved throughput, which, in a symmetric case, translates to always serving the user with the highest instantaneous rate [4].

4 Model

We consider the downlink data transmission from a base station to a set of n users that are uniquely indexed by the elements of the set $\mathcal{I}_n = \{1, \ldots, n\}$. The data is in the form of a file whose size is known at the beginning and is carried through elastic flows to the users. When there are n users, the i-th user is served at a *long-term rate* with the i-th component of the vector $\boldsymbol{c}_n = \{c_{n1}, \ldots, c_{nn}\} \in \mathcal{C}_n \subset \mathbb{R}^n_+$ where $\mathcal{C}_n$ is called the rate region. The term 'long-term rate' is used to refer to the average service rate offered to a flow when the number of flows is fixed. This means whenever a file transfer ends, the flow departs and the rate region changes (loses one dimension) and consequently the long-term average service rate also changes in general from that point onwards. The same principle applies when new flows arrive in the dynamic case. These rate regions, which are referred to by other names like 'capacity region', 'opportunistic capacity region', *etc.* in the literature arise as a consequence of opportunistic scheduling, multiuser interference [1], *etc.* One such rate region that arises as a result of opportunistic scheduling is elaborated in this section which is developed by Borst [4].

Consider a system whose users are served one at a time by the base station in small fixed-length time slots. The data rate supported by user $i \in \mathcal{I}_n$ in time slot beginning at t is given by some stationary and ergodic discrete-time stochastic process $R_i(t)$ which the base station is aware of by some feedback information. With this information at hand, the scheduler in the base station[2] selects the user it wants to transmit the data to based on some rule that is a function of the data rate supported by the users at that time. The transfer is then done at the rate supported by the selected user. The situation is similar to the CDMA/EVDO system discussed in Section 2. Additionally, we also assume that when the data transfer for a user is complete, the base station stops keeping track of the rate associated with that user and after that does not provide any more time slot to it. We also assume that the base station always has enough data in its buffer to transmit the amount of bits it wants to transmit (non-idling system) until the data transfer for a user is completed.

Let the random vector $(R_1, \ldots, R_n)$ have the same distribution as the joint stationary distribution of the allocable rates. When $(R_1, \ldots, R_n)$ takes discrete values in a finite set $\mathcal{R}_n \subset \mathbb{R}^n$, $p(\boldsymbol{r})$ is the stationary probability that the value of supported rate vector is $\boldsymbol{r} = (r_1, \ldots, r_n) \in \mathcal{R}_n$. The long-term supported data rate of the user i is represented by c_{ni} when there are n users in the system. The set $\mathcal{C}_n \in \mathbb{R}^n$, which is the set of all achievable long term data rate vectors, is given by

$$\mathcal{C}_n = \{T \in \mathbb{R}^n_+ : z(T) \geq 1\},$$

[2]The words 'scheduler' and 'base station' are used interchangeably from here onwards.

where $z(T)$ is the optimal value of the linear program

$$\begin{aligned}
&\max z,\\
&\text{sub } z \le z_i = \sum_{\boldsymbol{r}\in\mathcal{R}_n} \frac{p(\boldsymbol{r})x_i(\boldsymbol{r})r_i}{T_i}, \quad i \in \mathcal{I}_n,\\
&\sum_{i=1}^{n} x_i(\boldsymbol{r}) \le 1, \quad \boldsymbol{r} \in \mathcal{R}_n,\\
&x_i(\boldsymbol{r}) \ge 0, \quad i \in \mathcal{I}_n.
\end{aligned}$$

$\mathcal{C}_n$ is called the *opportunistic rate region.* The rate region is *symmetric* when the data rates supported by all the users have identical marginal distributions. Otherwise, the rate region is said to be *asymmetric*.

Furthermore, from the complementary slackness condition of the above linear program we see that if we assign the weight w_i to the flow $i \in \mathcal{I}_n$ and allocate the time slot beginning at t to the user $i^* = \arg\max_{i\in\mathcal{I}_n} w_i R_i(t)$ (when i^* is not unique, the selection is done by some well defined tie-breaking rule among such users), by appropriately choosing these weights w_i, any non-dominated rate vector in the rate region as the long-term service rate can be achieved. One such rate region for a simple case of two users is discussed in detail in Appendix B.

4.1 Static setting

In this setting, no new flows arrive once the service has begun. The rate region for the static setting can either be symmetric or asymmetric.

4.1.1 Asymmetric case

For the *asymmetric* case, we consider a two-user system that has flows of sizes l_1 and l_2 and where they are being served by a single base station. It is assumed that the scheduler knows the sizes of the flows and based on that information, it allocates the service rates $\boldsymbol{c} = (c_{21}, c_{22}) \in \mathcal{C}_2$ to them. When one of the flows leaves after its service requirement is fulfilled, the one that remains is then served at the highest possible rate that can be allocated to it. Let us assume those rates to be a_1 for flow 1 and a_2 for flow 2. A service policy $\vec{\pi}$ is defined as $\vec{\pi} = (\boldsymbol{c}, a), \boldsymbol{c} \in \mathcal{C}_2, a \in \{a_1, a_2\}$. Let, Π be a set of all such service policies. If $t_1^{\vec{\pi}}$ and $t_2^{\vec{\pi}}$ be the total times spent by the flows 1 and 2 in the system under policy $\vec{\pi}$, then the total delay is defined as,

$$T^{\vec{\pi}} = t_1^{\vec{\pi}} + t_2^{\vec{\pi}}.$$

The optimal policy $\vec{\pi}^*$ is then defined as the policy that minimizes the cumulative delay of the two flows, *i.e.*,

$$T^{\vec{\pi}^*} = \min_{\vec{\pi}\in\Pi} T^{\vec{\pi}}. \tag{3}$$

We want to find the allocation of the service rates to the flows in form of the optimal policy $\vec{\pi}^* = (\boldsymbol{c}^*, a^*)$, which are possibly based on the sizes of the jobs, that would minimize the cumulative delay. Moreover, the problem essentially reduces to finding the rate pair $\boldsymbol{c}^*$ as the remaining a^* can be one of two values depending on which job leaves the system first. Note that in a time slotted system such optimal long term rates can be assigned to a pair of flows by some weight based strategy augmented with a tie breaking rule as discussed in Section 4.

4.1.2 Symmetric case

We consider a system with n jobs at the beginning that are indexed by the elements of the index set, $\mathcal{I}_n = \{1, \ldots, n\}$, in the decreasing order of their sizes, *i.e.*, $s_{ni}, i \in \mathcal{I}_n$, is the size of the i-th largest job when there are n jobs. The sizes are measured in time units required to process the job at some constant rate. The *scheduler* can assign a *service rate vector* $\boldsymbol{c}_n = (c_{n1}, \ldots, c_{nn}) \in \mathcal{C}_n$ to process the i-th job with rate $c_{ni},\ i \in \mathcal{I}_n$. Here, $\mathcal{C}_n \subset \mathbb{R}^n_+$ is the abstract *rate region*. This rate region can arise as a result of opportunistic scheduling as discussed in Section 4, although there can be other causes for its existence [1]. The processing begins at $t = 0$ and as the system evolves, at a certain point of time one of the jobs is completed which leaves what remains of the unfinished $n-1$ jobs to be processed. A new service rate vector, $\boldsymbol{c}_{n-1}$ is then chosen from the rate region $\mathcal{C}_{n-1} \subset \mathbb{R}^{n-1}_+$ to process these remaining jobs. The remaining jobs are then reindexed according to their sizes, *i.e.*, if $s_{n-1,1}, \ldots\ s_{n-1,n-1}$ be the sizes of the remaining jobs, $s_{n-1,i} \geq s_{n-1,j}$ for all $i < j,\ i,j \in \mathcal{I}_{n-1}$. This process continues until all the jobs are completed. It is assumed that while choosing the rate vector, the scheduler is always aware of the remaining sizes of the jobs and the rate region.

A *service policy* $\vec{\pi} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n)$ is defined as a sequence of service rate vectors $\boldsymbol{c}_k \in \mathcal{C}_k,\ k \in \mathcal{I}_n$. Here $\boldsymbol{c}_k$ is the rate vector the scheduler assigns from the rate region when there are k jobs in the system (called phase k from here onwards). Let Π_n be the set of all such policies,

$$\Pi_n = \{\ \vec{\pi} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n) : \boldsymbol{c}_k \in \mathcal{C}_k,\ k \in \mathcal{I}_n\}.$$

Let $t_i^{\vec{\pi}}$ be the time when the job with original size s_{ni} completes under policy $\vec{\pi}$. The *cumulative delay* is defined as

$$T^{\vec{\pi}} = \sum_{i=1}^{n} t_i^{\vec{\pi}}.$$

If $\vec{\pi}^*$ be a policy that minimizes the cumulative delay then,

$$T^{\vec{\pi}^*} = \min_{\vec{\pi} \in \Pi_n} T^{\vec{\pi}}.$$

The sequence of service rate vectors that give the optimal policy $\vec{\pi}^*$ are denoted by $(\boldsymbol{c}_1^*, \ldots, \boldsymbol{c}_n^*)$. The optimization problem is then to find this optimal policy $\vec{\pi}^*$.

4.2 Dynamic setting

Moreover, in the dynamic setting, which we study exclusively through simulation, we allow for the arrival of new flows. The flows are assumed to arrive according to the Poisson process of intensity λ [per second]. The file sizes associated with the flows are assumed to be independent and have the exponential distribution with a mean $1/\mu$ [bits]. The scheduler is fully aware of the size of all the arriving flows, and the remaining size of the ones being processed. Once again the time scales are assumed to be separated (Subsection 3.3). After a new flow arrives, a new rate vector is $\boldsymbol{c}$ chosen from the symmetric rate region $\mathcal{C}_n$ where n is the total number of flows in the system after the arrival of the new flow. Let D denote the delay experienced by a typical flow. When the queue has stabilized D is assumed to have a stationary distribution and its average value $\mathbf{E}[D]$ gives the mean delay of the flows.

5 Asymmetric case with two jobs

Asymmetric rate regions may arise when the different flows a base station is serving have different marginal distributions of the achievable rate process. The developments of Section 4 are still valid and we get a rate region from where the base station can assign a long term rate vector. The analysis, however, is more complex than for a symmetric case which is discussed in Section 6. We demonstrate such an analysis for a simple case of two jobs that have non-identical distribution of the stationary rate process. If certain conditions are imposed on the geometry of the rate region in such a case, we will see that we can get optimal vectors in the rate region that minimize the cumulative delay. Under these conditions, the optimal rates are easily characterized and the selection of these rate pairs can be done on the basis of a simple criterion that is based on the initial sizes of the flows.

We recall the model introduced in Section 4.1.1 that has the flows with non-identical rate processes. The rates are allocated from a convex region $\mathcal{C}_2$ bounded by the co-ordinate axes and the curve $f(x)$ with the following properties:

- $f(x)$ is decreasing and concave in the interval $0 \leq x \leq a_1$, *i.e.*, $f'(x) \leq 0$ and $f''(x) \leq 0$ for all $x \in [0, a_1]$, if they exist at all,
- $f(0) = a_2, f(a_1) = 0$, *i.e.*, one job can be served at the maximum rate of a_1 and the other job at a maximum rate of a_2.

An example of such rate region is shown in Figure 4.

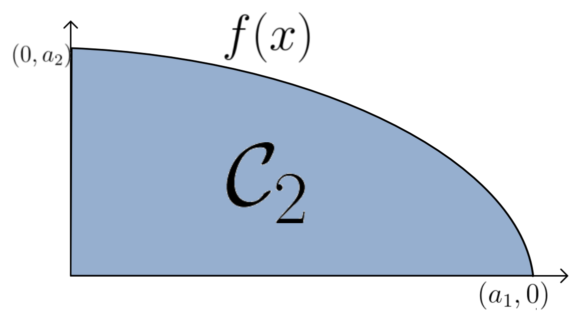

Figure 4: *An example of rate region.*

We assume that the server is able to allocate any rate pair from this region. Once the job of one of the users is complete, the one that remains is served either at the rate a_1 or a_2, depending on which job leaves the system first.

Proposition 1. *If the bounding curve $f(x)$ of the capacity region has the properties mentioned above there exists a pair of optimal points at the boundary curve of the rate region $(x_1^*, f(x_1^*))$ and $(x_2^*, f(x_2^*))$ [3] which give the minimum cumulative delay depending on the sizes of l_1 and l_2 where l_1 is the initial size of the job that is*

[3] (As only two dimensions are involved in this case we resort to a simpler notation of $(x, f(x))$ instead of (c_{21}, c_{22}) which would have been the more consistent choice for the rate pair.)

served at the rate $x \leq a_1$ and l_2 is the initial size of the job that is served at the rate $f(x) \leq a_2$. More specifically, if $\frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)} \leq \frac{l_2}{l_1}$, then $(x_1^*, f(x_1^*))$ is the optimal rate pair, otherwise $(x_2^*, f(x_2^*))$ is the optimal rate pair.

Proof. Clearly, to minimize the aggregate delay the server must serve the users at the highest possible rate which means that the service rate is somewhere in the border curve of the capacity region. Thus, we limit ourselves to finding a rate pair $\{(x, f(x)) : 0 \leq x \leq a_1\}$ that lies on the border curve and that minimizes the total delay. To that end, we define the service policy x that allocates the service rates x and $f(x)$ to users 1 and 2 respectively until one of the jobs departs. The remaining job is then served at the rate a_1 or a_2 from that point onwards. Let $t_1(x)$ and $t_2(x)$ be the times taken to complete the jobs when they are *always* served with the rates x and $f(x)$ respectively, *i.e.*,

$$t_1(x) = \frac{l_1}{x} \quad \text{and}$$
$$t_2(x) = \frac{l_2}{f(x)}$$

We then define two cost functions (that represent the cumulative delay of the jobs) as

$$h_1(x) = \frac{2l_1}{x} + \left(l_2 - \frac{l_1 f(x)}{x}\right) \cdot \frac{1}{a_2} \quad \text{and}$$
$$h_2(x) = \frac{2l_2}{f(x)} + \left(l_1 - \frac{l_2 x}{f(x)}\right) \cdot \frac{1}{a_1},$$

which can be rearranged as

$$h_1(x) = \frac{l_1}{a_2}\left(\frac{2a_2 - f(x)}{x}\right) + \frac{l_2}{a_2} \quad \text{and} \tag{4}$$
$$h_2(x) = \frac{l_2}{a_1}\left(\frac{2a_1 - x}{f(x)}\right) + \frac{l_1}{a_1}. \tag{5}$$

Depending on which job leaves the system first, we use the appropriate cost function, *i.e.*,

$$h(x) = \begin{cases} h_1(x) & \text{if job 1 leaves first, } \textit{i.e.}\text{, if } t_1(x) < t_2(x) \iff f(x) < \frac{l_2}{l_1}x, \\ h_2(x) & \text{if job 2 leaves first, } \textit{i.e.}\text{, if } t_1(x) > t_2(x) \iff f(x) > \frac{l_2}{l_1}x. \end{cases}$$

For any pair of policies $x_1, x_2 \in [0, a_1]$,

$$
\begin{aligned}
& h_1(x_1; l_1, l_2) \le h_2(x_2; l_1, l_2) \\
\Leftrightarrow\ & \frac{l_1}{a_2}\left(\frac{2a_2 - f(x_1)}{x_1}\right) + \frac{l_2}{a_2} \le \frac{l_2}{a_1}\left(\frac{2a_1 - x_2}{f(x_2)}\right) + \frac{l_1}{a_1} \\
\Leftrightarrow\ & l_1\left(\frac{2}{x_1} - \frac{f(x_1)}{a_2 x_1} - \frac{1}{a_1}\right) \le l_2\left(\frac{2}{f(x_2)} - \frac{x_2}{a_1 f(x_2)} - \frac{1}{a_2}\right) \\
\Leftrightarrow\ & \frac{l_2}{l_1} \ge \frac{\left(\frac{2}{x_1} - \frac{f(x_1)}{a_2 x_1} - \frac{1}{a_1}\right)}{\left(\frac{2}{f(x_2)} - \frac{x_2}{a_1 f(x_2)} - \frac{1}{a_2}\right)} \\
\Leftrightarrow\ & \frac{l_2}{l_1} \ge \frac{f(x_2)\left(2 - \frac{x_1}{a_1} - \frac{f(x_1)}{a_2}\right)}{x_1\left(2 - \frac{x_2}{a_1} - \frac{f(x_2)}{a_2}\right)}
\end{aligned}
$$

i.e.,

$$
h_1(x_1; l_1, l_2) \le h_2(x_2; l_1, l_2) \iff \frac{l_2}{l_1} \ge \frac{f(x_2)\left(2 - \frac{x_1}{a_1} - \frac{f(x_1)}{a_2}\right)}{x_1\left(2 - \frac{x_2}{a_1} - \frac{f(x_2)}{a_2}\right)} \tag{6}
$$

Now, we introduce two auxiliary functions $g_1(x)$ and $g_2(x)$ defined as,

$$
\begin{aligned}
g_1(x) &= \frac{2a_2 - f(x)}{x} \text{ for } x \in [0, a_1] \quad \text{and} \\
g_2(x) &= \frac{2a_1 - x}{f(x)} \text{ for } x \in [0, a_1].
\end{aligned}
$$

The functions $g_1(x)$ and $g_2(x)$ can be used for further analysis instead of the actual cost functions $h_1(x)$ and $h_2(x)$ as these two sets of functions (cost and auxiliary) are related to each other by simple linear transformations. The important advantage these auxiliary functions offer is that they are independent of the job sizes. Let x_1^* and x_2^* be the points where the auxiliary functions are minimized, *i.e.*,

$$
\begin{aligned}
g_1(x_1^*) &= \min_{x \in [0, a_1]} g_1(x) \quad \text{and} \\
g_2(x_2^*) &= \min_{x \in [0, a_1]} g_2(x).
\end{aligned}
$$

And consequently,

$$
\begin{aligned}
h_1(x_1^*) &= \min_{x \in [0, a_1]} h_1(x) \text{ for any } l_1, l_2 \quad \text{and} \\
h_2(x_2^*) &= \min_{x \in [0, a_1]} h_2(x) \text{ for any } l_1, l_2.
\end{aligned}
$$

Consider the auxiliary cost function $g_1(x)$. If the cost is set to a constant value k, the set of rate pairs where this cost is achieved, L, is

$$
L = \{(x, y) : y = 2a_2 - kx\}.
$$

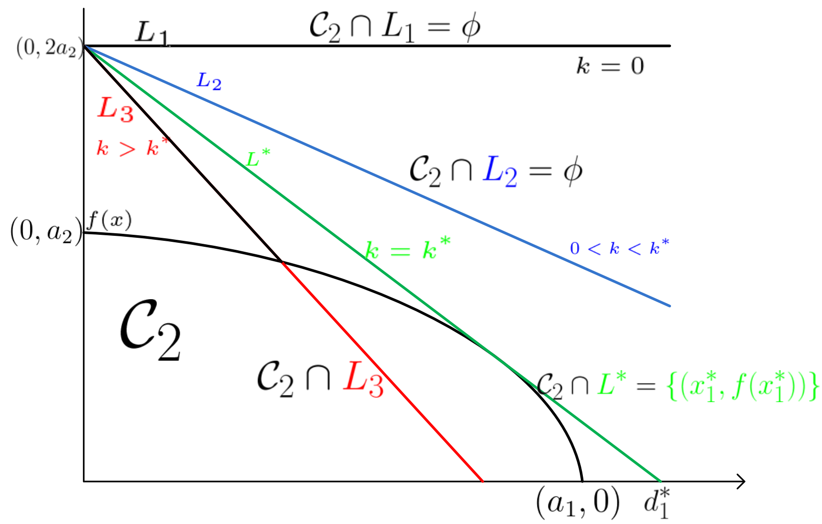

Figure 5: *An illustration of how the auxiliary cost function $g_1(x)$ is related to the our rate region. L_1 represents the case when the cost is zero. Clearly there are no rate pairs that give this cost. When the cost is slowly increased, until the cost becomes k^*, there are no feasible rate pairs that give these value of cost (represented by the blue line in the figure). When the line drawn from $(0, 2a_2)$ is tangent to the curve $f(x)$ (the green line, L^*, in the figure), the rate pair $(x_1^*, f(x_1^*))$ gives the unique minimum possible cost. On further increasing the cost we get a set of points (*i.e., *the red portion of the line L_3) in the rate region which give that particular value of cost*

Since all the rate pairs in L do not fall in our rate region, $\mathcal{C}_2$, the set of all *feasible* rate pairs with a constant cost $g_1(x) = k$ is $\mathcal{C}_2 \cap L$. The cost would be minimized if k is taken as small as possible, provided $\mathcal{C}_2 \cap L$ is not an empty set. We can clearly see that the elements of L are the points of a straight line that has a slope $-k$ and that passes through the point $(0, 2a_2)$. To minimize the cost, we have to take k as small as possible. However, if we decrease the cost k indefinitely, there would be no feasible rate pairs. Thus, the minimum cost is achieved when the line[4] L is a tangent[5] to the curve $f(x)$ and optimal rate pair, $(x_1^*, f(x_1^*))$, is the point of intersection of L and $f(x)$ where the optimal cost is k^*. By further increasing the cost, we get several feasible points where the cost is constant. By similar reasoning we get the optimal point $(x_2^*, f(x_2^*))$, which is the point of intersection of the tangent drawn to the curve $f(x)$ from the point $(2a_1, 0)$. This is further illustrated in Figure 5.

Now, let d_1 and d_2 be the intercepts of the tangents drawn to the border curve $f(x)$ at the points $(x_1, f(x_1))$ and $(x_2, f(x_2)))$ respectively on the x-axis. Since $f(x)$ is decreasing and concave, $d_1 \leq d_2$ if and only if $x_1 \geq x_2$ (see Figure 6 for further illustration).

The x-intercept d_1^* of the tangent to the border curve at point $(x_1^*, f(x_1^*))$ (*i.e.,*

[4] L has been referred to both as a line and as a set from this point onwards.

[5] Strictly speaking L is a tangent only when $f(x)$ is smooth, *i.e.*, has continuous first derivatives. In general, the minimum cost is given by that L which has the *greatest* slope and has at least one point common with $\mathcal{C}_2$. The optimal point(s) then can be found in the set $L \cap \mathcal{C}_2$.

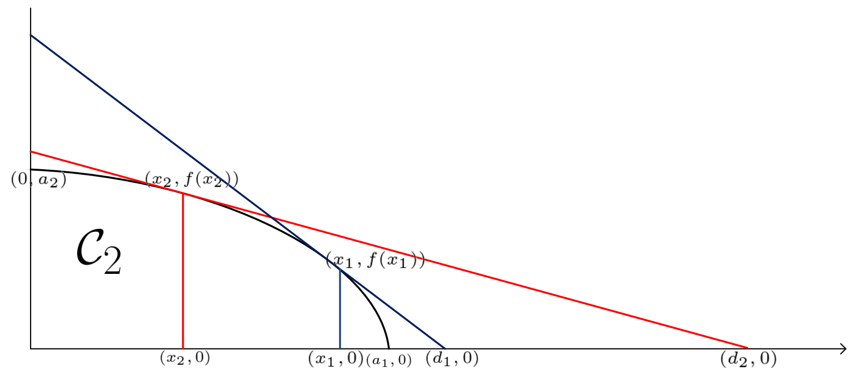

Figure 6: *A graphical proof that for our border curve* $d_1 \leq d_2$ *if and only if* $x_1 \geq x_2$

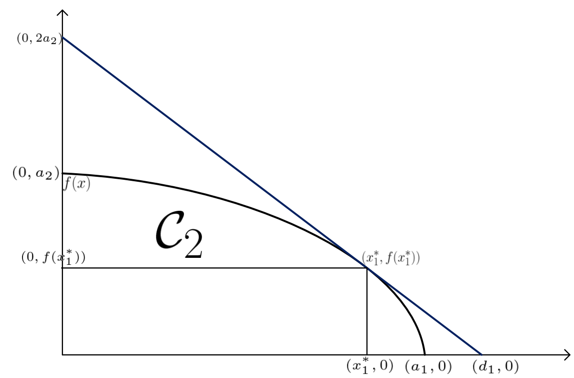

Figure 7: *Sketch to prove that* $d_1^* = \frac{2a_2 x_1^*}{2a_2 - y_1^*}$

the line L_1) on the x-axis is given by (see Figure 7)

$$\begin{aligned} d_1^* &= \frac{2a_2 x_1^*}{2a_2 - y_1^*} \\ &\leq \frac{2a_2 a_1}{2a_2 - y_1^*} \\ &\leq \frac{2a_2 a_1}{2a_2 - a_2} \\ &= 2a_1 \\ \Leftrightarrow d_1^* &\leq 2a_1. \end{aligned} \tag{7}$$

Since, $2a_1$ is the x-intercept of the tangent drawn to the point $(x_2^*, f(x_2^*))$, from (7) we can conclude that, $x_1^* \geq x_2^*$ and therefore $f(x_1^*) \leq f(x_2^*)$. From Figure 8 we observe that for $x \geq x_1^*$, $-g_1(x) = -\frac{2a_2 - f(x)}{x}$, which is the slope of the line joining the point $(0, 2a_2)$ and the point $(x, f(x))$, is decreasing with increasing x, implying that the cost is monotonically increasing. Similarly $-g_1(x)$ monotonically decreases when x is steadily decreased from x^*. So, we can conclude that

- $g_1(x)$ and hence $h_1(x)$ is decreasing for all $x \leq x_1^*$ and

- $g_1(x)$ and hence $h_1(x)$ is increasing for all $x \geq x_1^*$

Similarly, by taking auxiliary function $g_2(x)$ we can conclude the following.

- $g_2(x)$ and hence $h_2(x)$ is decreasing for all $x \leq x_2^*$ and
- $g_2(x)$ and hence $h_2(x)$ is increasing for all $x \geq x_2^*$

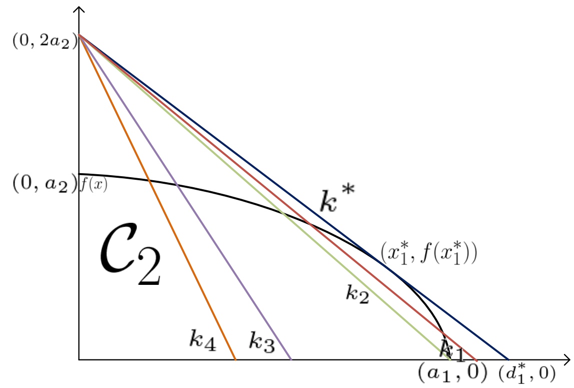

Figure 8: *A demonstration of how the cost function $g_1(x)$ is changing. Since the cost function obtains its minimum value, k^* at $(x_1^*, f(x_1^*))$ when the rate allocation is changed from this value the cost increases. The figure shows that $g_1(x)$ is increasing for $x \geq x_1^*$ and it is decreasing for $x \leq x_1^*$.*

As $(x_2^*, f(x_2^*))$ minimizes $g_2(x)$,

$$
\begin{aligned}
& g_2(x_2^*) \leq g_2(x_1^*) \\
\Leftrightarrow\ & \frac{2a_1 - x_2^*}{f(x_2^*)} \leq \frac{2a_1 - x_1^*}{f(x_1^*)} \\
\Leftrightarrow\ & \frac{2 - \frac{x_2^*}{a_1}}{f(x_2^*)} \leq \frac{2 - \frac{x_1^*}{a_1}}{f(x_1^*)} \\
\Leftrightarrow\ & f(x_1^*)\left(2 - \frac{x_2^*}{a_1}\right) \leq f(x_2^*)\left(2 - \frac{x_1^*}{a_1}\right) \\
\Leftrightarrow\ & f(x_1^*)\left(2 - \frac{x_2^*}{a_1}\right) - \frac{f(x_1^*)f(x_2^*)}{a_2} \leq f(x_2^*)\left(2 - \frac{x_1^*}{a_1}\right) - \frac{f(x_1^*)f(x_2^*)}{a_2} \\
\Leftrightarrow\ & \frac{f(x_1^*)}{f(x_2^*)} \leq \frac{2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}}{2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}} \\
\Leftrightarrow\ & \frac{f(x_1^*)}{x_1^*} \leq \frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)}.
\end{aligned}
\tag{8}
$$

As $(x_1^*, f(x_1^*))$ minimizes $g_1(x)$,

$$
\begin{aligned}
& g_1(x_2^*) \geq g_1(x_1^*) \\
& \Leftrightarrow \frac{2a_2 - f(x_2^*)}{x_2^*} \geq \frac{2a_2 - f(x_1^*)}{x_1^*} \\
& \Leftrightarrow \frac{2 - \frac{f(x_2^*)}{a_2}}{x_2^*} \geq \frac{2 - \frac{f(x_1^*)}{a_2}}{x_1^*} \\
& \Leftrightarrow x_1^* \left(2 - \frac{f(x_2^*)}{a_2}\right) \geq x_2^* \left(2 - \frac{f(x_1^*)}{a_2}\right) \\
& \Leftrightarrow x_1^* \left(2 - \frac{f(x_2^*)}{a_2}\right) - \frac{x_1^* x_2^*}{a_1} \geq x_2^* \left(2 - \frac{f(x_1^*)}{a_2}\right) - \frac{x_1^* x_2^*}{a_2} \\
& \Leftrightarrow \frac{x_1^*}{x_2^*} \geq \frac{2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}}{2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}} \\
& \Leftrightarrow \frac{f(x_2^*)}{x_2^*} \geq \frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)}.
\end{aligned} \tag{9}
$$

Thus from (8) and (9), we get,

$$
\frac{f(x_1^*)}{x_1^*} \leq \frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)} \leq \frac{f(x_2^*)}{x_2^*}. \tag{10}
$$

So if,

$$
\frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)} \leq \frac{l_2}{l_1} \leq \frac{f(x_2^*)}{x_2^*}
$$

then policy x_1^* is optimal and

$$
\min_{x \in [0, a_1]} h(x) = h_1(x_1^*).
$$

Similarly, if

$$
\frac{l_2}{l_1} \geq \frac{f(x_2^*)}{x_2^*},
$$

as a consequence of (6) and (10), $h_1(x_1^*) \leq h_2(x_2^*)$, and therefore x_1^* is still optimal implying

$$
\min_{x \in [0, a_1]} h(x) = h_1(x_1^*).
$$

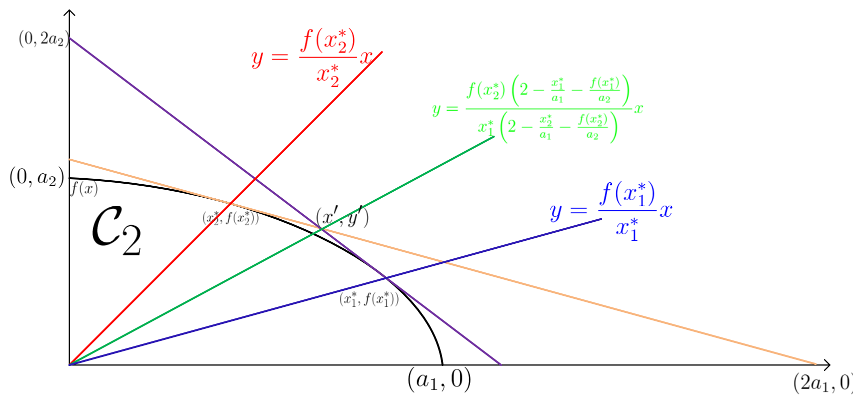

Figure 9: *Figure to demonstrate* (10). *The three sides of* (10) *are the slopes of the blue, the green and the red lines respectively.*

Thus we can see that x_1^* is optimal if and only if $\frac{l_2}{l_1} \geq \frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$. On the other hand, if

$$\frac{l_2}{l_1} \leq \frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)},$$

then we can similarly prove that the policy x_2^* is optimal and

$$\min_{x \in [0, a_1]} h(x) = h_2(x_2^*).$$

$\square$

Remark 2. From Proposition 1 we can see that if there are only two jobs, there can be only two possible rate pairs that give the minimum cumulative delay, irrespective of the sizes of the jobs to be processed, when the rate region has aforementioned properties. The choice of the *optimal rate pair* between these two pairs for any two jobs then depends on the relative sizes of the jobs at hand and the quantity $\frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$. Once the service has commenced, the service rate remains constant until one of the jobs is completed, after which the remaining job is processed at the highest rate possible for it. This implies that the job sizes decrease at constant rate depending on their position in the job-size space (see Figure 10).

Remark 3. The optimal rate pairs not only exist but can also be determined by a simple procedure as follows. The optimal rate pair $(x_1^*, f(x_1^*))$ is given by the point of intersection of the tangent to the curve $f(x)$ from the point $(0, 2a_2)$, where a_2 is the highest possible processing rate for job 2. Similarly, the rate pair $(x_2^*, f(x_2^*))$ is given by the point of intersection of the tangent to the curve $f(x)$ from the point $(0, 2a_1)$, where a_1 is the highest possible processing rate for job 1. This is shown in Figure 9.

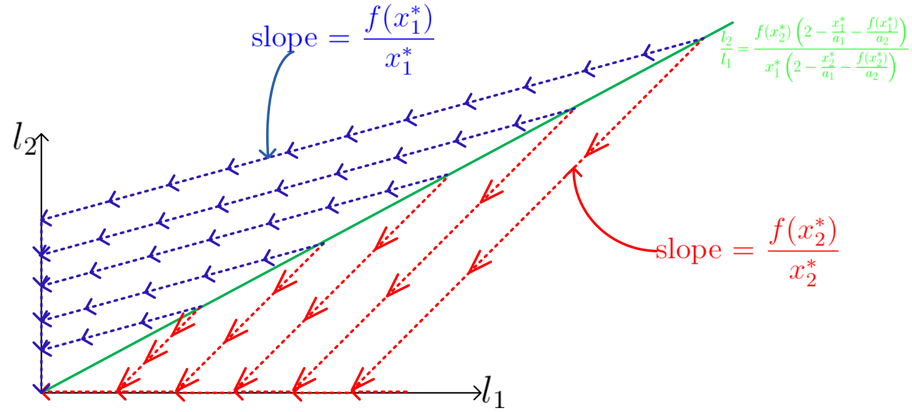

Figure 10: *A demonstration of how the job sizes decrease once the optimal service rate has been allocated. If the job sizes* (l_1, l_2) *lie on the half-plane* $\frac{l_2}{l_1} > \frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$ *then as the jobs are served at the optimal rate the they follow the trajectories parallel to the ones indicated by the blue lines. On the other hand, on the half-plane* $\frac{l_2}{l_1} < \frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$ *the trajectories parallel to the red lines are followed when the optimal policy is applied. If the size pair is on the line* $\frac{l_2}{l_1} = \frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$*, then both the trajectories gives the same cumulative delay. We can say that the quantity* $\frac{l_2}{l_1}$ *acts as a* test parameter *whose value relative to* $\frac{f(x_2^*)\left(2-\frac{x_1^*}{a_1}-\frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2-\frac{x_2^*}{a_1}-\frac{f(x_2^*)}{a_2}\right)}$ *determines which one of the two rate pairs should be used to get the best result.*

5.1 Numerical example

Consider a rate region that is bounded by the first quadrant of the curve

$$\left(\frac{x}{a_1}\right)^{\alpha} + \left(\frac{f(x)}{a_2}\right)^{\alpha} = 1 \quad \alpha \geq 1.$$

Notice that it satisfies both the conditions mentioned in Proposition 1. The optimal rate pair for this rate region can therefore be determined by the procedure mentioned in Remark 3.

The slope of tangent to any point on this border curve is

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = f'(x) = -\left(\frac{a_2}{a_1}\right)^{\alpha}\left(\frac{x}{f(x)}\right)^{\alpha-1}.$$

The equation of any line with slope m through $(0, 2a_2)$ is

$$y - 2a_2 = mx.$$

If this line is a tangent to the curve $f(x)$ at some point $(x_1^*, f(x_1^*))$ then,

$$f(x_1^*) - 2a_2 = -\left(\frac{a_2}{a_1}\right)^{\alpha}\left(\frac{x_1^*}{f(x_1^*)}\right)^{\alpha-1} x_1^*. \tag{11}$$

Moreover, $(x_1^*, f(x_1^*))$ lies on the border curve, so satisfies

$$\left(\frac{x_1^*}{a_1}\right)^{\alpha} + \left(\frac{f(x_1^*)}{a_2}\right)^{\alpha} = 1. \tag{12}$$

By solving (11) and (12) we get

$$\begin{aligned} x_1^* &= a_1\left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}, \\ f(x_1^*) &= \frac{a_2}{2^{\frac{1}{\alpha-1}}}. \end{aligned} \tag{13}$$

Similarly, we can also get the other optimal rate pair $(x_2^*, f(x_2^*))$

$$\begin{aligned} x_2^* &= \frac{a_1}{2^{\frac{1}{\alpha-1}}}, \\ f(x_2^*) &= a_2\left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}. \end{aligned} \tag{14}$$

Finally, the *test parameter* is given by

$$\begin{aligned} \frac{f(x_2^*)\left(2 - \frac{x_1^*}{a_1} - \frac{f(x_1^*)}{a_2}\right)}{x_1^*\left(2 - \frac{x_2^*}{a_1} - \frac{f(x_2^*)}{a_2}\right)} &= \frac{a_2\left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}\left[2 - \frac{a_1\left(1-2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}}{a_1} - \frac{1}{a_2}\frac{a_2}{2^{\frac{1}{\alpha-1}}}\right]}{a_1\left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}\left[2 - \frac{1}{a_1}\frac{a_1}{2^{\frac{1}{\alpha-1}}} - \frac{a_2\left(1-2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}}{a_2}\right]} \\ &= \frac{a_2\left[2 - \left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}} - 2^{-\frac{1}{\alpha-1}}\right]}{a_1\left[2 - 2^{-\frac{1}{\alpha-1}} - \left(1 - 2^{-\frac{\alpha}{\alpha-1}}\right)^{\frac{1}{\alpha}}\right]} \\ &= \frac{a_2}{a_1}. \end{aligned}$$

Thus, if l_1 and l_2 be the job sizes, then from Proposition 1 the optimal rate pair, $(x_1^*, f(x_1^*))$, given by (13) should be used if $\frac{l_2}{l_1} \geq \frac{a_2}{a_1}$. On the other hand, if $\frac{l_2}{l_1} < \frac{a_2}{a_1}$, $(x_2^*, f(x_2^*))$, given by (14) is the optimal rate pair. Moreover, in this particular case, as $x_1^* \geq x_2^*$ and $f(x_1^*) \leq f(x_2^*)$, if we consider the normalized sizes of the flows (*i.e.*, the minimum time required to serve them), $\frac{l_1}{a_1}$ and $\frac{l_2}{a_2}$, the optimal policies still adhere to the SRPT-FM principle as when $\frac{l_1}{a_1} \leq \frac{l_2}{a_2}$ the policy $(x_1^*, f(x_1^*))$ is the optimal policy.

Moreover, it can be easily shown that the PF scheduler always assigns the rate pair $(2^{-\frac{1}{\alpha}}a_1, 2^{-\frac{1}{\alpha}}a_2)$ that is nearly optimal for large values of α. Again, some weight based strategy as discussed in Section 4 can be employed to achieve either the optimal or the PF rates rates in a time slotted system.

5.2 More than two jobs in asymmetric case

When there are three jobs in the asymmetric case we have to consider a total of six cases that depend on the sizes of the jobs and the service rates they are processed at. Although it is possible to do so, the analysis is complicated and no structure is uncovered for cases with more than three jobs. Therefore, in Section 6 we shift our attention to the symmetric case which, we will see, is more amenable to an analytical treatment and reveals some interesting properties.

6 Symmetric case with any number of jobs

In Section 5 we saw that the optimal rates can be obtained for a specific case of only two jobs in asymmetric rate region. When the number of jobs increases, the complexity of analysis grows in such a setting. Now, in this section we see how, for the model introduced in Section 4.1.2, the optimal rates and policies behave, *i.e.*, when the setting is symmetric and at the same time the rate regions have certain properties.

We assume, for a model described in Section 4, that the capacity regions $\mathcal{C}_k$ have the following properties for all $k \in \mathcal{I}_n$.

i **(Compactness)** $\mathcal{C}_k$ is a *compact* region of $\mathbb{R}_+^k$, *i.e.*, $\mathcal{C}_k$ is closed and bounded;

ii **(Symmetry)** $\mathcal{C}_k$ is symmetric, *i.e.*, if $\boldsymbol{c} \in \mathcal{C}_k$, then any permutation of its components, $\tilde{\boldsymbol{c}}$, also lies in $\mathcal{C}_k$.

6.1 One job

If there is only one job at the beginning, *i.e.*, $n = 1$, then the optimal service policy that minimizes delay is the one that assigns the highest capacity to the only job present, *i.e.*,

$$c_1^* := \boldsymbol{c}_1^* = \max\{c \in \mathcal{C}_1\} = c_{11}^*.$$

The existence of this policy is guaranteed by the compactness property (i). The optimal policy is independent of the size of the job.

6.2 More than one job

We now consider the case where there is more than one job present at the beginning, *i.e.*, $n > 1$. We recall that a service policy is a sequence of service rate vectors $\vec{\pi} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n)$. The cumulative delay is then

$$T^{\vec{\pi}} = \sum_{k=1}^{n} t_k^{\vec{\pi}} = \sum_{k=1}^{n} k T_k^{\vec{\pi}}, \tag{15}$$

where $T_k^{\boldsymbol{\pi}}$ is the length of phase k of the policy $\vec{\pi}$.

Let $\{g_k(\cdot)\}_{k=1}^n$ be a sequence of functions such that

$$g_k : \mathcal{C}_k \longrightarrow \mathbb{R}_+ \quad \forall k \in \mathcal{I}_n$$

and defined recursively as,

$$\begin{aligned}
g_1(\boldsymbol{c}_1) &= \frac{1}{c_1}, & G_1^* &= \min_{\boldsymbol{c}_1 \in \mathcal{C}_1} g_1(\boldsymbol{c}_1), \quad g_1(c_1^*) = G_1^*,\\
g_k(\boldsymbol{c}_k) &= \frac{1}{c_{kk}} \left(k - \sum_{j=1}^{k-1} c_{kj} G_j^* \right), & G_k^* &= \min_{\boldsymbol{c}_k \in \mathcal{C}_k} g_k(\boldsymbol{c}_k), \quad g_k(\boldsymbol{c}_k^*) = G_k^*,\\
& & & \forall k \in \mathcal{I}_n.
\end{aligned}$$

The existence of the minimum values of $\{g_k(\cdot)\}_{k=1}^n$ is guaranteed by the compactness property. Moreover, for $g_k(\cdot)$ and G_k^* to be well defined, it is sufficient that the rate regions $\mathcal{C}_k$ be nested.

Lemma 4. *Let $\boldsymbol{a} = (a_1, \ldots, a_n), a_i \in \mathbb{R}^+$ be an n-dimensional row vector with $a_1 < a_2 < \cdots < a_n$, let $B = \{b_1, \ldots, b_n\}$ be a set with $b_i \in \mathbb{R}^+$, $B' = \{\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_{n!}\}$ be the set of all n-dimensional vectors whose elements are the permutation of the of the elements of B. Then,*

i the components of $\boldsymbol{b}^* = \arg\max_{\boldsymbol{b}' \in B'} \boldsymbol{a} \cdot \boldsymbol{b}'$ have the property $b_1^* \leq \cdots \leq b_n^*$ and

ii the components of $\boldsymbol{b}^{**} = \arg\min_{\boldsymbol{b}' \in B'} \boldsymbol{a} \cdot \boldsymbol{b}'$ have the property $b_1^{**} \geq \cdots \geq b_n^{**}$,

where b_i^* and b_i^{**} represent the i-th component, $i \in \mathcal{I}_n$, of $\boldsymbol{b}^*$ and $\boldsymbol{b}^{**}$ respectively.

Proof. We begin step 0 with the vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_n) \in B'$ whose elements are arranged in no particular order. Let $o(k)$ denote the index of the k-th smallest element of $\boldsymbol{b}$, *i.e.*, $o(1) = \arg\min_{i \in \mathcal{I}_n}\{b_i\}$[6]. In step 1 consider another vector $\boldsymbol{b}' = (b_{o(1)}, b_2, \ldots, b_1, \ldots, b_n)$ which is obtained by switching the positions of b_1 and $b_{o(1)}$ in $\boldsymbol{b}$. Without the loss of generality we can assume that $o(1) = k$, *i.e.*, the k-th element of $\boldsymbol{b}$ is the smallest element. Then,

$$\begin{aligned}
\boldsymbol{a} \cdot \boldsymbol{b}' - \boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a} \cdot (\boldsymbol{b}' - \boldsymbol{b}),& \\
&= (a_1, \ldots, a_n) \cdot (b_{o(1)} - b_1, 0, \ldots, b_1 - b_k, \ldots, 0), \\
&= (a_1, \ldots, a_n) \cdot (b_{o(1)} - b_1, 0, \ldots, b_1 - b_{o(1)}, \ldots, 0), \\
&= (b_1 - b_{o(1)})(a_k - a_1), \\
&\geq 0, \\
\Rightarrow \boldsymbol{a} \cdot \boldsymbol{b}' \geq \boldsymbol{a} \cdot \boldsymbol{b}&
\end{aligned}$$

Next, for step 2 consider another vector $\boldsymbol{b}'' = (b_{o(1)}, b_{o(2)}, b_3, \ldots, b_2, \ldots, b_1, \ldots, b_n)$ that is obtained by switching the positions of b_2 and $b_{o(2)}$ in vector $\boldsymbol{b}'$. Here $o(2)$ denotes the index of the second smallest element of $\boldsymbol{b}'$ which in this case is assumed to be its l-th element without the loss of generality. Then,

$$\begin{aligned}
\boldsymbol{a} \cdot \boldsymbol{b}'' - \boldsymbol{a} \cdot \boldsymbol{b}' = \boldsymbol{a} \cdot (\boldsymbol{b}'' - \boldsymbol{b}'),& \\
&= (a_1, \ldots, a_n) \cdot (0, b_{o(2)} - b_2, \ldots, b_2 - b_{o(2)}, \ldots, 0), \\
&= (b_2 - b_{o(2)})(a_l - a_2), \\
&\geq 0, \\
\Rightarrow \boldsymbol{a} \cdot \boldsymbol{b}'' \geq \boldsymbol{a} \cdot \boldsymbol{b}'&
\end{aligned}$$

Continuing in this way, at each step, we successively get non decreasing dot products by constructing a new vector through switching the positions of the k-th and the k-th smallest elements of the vector in the $(k-1)$-th step. Thus, in (a maximum of)

[6] If there are multiple minimum values, only the one with the smallest index is considered. This always makes $o(1)$ a singleton set

$n-1$ steps, we get the vector $\boldsymbol{b}^* = (b_1^*, \ldots, b_n^*) \in B'$ that gives the largest possible dot product $\boldsymbol{a} \cdot \boldsymbol{b}^*$ and has the property $b_i^* \geq b_j^*$for all $i \geq j$.

Similar arguments can be used to prove Part *ii* of the lemma. In this case we take any vector $\boldsymbol{b}$ in step 0. After that, in $(k-1)$-th step where, $k \in \mathcal{I}_n$, we exchange the positions of the k-th and the k-th *largest* elements of the vector to get a new vector for the k-th step. We see that each new vector gives either a smaller or the same dot product as the previous one. Whichever vector we begin with, after performing this operation (a maximum of $n-1$ times) we always end up with the same vector $\boldsymbol{b}^{**}$ with the property $b_i^{**} \leq b_j^{**}$ for all $i \geq j$ that gives the smallest possible dot product with $\boldsymbol{a}$. □

Proposition 5. *If the capacity regions are such that*

$$G_1^* < \cdots < G_n^*,$$

then $c_{k,j+1}^* \geq c_{k,j}^*$ *for all* $k = 2, \ldots, n$ *and* $j = 1, \ldots, k-1$.

Proof. 1° Let $k \in \{2, \ldots, n\}$ and $j \in \{1, \ldots, k-2\}$. Then, G_k^* is defined as,

$$G_k^* = \min_{\boldsymbol{c}_k \in \mathcal{C}_k} g_k(\boldsymbol{c}_k) = \min_{\boldsymbol{c}_k \in \mathcal{C}_k} \frac{1}{c_{kk}} \left(k - \sum_{j=1}^{k-1} c_{kj} G_j^* \right). \tag{16}$$

We consider a fixed value of k. Obviously, the summation term in (16) has to be maximized to get the desired minimum value of the auxiliary function $g_k(\boldsymbol{c}_k)$. Since $\mathcal{C}_k$ is symmetric, when any optimal rate vector $\boldsymbol{c}_k^*$, whose service rates $c_{k,j+1}^* \ngeq c_{k,j}^*$ for all $j = 1, \ldots, k-2$ is found, a vector with any permutation of rates offered by it can also be found in the rate region. From Lemma 4 we can infer that one such permutation of the optimal rate vector, say $\boldsymbol{c}_k^{*\prime}$, that has the property ${c_{k,j+1}^*}' \geq {c_{k,j}^*}'$ for all $j = 1, \ldots, k-2$ should maximize the summation term and therefore minimize the whole expression. This makes $\boldsymbol{c}_k^{*\prime}$ and not $\boldsymbol{c}_k^*$ the optimal one. So, if $c_{k,j+1}^* \ngeq c_{k,j}^*$ is assumed for the optimal rate vector, we always get a *more optimal* rate vector by permuting its elements. This is a contradiction and thus, we can conclude $c_{k,j+1}^* \geq c_{k,j}^*$ for all $k = 2, \ldots, n$ and $j = 1, \ldots, k-2$.

2° Consider the remaining case where $k \in \{2, \ldots, n\}$ and $j = k-1$. The *optimal* rate vector $\boldsymbol{c}_k^* = (c_{k,1}^*, \ldots, c_{k,k-1}^*)$, when there are k jobs in the system is defined as,

$$\boldsymbol{c}_k^* = \arg\min_{\boldsymbol{c}_k \in \mathcal{C}_k} g_k(\boldsymbol{c}_k)$$
$$G_k^* = g_k(\boldsymbol{c}_k^*).$$

Due to the symmetry assumption, an another rate k-tuple, $\boldsymbol{c}_k^{*\prime} = (c_{k,1}^*, \ldots, c_{k,k}^*, c_{k,k-1}^*)$ is also a feasible rate vector. Notice that $\boldsymbol{c}_k^{*\prime}$ is obtained by exchanging the positions last two elements of $\boldsymbol{c}_k^*$. From the definition of minimum value

$$\begin{aligned}
& g(\boldsymbol{c}_k^*) \leq g(\boldsymbol{c}_k^{*\prime}) \\
& \Leftrightarrow \frac{1}{c_{kk}^*}\left(k - \sum_{j=1}^{k-1} c_{kj}^* G_j^*\right) \leq \frac{1}{c_{k,k-1}^*}\left(k - \sum_{j=1}^{k-2} c_{kj}^* G_j^* - c_{kk}^* G_{k-1}^*\right) \\
& \Leftrightarrow c_{k,k-1}^*\left(k - \sum_{j=1}^{k-1} c_{kj}^* G_j^*\right) \leq c_{kk}^*\left(k - \sum_{j=1}^{k-2} c_{kj}^* G_j^* - c_{kk}^* G_{k-1}^*\right) \\
& \Leftrightarrow (c_{k,k-1}^* - c_{kk}^*)k - (c_{kk-1}^* - c_{kk}^*)\sum_{j=1}^{k-2} c_{kj}^* G_j^* - (c_{k,k-1}^{*2} - c_{kk}^{*2})G_{k-1}^* \leq 0 \\
& \Leftrightarrow (c_{k,k-1}^* - c_{kk}^*)\left(k - \sum_{j=1}^{k-1} c_{kj}^* G_j^* - c_{kk}^* G_{k-1}^*\right) \leq 0 \\
& \Leftrightarrow (c_{k,k-1}^* - c_{kk}^*)(c_{kk}^* G_k^* - c_{kk}^* G_{k-1}^*) \leq 0 \\
& \Leftrightarrow c_{kk}^*(c_{k,k-1}^* - c_{kk}^*)(G_k^* - G_{k-1}^*) \leq 0
\end{aligned}$$

Since the quantities c_{kk}^* and $(G_k^* - G_{k-1}^*)$ are both positive, the above inequality holds only if $c_{kk}^* \geq c_{k,k-1}^*$ for $k \in \{2, \ldots, n\}$.

This completes the proof. □

Theorem 6. *If the capacity regions $\mathcal{C}_1, \ldots, \mathcal{C}_n$ are such that*

$$G_1^* < \cdots < G_n^*,$$

then the optimal operating policy is $\vec{\pi}^* = (\boldsymbol{c}_1^*, \ldots, \boldsymbol{c}_n^*)$ for all sizes $s_{n1} \geq \cdots \geq s_{nn}$. The cumulative delay $T^{\vec{\pi}^*}$ satisfies

$$T^{\vec{\pi}^*} = \sum_{k=1}^{n} s_{nk} G_k^*.$$

In addition, $c_{k,j+1}^* \geq c_{k,j}^*$ for all $k = 2, \ldots, n$ and $j = 1, \ldots, k-1$ so that the optimal policy applies the SRPT-FM principle.

Proof. The result can be proved by induction. For $n = 1$, the result is clearly true

$$T^{\vec{\pi}^*} = \frac{s_{11}}{c_1^*} = \min_{c_1 \in \mathcal{C}_1} \frac{s_{11}}{c_1^*} = \min_{\vec{\pi} \in \Pi_1} T^{\vec{\pi}}.$$

In addition, $G_1^* = \frac{1}{c_1^*}$ so that $T^{\vec{\pi}^*} = s_{11} G_1^*$ as claimed. Now, assume that the result is true for $n = m$. We will prove this assumption implies that the result is true for $n = m + 1$.

Let us first assume that the SRPT-FM principle is followed when there are $m + 1$ jobs in the system. This means, the first job to leave the system will be the smallest job with the index $(m + 1)$ that has the size $s_{m+1,m+1}$ and gets the highest possible processing rate $c_{m+1,m+1}^*$. After its departure at $\frac{s_{m+1,m+1}}{c_{m+1,m+1}^*}$ time units from the beginning, the sizes of the jobs still remaining in the queue will be $s_{m+1,i} - \frac{s_{m+1,m+1}}{c^*_{m+1,m+1}} c^*_{m+1,i}$, $i = 1, \ldots, m$. The cumulative delay for this policy is thus given by,

$$T^{\vec{\pi}^*} = (m+1)\frac{s_{m+1,m+1}}{c^*_{m+1,m+1}} + \sum_{i=1}^{m} \left(s_{m+1,i} - \frac{s_{m+1,m+1}}{c^*_{m+1,m+1}} c^*_{m+1,i} \right) G^*_i, \tag{17}$$

where $\vec{\pi}^* = (\boldsymbol{c}^*_1, \ldots, \boldsymbol{c}^*_{m+1})$. The second part of (17) comes from the application of our induction hypothesis—after the departure of one job, only m jobs now remain in the system and therefore the induction hypothesis can be applied. Proceeding further,

$$\begin{aligned}
T^{\vec{\pi}^*} &= \sum_{i=1}^{m} s_{m+1,i} G^*_i + s_{m+1,m+1} \frac{1}{c^*_{m+1,m+1}} \left((m+1) - \sum_{i=1}^{m} G^*_i c^*_{m+1,i} \right), \\
&= \sum_{i=1}^{m} s_{m+1,i} G^*_i + s_{m+1,m+1} G^*_{m+1}, \\
&= \sum_{i=1}^{m+1} s_{m+1,i} G^*_i.
\end{aligned}$$

We will now show that if the SRPT-FM principle is not followed when there are $(m+1)$ jobs, the cumulative delay is larger than the case when this principle is followed, making SRPT-FM the optimal policy.

Let $\vec{\pi} = (\boldsymbol{c}^*_1, \ldots, \boldsymbol{c}^*_m, \boldsymbol{c}_{m+1})$ be the policy that is applied when there are $m+1$ jobs in the system. Since it is not necessarily SRPT-FM, any job among the $m+1$ present may leave first which is denoted by the index j. Let $o(k)$ denote the index of the k-th largest job after the first job departs. Then, from the definition of cumulative delay and our induction hypothesis,

$$\begin{aligned}
T^{\vec{\pi}} &= (m+1)\frac{s_{m+1,j}}{c_{m+1,j}} + \sum_{i=1}^{m} \left(s_{m+1,o(i)} - \frac{s_{m+1,j}}{c_{m+1,j}} c_{m+1,o(i)} \right) G^*_i, \\
&= \sum_{i=1}^{m} s_{m+1,o(i)} G^*_i + s_{m+1,j} \frac{1}{c_{m+1,j}} \left((m+1) - \sum_{i=1}^{m} G^*_i c_{m+1,o(i)} \right), \\
&\overset{(i)}{\geq} \sum_{i=1}^{m} s_{m+1,o(i)} G^*_i + s_{m+1,j} \frac{1}{c^*_{m+1,m+1}} \left((m+1) - \sum_{i=1}^{m} G^*_i c^*_{m+1,i} \right), \\
&= \sum_{i=1}^{m} s_{m+1,o(i)} G^*_i + s_{m+1,j} G^*_{m+1}, \\
&\overset{(ii)}{\geq} \sum_{i=1}^{m+1} s_{m+1,i} G^*_i.
\end{aligned}$$

Inequality *(i)* comes from the fact that the auxiliary function $g_{m+1}(\boldsymbol{c}_{m+1})$ is minimized when $\boldsymbol{c}_{m+1} = \boldsymbol{c}^*_{m+1}$. Inequality *(ii)* comes from Lemma 4. We can therefore conclude that the cumulative delay for any policy other than SRPT-FM is greater than that of SRPT-FM for any number of jobs. This completes the proof. $\square$

7 Alpha-ball rate region

Consider an α - ball shaped rate region in which when there are n jobs in the system, the elements of the service rate vector $\boldsymbol{c}_n = (c_{n1}, \ldots, c_{nn})$ satisfies the inequality

$$c_{n1}^{\alpha} + c_{n2}^{\alpha} + \cdots + c_{nn}^{\alpha} \leq 1 \tag{18}$$

for all $n \in \mathbb{Z}_+$. Clearly, the region is compact as well as symmetric. For such rate region, it is possible to obtain analytical expressions for the optimal value of the auxiliary function $G_k^* = g_k(\boldsymbol{c}_k^*)$ and the optimal service rate vectors $(\boldsymbol{c}_1^*, \boldsymbol{c}_2^*, \ldots, \boldsymbol{c}_n^*)$ for $k \in \{1, \ldots, n\}$.

For this, we begin by recalling the definitions of the auxiliary function $g_k(\cdot)$ and its optimal value G_k^* which are defined as,

$$\begin{aligned}
g_k(\boldsymbol{c}_k) &:= \frac{1}{c_{kk}} \left(k - \sum_{j=1}^{k-1} c_{kj} G_j^* \right), \\
G_k^* &:= \min_{\boldsymbol{c}_k \in \mathcal{C}_k} g_k(\boldsymbol{c}_k) = \min_{\boldsymbol{c}_k \in \mathcal{C}_k} \frac{1}{c_{kk}} \left(k - \sum_{j=1}^{k-1} c_{kj} G_j^* \right) \text{ and} \\
\boldsymbol{c}_k^* &:= \arg\min_{\boldsymbol{c}_k \in \mathcal{C}_k} g_k(\boldsymbol{c}_k)
\end{aligned}$$

for all $k \in \mathbb{Z}_+$. When there are n jobs, the components of the optimal rate vector $\boldsymbol{c}_n^* = (c_{n1}^*, c_{n2}^*, \ldots, c_{nn})$ satisfy the linear equation

$$\sum_{k=1}^{n} G_k^* c_{nk}^* = n. \tag{19}$$

Moreover, the optimal rate should fall on the border of the α-ball (18), *i.e.*, its components satisfy the equation

$$c_{n1}^{*\alpha} + c_{n2}^{*\alpha} + \cdots + c_{nn}^{*\alpha} = 1. \tag{20}$$

If the optimal rate vector is unique then the hyperplane given by (19) should be a tangent to the α-ball (20). This means, at the point of intersection of (20) and the hyperplane (19) the gradients of both the hypercurves must be proportional.

The gradient of the ball is given by

$$\nabla_n (c_{n1}^{*\alpha} + c_{n2}^{*\alpha} + \cdots + c_{nn}^{*\alpha}) = \alpha (c_{n1}^{*(\alpha-1)} \hat{\boldsymbol{c}}_{n1} + c_{n2}^{*(\alpha-1)} \hat{\boldsymbol{c}}_{n2} + \cdots + c_{nn}^{*(\alpha-1)} \hat{\boldsymbol{c}}_{nn}), \tag{21}$$

and the gradient of the hyperplane is given by

$$\nabla_n (G_1^* c_{n1}^* + \cdots + G_n^* c_{nn}^*) = G_1^* \hat{\boldsymbol{c}}_{n1} + \cdots + G_n^* \hat{\boldsymbol{c}}_{nn}. \tag{22}$$

Here $\hat{\boldsymbol{c}}_{ni}$ represents the unit vector in the c_{ni} direction of the rate space. As (21) and (22) point to the same direction,

$$\begin{aligned}
&G_1^*\hat{\boldsymbol{c}}_{n1} + \cdots + G_n^*\hat{\boldsymbol{c}}_{nn} = \lambda\alpha(c_{n1}^{*(\alpha-1)}\hat{\boldsymbol{c}}_{n1} + c_{n2}^{*(\alpha-1)}\hat{\boldsymbol{c}}_{n2} + \cdots + c_{nn}^{*(\alpha-1)}\hat{\boldsymbol{c}}_{nn})\\
&\Rightarrow \frac{G_1^*}{c_{n1}^{*(\alpha-1)}} = \cdots = \frac{G_n^*}{c_{nn}^{*(\alpha-1)}} = \lambda\alpha\\
&\Leftrightarrow c_{ni}^{*(\alpha-1)} = \frac{G_i^*}{\lambda\alpha} \quad \text{for all } i \in \{1,\ldots,n\},
\end{aligned} \tag{23}$$

where λ is the constant of proportionality (the Lagrange multiplier). Using (23) in (19) and (20), we get

$$\begin{aligned}
{G_1^*}^{\frac{\alpha}{\alpha-1}} + \cdots + {G_n^*}^{\frac{\alpha}{\alpha-1}} &= n(\lambda\alpha)^{\frac{1}{\alpha-1}} \qquad \text{and}\\
{G_1^*}^{\frac{\alpha}{\alpha-1}} + \cdots + {G_n^*}^{\frac{\alpha}{\alpha-1}} &= (\lambda\alpha)^{\frac{\alpha}{\alpha-1}},
\end{aligned} \tag{24}$$

implying that $\lambda\alpha = n$. Now using this value of $\lambda\alpha$ in (24) we get,

$${G_1^*}^{\frac{\alpha}{\alpha-1}} + \cdots + {G_n^*}^{\frac{\alpha}{\alpha-1}} = n^{\frac{\alpha}{\alpha-1}}, \tag{25}$$

which can be used to recursively calculate the values of G_k^* for all $k \in \{1,\ldots,n\}$ by noting that $G_1^* = 1$, *i.e.*,

$$\begin{aligned}
{G_2^*}^{\frac{\alpha}{\alpha-1}} &= 2^{\frac{\alpha}{\alpha-1}} - G_1^* = 2^{\frac{\alpha}{\alpha-1}} - 1,\\
{G_3^*}^{\frac{\alpha}{\alpha-1}} &= 3^{\frac{\alpha}{\alpha-1}} - {G_2^*}^{\frac{\alpha}{\alpha-1}} - {G_1^*}^{\frac{\alpha}{\alpha-1}} = 3^{\frac{\alpha}{\alpha-1}} - 2^{\frac{\alpha}{\alpha-1}},\\
&\vdots\\
{G_n^*}^{\frac{\alpha}{\alpha-1}} &= n^{\frac{\alpha}{\alpha-1}} - (n-1)^{\frac{\alpha}{\alpha-1}}.
\end{aligned}$$

Finally, the rates c_{kj}^* are given by

$${c_{kj}^*}^{\alpha-1} = \frac{G_j^*}{k}$$

for $k \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,k\}$. Clearly, for $\alpha \geq 1$, $G_1^* < \cdots < G_n^*$ and $c_{k,j+1}^* \geq c_{k,j}^*$ for all $k = 2,\ldots,n$ and $j = 1,\ldots,k-1$, in accordance with Theorem 6.

8 Simulation for the dynamic setting

As discussed in Section 4, flow-level simulation is used in this thesis to study the application of the optimal policy in the static setting for the dynamic flow arrivals. The rate region is assumed to take the shape of α-ball described in Section 7. This allows us to use the results of Section 7 and analytically determine the long-term service rate that should be allocated to each flow in the system, allowing the simulation to be done at the flow-level. Then, for the same arrivals and service time requirements (which aids in reducing the variance), we apply the PF policy and compare it with the result obtained by the application of the optimal policy. As a symmetric rate region is considered, the long-term service rate in case of PF policy can also be easily determined. For various arrival rates and different values of α the simulations are done in the dynamic setting and compared with the PF policy in the same.

8.1 Description of the simulator

The simulator is implemented in Mathematica. The main body of the simulator is a function that takes the arrival rate (λ), the number of arrivals (n) and the policy to be applied (p) as the input arguments. The basic events are the arrival and the departure of flows which change the various state variables. A simplified structure of the simulator is summarized below:

Events

- Flow arrival `ARR`
- Flow departure `DEP`

State variables

- The ordered set of the flows currently queued in the system sorted by the amount of bits left to transmit ($\mathbf{Q}$).
- The list of the arrival times of the flows in the system, $\mathbf{A}$, indexed by the elements of $\mathbf{Q}$.
- The list of the bits remaining to transmit for each flow, $\mathbf{R}$, indexed by the elements of $\mathbf{Q}$.
- The departure time of the smallest job in the system (T_d)
- The time instant when the previous event (arrival or departure) occurred (t_p)
- The current service rate vector (r)

Special event

- The end of the simulation `END`

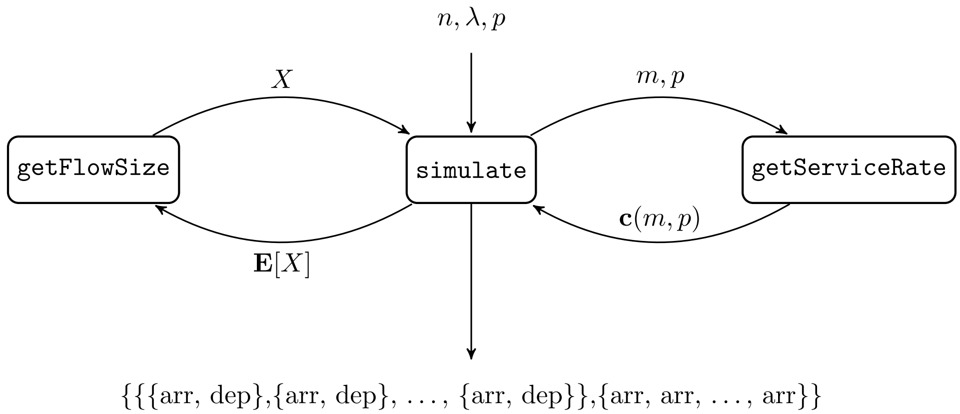

Figure 11: *The structure of the simulator. The* `simulate` *module does the main simulation when the relevant parameters are supplied to it. The functions—*`getFlowSize` *and* `getServiceRate` *are called when new flows arrive. The simulation ends when 'n' flows have arrived, at which point* `simulate` *returns the arrival and the departure times of the flows that have been served.*

The simulator is implemented in Mathematica in the module `simulate` which is called with the input arguments parameters n, λ and p. With every new arrival of a flow this module calls the external functions—`getFlowSize` and `getServiceRate` to get the size of the flow and the service rate vector to process these flows by supplying the appropriate parameters. It can be summarized as in Figure 11. The source code of the simulator is given in Appendix A.

8.2 Numerical results

A simulation for 50000 arrivals has been performed and the first 20000 observations have been ignored in the final analysis to remove the effect of transients and observe only the steady state behavior. The flows arrive according to the Poisson process with different arrival rates and have a size with an exponential distribution [with mean 1 bit].

The results from the simulation are summarized as in Figures 12–16. Clearly the optimal policy in the static case performs better than the PF policy for a wide range of arrival rates. We also observe that after some particular value of the arrival rate, the performance of optimal policy begins to somewhat decline. This is because the PF policy in this rate region is able to extract the highest possible opportunism from the system. At lower arrival rates when there are few flows in the queue, the difference of opportunistic gain between the PF and the optimal policy for the static case is small. As the arrival rate increases, so does the queue length and thus the difference between the opportunistic gain of the PF policy and the static-case optimal policy. This implies that by consistently operating at the optimal policy, some portion of opportunism is always lost at the higher arrival rates and this loss cannot be compensated by the SRPT-like behavior of the optimal policy. Since the PF policy can utilize all the opportunistic gain of the system, its performance relative to the optimal policy gets better with increasing arrival rate.

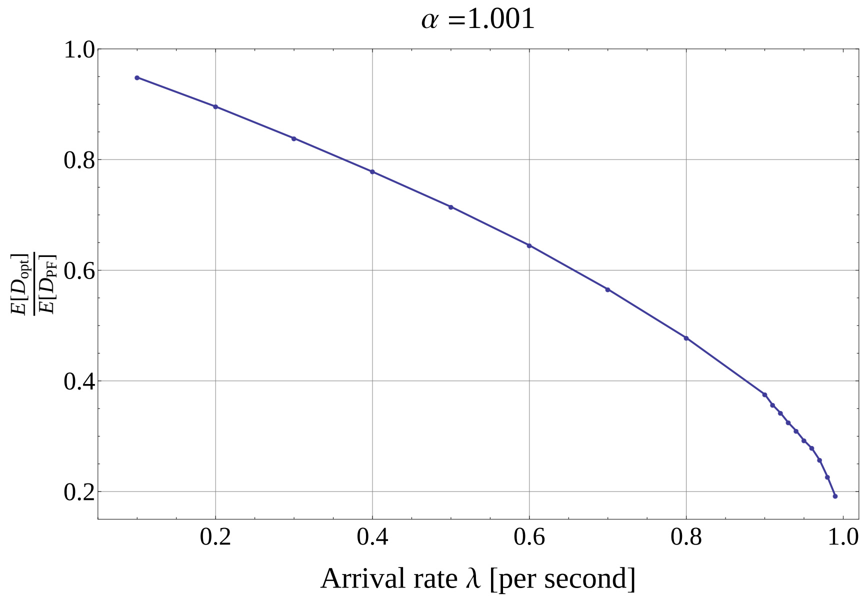

Figure 12: *The performance of the optimal policy against the PF policy when* $\alpha = 1.001$. *The rate region is very close to being non opportunistic. So, the optimal policy, that is close SRPT is clearly better than the PF policy which is similar to PS in this case.*

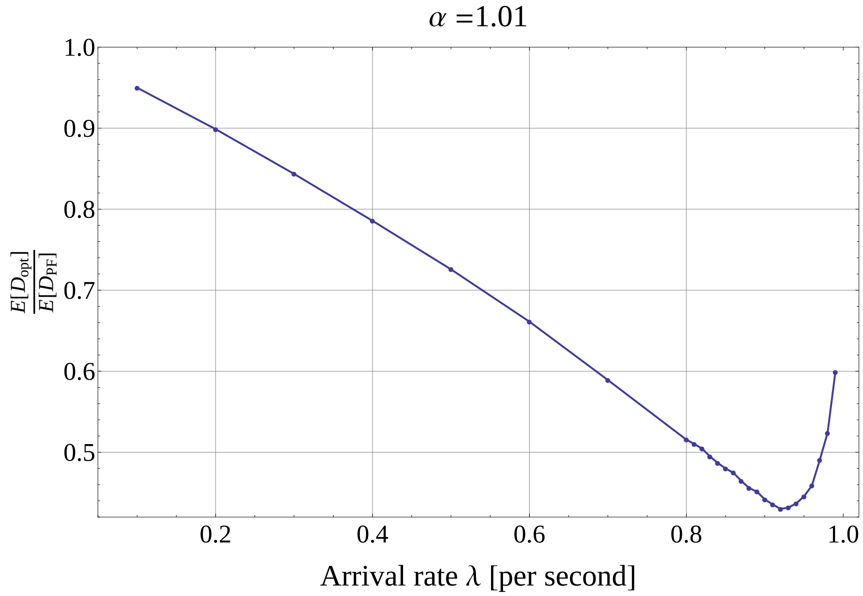

Figure 13: *The performance of the optimal policy against the PF policy when* $\alpha = 1.01$. *The optimal static-case policy performs better than the PF policy in this almost non-opportunistic case. The performance of the optimal policy depends on the arrival rate. Like the case of a non-opportunistic* $M/G/1$ *queue, where the SRPT policy performs consistently better than the PS policy, here also we observe that the optimal static-case policy performs better than the PF policy. In this case, the optimal static-case policy is akin to the SRPT policy in* $M/G/1$ *while the PF policy is equivalent to PS. However, at very high arrival rates, the gain achieved by the application of the optimal policy begins to decline.*

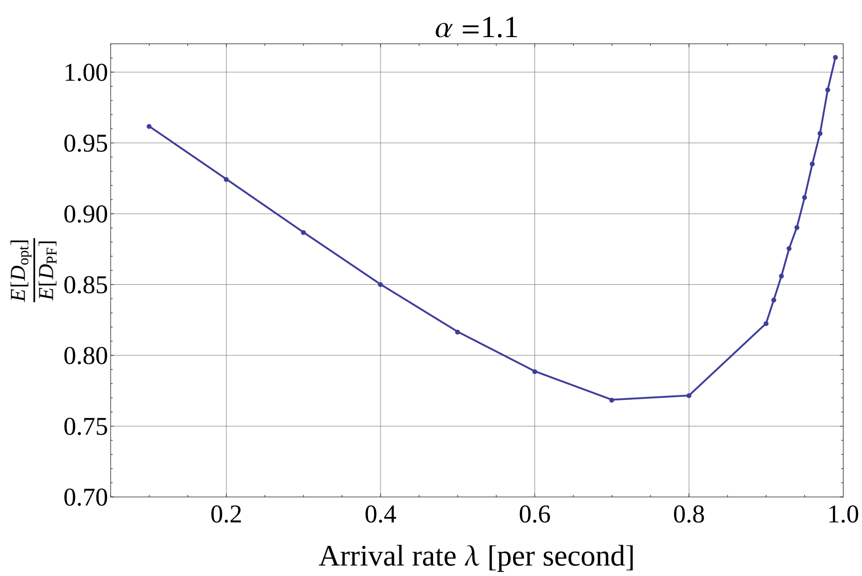

Figure 14: *The performance of the optimal policy against the PF policy when* $\alpha = 1.1$. *The optimal static-case policy is consistently better than the PF policy. At high arrivals we again see that the performance gain of the optimal policy starts to decline.*

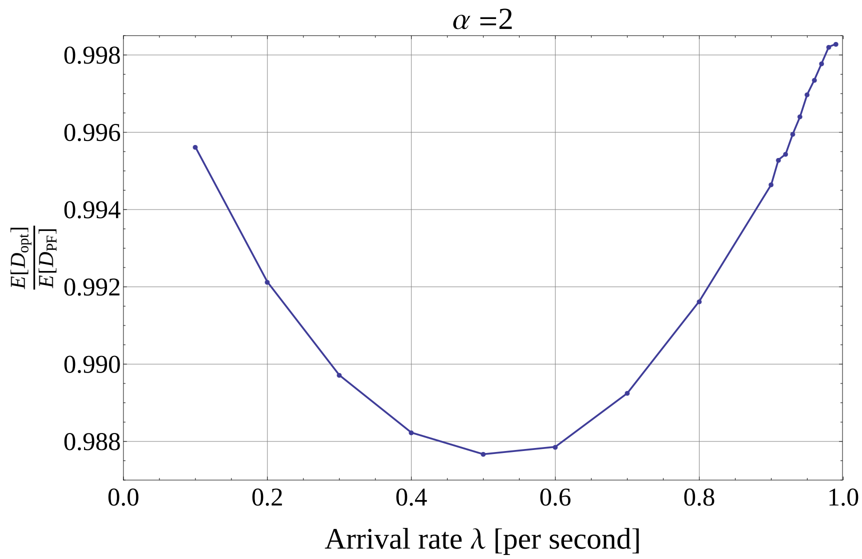

Figure 15: *The performance of the optimal policy against the PF policy when $\alpha = 2$. The optimal static-case policy seems to be marginally better that the PF policy. The two policies are almost the same for very low and very high arrival rates.*

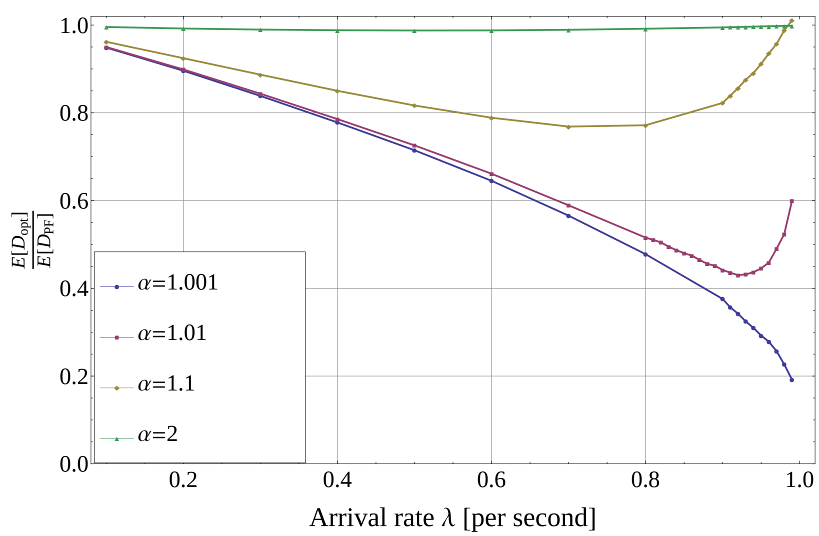

Figure 16: *The performance of the optimal policy against the PF policy for different values of α. For values of $\alpha \approx 1$ we see that the optimal policy is better than the PF policy and the gain with the optimal policy is arrival rate dependent. Furthermore, the optimal policy performs at least as good as the PF policy for a wide range of arrival rates.*

9 Conclusions

In this thesis we have studied the problem of optimally scheduling elastic flows in an opportunistic environment, mainly inspired by its applications in cellular wireless networks. More specifically, we have devised policies that minimize the *cumulative delays* of flows in various static settings. In a dynamic setting we have studied the effects of the application of these policies on *average* delays of the flows. We have focused our attention mainly in the rate regions that arise due to fast scheduling in a time scale separated wireless environment. The schedulers here can flexibly schedule the flows and have complete information about flow sizes while doing so. The results are valid for a general queueing/scheduling system that involves some kind of rate region where the scheduler can choose the customer to provide service based on the information it has about the remaining work of customer.

9.1 Summary of work accomplished

The main work of this thesis has been presented in Sections 5–8. First we have studied the asymmetric setting involving only two jobs that can arise, *e.g.*, when a base station tries to schedule flows to two users that have different spatial locations. We see that if the rate region involved in this case is convex and bounded by a decreasing curve, there can only be pair of possible *policies* that can give the optimal result. The choice between these two policies can be done with the help of the information about the size of the flows. As an example, we apply this result in a model rate region for which we are able to get the explicit expressions of the optimal long-term service rates, which bear a striking resemblance to the SRPT-FM scheduling policy.

In the asymmetric case, when more than two flows are involved the analysis becomes too complicated. However, when we analyze the symmetric, compact and nested rate regions in Section 6 we see that we can, in general, characterize the optimal policies when $G_1^* < \cdots < G_n^*$, where $G_k^*, k \in \{1 \dots n\}$ are defined in Section 6.2. Under such conditions, the optimal policy follows the SRPT-FM principle and the cumulative delay has a simple expression that is based on the values of G_i^* and the initial sizes of the flows. We then determine the optimal rate vector for a simple rate region, called the α-ball region, that satisfies the above stated properties.

In the dynamic setting we study the application of this static-case optimal policy for the α-ball rate region. We see that the optimal policy performs no worse than the baseline PF policy and the performance of this policy is dependent on the arrival rate. When all opportunism is absent the static-case optimal policy is always better than the PF policy and its performance gets better as the arrival rate increases. When the opportunism of the system increases, this kind of strong optimality is not observed primarily because the static-case optimal policy does not fully exploit the opportunism of the system and forsakes a part of the opportunistic gain by providing higher rate to smaller flows.

9.2 Future works

The work presented in this thesis can be extended in various ways. Some future directions of work are listed below.

General asymmetric setting

In this thesis we have studied the nature of optimal policies for only two flows in an asymmetric setting. The analysis for more than two flows is not analytically tractable by similar methods. Since the generalized asymmetric setting is more realistic in terms of applications, it will be worthwhile to analyze these environments either analytically or through simulation and observe the optimal policies.

Results for more realistic rate regions

The rate region we have considered in the thesis is a highly contrived one. Although similar rate region may be found in real-life cases, making analogous analysis for realistic rate regions in actual wireless channels would be a good direction for future work.

Optimal policy for the dynamic setting

We have devised optimal policy for the static setting. However, the optimal policy for the dynamic setting in which flows arrive also has not been studied. We have observed some connection between the SRPT-FM policy and optimality. The policy considered here seems to do a good job in the dynamic setting as well, but it is not known if this is the optimal policy. So, future work may involve studying policies based on SRPT-FM that give optimal results in the dynamic setting as well.

References

[1] R. Ahlswede. The capacity region of a channel with two senders and two receivers. *The Annals of Probability*, 2(5):805–814, 1974.

[2] 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. *Communications Magazine, IEEE*, 38(7):70–77, July 2000.

[3] Q. Bi and S. Vitebsky. Performance analysis of 3G-1X EVDO high data rate system. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 1, pages 389–395. IEEE, Mar. 2002.

[4] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking*, 13(3):636–647, 2005.

[5] S. Borst and M. Jonckheere. Flow-level stability of channel-aware scheduling algorithms. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*. IEEE, 2006.

[6] H. Holma and A. Toskala. *LTE for UMTS-OFDMA and SC-FDMA based radio access*. Wiley, Chichester, 2009.

[7] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *IEEE Vehicular Technology Conference*, volume 3, pages 1854–1858. IEEE, 2000.

[8] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

[9] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *The Journal of the Operational Research Society*, 49(3):237–252, 1998.

[10] L. Kleinrock. *Queueing Systems. Volume 1: Theory*. Wiley-Interscience, 1st edition, 1975.

[11] L. Kleinrock. *Queueing Systems. Volume 2: Computer Applications*. Wiley-Interscience, 1st edition, 1976.

[12] J. Liu, A. Proutiere, Y. Yi, M. Chiang, and H. Poor. Stability, fairness, and performance: A flow-level study on nonconvex and time-varying rate regions. *Information Theory, IEEE Transactions on*, 55(8):3437 –3456, Aug. 2009.

[13] X. Liu, E. Chong, and N. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 41(4):451–474, 2003.

[14] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15(1):185–201, 2000.

[15] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *Networking, IEEE/ACM Transactions on*, 8(5):556 –567, Oct. 2000.

[16] M. L. Pinedo. *Scheduling : Theory, Algorithms, and Systems.* Springer, New York, 3rd edition, 2008.

[17] B. Sadiq and G. de Veciana. Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics. In *Proc. 22nd International Teletraffic Congress*, 2010.

[18] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.

[19] S. Segan. Verizon wireless sharpens LTE roadmap. *PC Magazine*, 2009.

[20] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

[21] A. Stolyar. On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation. *Operations Research*, 53(1):12–25, 2005.

Appendices

A Mathematica Source Code

The code of the *main simulator* is listed.

```
simulate[n_Integer, \[Lambda]_, policy_String] :=
  Module[
   {collectedStatistics = {}, nextArrivalInstant, queue,
    arrivalsCoun, nextDepartureInstant, previousEventInstant,
    currentRate, arrivalsCount, presentTime,
    flowSize, \[Mu] = 1, \[Alpha] = 1.001, serviceRate = {}},
   (*Initialize the state variables*)
   queue = {};
   (*Generate the next arrival and its size*)
   nextArrivalInstant = 0;
   nextDepartureInstant = 0;
   nextArrivalInstant = generateNextArrival[\[Lambda]];
   flowSize = getFlowSize[\[Mu]];
   arrivalsCount = 1;
   nextDepartureInstant = nextArrivalInstant + 1;
   previousEventInstant = nextArrivalInstant;
   currentRate = {};
   While [arrivalsCount <= n,
    (*if there is an arrival before the next scheduled departure*)
    If[nextArrivalInstant < nextDepartureInstant,
      presentTime = nextArrivalInstant;
      (*If there is any job in the queue update the remaining work*)
      If[Length[queue] != 0,
       queue[[All, 3]] =
         queue[[All, 3]] - (presentTime - previousEventInstant)*
           currentRate;
       ];
      (*queue={arrival time, original size, remaining size}*)
      AppendTo[queue, {presentTime, flowSize, flowSize}];
      (*sort the queue by remaining size in the ascending order (the \
smallest first)*)
      queue = Sort[queue, #1[[3]] < #2[[3]] &];
      (*get the new service rates for the flows*)
      serviceRate = getServiceRate[Length[queue], \[Alpha], policy];
      (*calculate the next departure time if no new arrival occurs*)
      nextDepartureInstant =
       presentTime + queue[[1, 3]]/serviceRate[[1]];
      (*Update the state variables*)
      currentRate = serviceRate;
      previousEventInstant = presentTime;
      (*generate the next arrival*)
      nextArrivalInstant =
       presentTime + generateNextArrival[\[Lambda]];
      (*get the size of the next arrival*)
      flowSize = getFlowSize[\[Mu]];
      arrivalsCount++;
      ,(*If there is a departure before the next scheduled arrival*)
      presentTime = nextDepartureInstant;
      (*Collected Statistics={Arrival time, FlowSize, departure time,
      delay}*)
```

```
      AppendTo[
       collectedStatistics, {queue[[1, 1]], queue[[1, 2]],
        presentTime, presentTime - queue[[1, 1]]}];
      (*Update the unfinished work of the elements of the queue*)
      queue[[All, 3]] =
       queue[[All, 3]] - (presentTime - previousEventInstant)*
         currentRate;
      (*remove the first element of the queue whose work is finished*)
      queue = Rest[queue];
      If[Length[queue] != 0,
       serviceRate = getServiceRate[Length[queue], \[Alpha], policy];
       nextDepartureInstant =
        presentTime + queue[[1, 3]]/serviceRate[[1]];
       currentRate = serviceRate;
       previousEventInstant = presentTime;
       ,(*if the queue becomes empty go to the next arrival*)
       previousEventInstant = nextArrivalInstant;
       (*make sure you go to the arrival branch*)
       nextDepartureInstant = nextArrivalInstant + 1;
       currentRate = {};
       ];
      ];
    ];
   (*{Sort the collected statistics according to the arrival time}*)
   Sort[collectedStatistics, #1[[1]] < #2[[1]] &]
   ];
```

The codes for flow size generation, service rate vector calculation.

```
generateNextArrival[x_] := -Log[Random[]]/x

getFlowSize[x_] := -Log[Random[]]/x

getServiceRate[n_Integer, \[Alpha]_, policy_String] :=
 Module[
  {},
  Switch[policy,
   "pf",
   Table[n^(-1/\[Alpha]), {n}],
   "opt",
   G = Table[(i^(\[Alpha]/(\[Alpha] - 1)) - (i -
           1)^(\[Alpha]/(\[Alpha] - 1)))^((\[Alpha] -
          1)/\[Alpha]), {i, n}];
   c = (G/n)^(1/(\[Alpha] - 1));
   (*return the rates in the descending order*)
   Reverse[c],
   "ps",
   Table[1/n, {n}],
   _,
   getServiceRate::nnarg = "The policy `1` is not a valid policy";
   Message[getServiceRate::nnarg, policy]
   ]
  ]
```

B Two-user rate region

Here we consider a time-slotted system where base station is serving one of two users indexed by $\mathcal{I}_2 = \{1, 2\}$ in a time slot. For user $i \in \mathcal{I}_2$ the rate process is $R_i(t)$. Moreover, $R_1(t)$ and $R_2(t)$ are independent and have the same stationary distribution as $R(t)$, which takes values from the set $\mathcal{R} = \{r_1, r_2\}, r_1 > r_2$ such that,

$$\mathbf{P}\{R(t) = r_1\} = p,$$
$$\mathbf{P}\{R(t) = r_2\} = 1 - p, \forall t.$$

Let $\gamma = \frac{r_1}{r_2}$ be the ratio of the two possible value of rates. In the time slot beginning at t the base station serves user 1 when $R_1(t) > wR_2(t)$ with the rate $R_1(t)$. If $R_1(t) < wR_2(t)$ user 2 is served with the rate $R_2(t)$. If $R_1(t) = wR_2(t)$, the selection of the user to be served is made based on some well defined rule that is considered later. It is also assumed that the values of rate process change only at the beginning of a time slot and remain constant in that time slot. From Section 3 we know that such weight based strategy allows us to assign long-term throughputs to the users, which is represented by some extremal point of the rate region. We will see in this section that this strategy also enables us to determine the whole rate region for the simple case considered here.

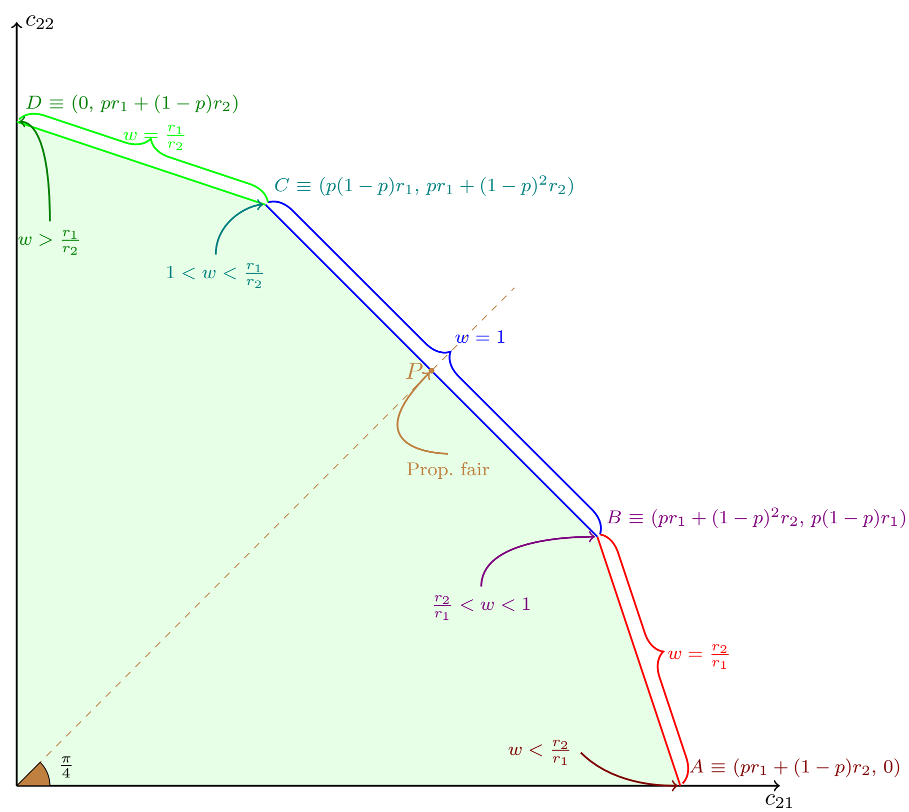

Figure B1: *Illustration of rate region*

$R_1(t)$	$wR_2(t)$	Prob.	Chosen user for case						
			$w < \frac{1}{\gamma}$	$w=\frac{1}{\gamma}$	$\frac{1}{\gamma} < w < 1$	$w = 1$	$1 < w < \gamma$	$w=\gamma$	$w > \gamma$
r_1	wr_1	p^2	1	1	1	1/2	2	2	2
r_1	wr_2	$p(1-p)$	1	1	1	1	1	1/2	2
r_2	wr_1	$p(1-p)$	1	1/2	2	2	2	2	2
r_2	wr_2	$(1-p)^2$	1	1	1	1/2	2	2	2

Table B1: *Table for various cases that can arise for different values of weight w.*

Case I: $w < \frac{1}{\gamma}$

In this case user 1 is always chosen as $R_1(t) > wR_2(t)$ and served at an average rate of $pr_1 + (1-p)r_2$. No service is provided to user 2. This corresponds to the point A in Figure B1.

Case II: $w=\frac{1}{\gamma}$

In this case the various permutation of rates that can occur is shown in Table B1. If the ties are always broken in the favor of user 1, we get similar results as Case I. On the other hand, if the ties are always broken in favor of user 2, the average service rates for user 1 and 2 are $pr_1 + (1-p)^2 r_2$ and $p(1-p)r_1$ respectively (corresponding to point B in Figure B1). If the tie breaking rule favors user 1 with a probability p' and user 2 with a probability $1-p'$, by adjusting the value of p' we can get intermediate results (on the red line in Figure B1) between these two extreme points (A and B in Figure B1).

Case III: $\frac{1}{\gamma} < w < 1$

Again from Table B1 we can see that ties do not occur and the average service rates, $pr_1 + (1-p)^2 r_2$ and $p(1-p)r_1$, are provided to the two users (corresponding to point B in Figure B1).

Case IV: $w = 1$

We refer to Table B1 for various permutations of rates that can occur. Ties are possible in this case and if they are always broken in the favor of user 1, we get similar results as Case III. On the other hand, if the ties are always broken in favor of user 2, the average service rates received by users 1 and 2 are $p^2 r_1$ and $pr_1 + (1-p)^2 r_2$ respectively (corresponding to point C in Figure B1). If the tie breaking rule favors user 1 with a probability p' and user 2 with a probability $1-p'$, by adjusting the value of p' we can get intermediate results (on the blue line in Figure B1) between these two extreme points (B and C in Figure B1). The scheduling is *proportionally fair* when $p' = 0.5$ (point P in Figure B1).

Case V: $1 < w < \gamma$

From Table B1 we see that ties never occur and the two users get rates $p(1-p)r_1$ and $pr_1 + (1-p)^2 r_2$ (corresponding to point C in Figure B1).

Case VI: $w=\gamma$

As seen from Table B1 ties can occur in this case. If the tie is always broken in the favor of user 1, we get similar results as Case V. On the other hand, if the ties are always broken in favor of user 2, the average service rates for user 1 and 2 are 0 and $pr_1 + (1-p)r_2$ (corresponding to point D in Figure B1). If the tie breaking rule favors user 1 with a probability p' and user 2 with a probability $1-p'$, by adjusting the value of p' we can get intermediate results (the green line in Figure B1) between these two extreme points (C and D in Figure B1).

Case VII: $w > \gamma$

In this case the time slots are always awarded to the user 2 as $wR_2(t) > R_1(t)$. So, the average throughput of user 1 is 0 while that of user 2 is $pr_1 + (1-p)r_2$ (point D in Figure B1).

From Proposition 1 (the 'tangent interpretation' of the minimal cost) we see that when the jobs are ordered such that c_{21} is provided to the bigger job and c_{22} is provided to the smaller job, C is the point which minimizes the aggregate flow delay. Thus we get

$$\begin{aligned}
G_1^* &= \frac{1}{pr_1 + (1-p)r_2}, \\
G_2^* &= \frac{1}{pr_1 + (1-p)^2 r_2} \left(2 - G_1^*(p(1-p)r_1)\right), \\
&= \frac{1}{pr_1 + (1-p)^2 r_2} \left(2 - \frac{p(1-p)r_1}{pr_1 + (1-p)r_2}\right), \\
&= \frac{p(1+p)r_1 + 2(1-p)r_2}{(pr_1 + (1-p)^2 r_2)(pr_1 + (1-p)r_2)}.
\end{aligned}$$

As $\dfrac{p(1+p)r_1 + 2(1-p)r_2}{(pr_1 + (1-p)^2 r_2)} > 1$, $G_2^* > G_1^*$ in this case.