The Dissertation Committee for Arjun Anand
certifies that this is the approved version of the following dissertation:

Schedulers for Next Generation Wireless Networks: Realizing QoE Trade-offs for Heterogeneous Traffic Mixes

Committee:

Gustavo de Veciana, Supervisor

François Baccelli

Sanjay Shakkottai

John Hasenbein

Haris Vikalo

Schedulers for Next Generation Wireless Networks: Realizing QoE Trade-offs for Heterogeneous Traffic Mixes

by

Arjun Anand,

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2018

Dedicated to my parents.

Acknowledgments

I would like to express my sincere gratitude to my PhD supervisor Prof. Gustavo de Veciana for his support and guidance during my PhD. His calm and systematic step-by-step approach towards problem solving has helped me a better researcher as well as a better person in life. I am extremely fortunate to have an advisor like him for my PhD.

I would also like to express my gratitude to my collaborator Prof. Sanjay Shakkottai for helping me with a very important part of my thesis. Working with him has been very exciting and his energy, enthusiasm, and passion for work has motivated me to work harder.

I would like to thank my friends and lab-mates who have made my stay in Austin fun and enjoyable. I would also like thank my family members for their invaluable support. Finally I would like my express gratitude to ECE Dept. and WNCG staff who were always approachable and very helpful in completing all the administrative processes during my course.

Schedulers for Next Generation Wireless Networks: Realizing QoE Trade-offs for Heterogeneous Traffic Mixes

Publication No. _____

Arjun Anand, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Gustavo de Veciana

In this thesis we will focus on the design of schedulers for next generation wireless networks which support application mixes, characterized by different, possibly complex, application/user Quality of Experience (QoE) metrics. The central problem underlying resource allocation for such systems is realizing QoE trade-offs among various applications/users given the dynamic loads and capacity variability they would typically see. In the first part of the thesis our focus is on applications where QoE depends on *flow-level* delay-based metrics. We consider system-wide metrics which directly capture both users' QoE metrics and appropriate QoE trade-offs among various applications for a wide range of system loads. This approach is different from the traditional wireless scheduler designs which have been driven by rate-based criteria, e.g., utility maximizing/proportionally fair, and/or queue-based packet schedulers which do not directly reflect the link between flow-level delays and users' QoE. In the second part of this thesis we address the key design challenges in

networks supporting Ultra Reliable Low Latency Communications (URLLC) traffic which requires extremely high reliability (99.999%) and very low delays (1 msec).

We will explore three different types flow delay-based metrics in this proposal, based on 1) overall *mean delay*; 2) *functions of mean delays*; and, 3) *mean of functions of delays*. We begin by considering minimization of mean flow delay for an M/GI/1 queuing model for a wireless Base Station (BS) where the flow size distributions are of the New Better than Used in Expectation + Decreasing Hazard Rate (NBUE +DHZ) type. Such a flow size distribution have been observed in real systems and we too validate this model based on collected data. Using a combination of analysis and simulation we show that our scheduler achieves good performance for users that might correspond to interactive applications like web browsing and/or stored video streaming and is robust to variations in system loads. Next we consider a generalization of this approach where we minimize a metric based on cost functions of the mean flow delays in a multi-class system where users/flows are classified based on their respective QoE requirements and each class's QoE requirement is modeled by its respective cost function. This approach helps us model QoE more accurately and gives us more flexibility in considering QoE trade-offs among heterogeneous user classes. We optimize two different metrics based on how we average the cost functions of delays, namely, *functions of mean delays*; and *mean of functions of delays*. The former can be used when users' experiences are sensitive to mean delays and while the latter can be used when user's experience is also sensitive to higher moments of delays, e.g., variance or soft thresholds on delay. Extensive simulations confirm the effectiveness of our proposed approaches at realizing various QoE trade-offs and

performance.

In 5G wireless networks URLLC traffic is expected to support many applications like industrial automation, mission critical traffic, virtual traffic etc, where the wireless network has to reliability transport small packets with very high reliability and low delays. We address the following aspects related to the system design for URLLC traffic, 1) quantifying the impact of various system parameters like system bandwidth, link SINR, delay and latency constraints on URLLC 'capacity'; 2) provisioning wireless system appropriately to meet URLLC Quality of Service (QoS) requirements; and, 3) designing efficient Hybrid Automatic Repeat Request (HARQ) schemes for transmitting small packets. Further, due the heterogeneity in delay requirements between URLLC and other types of traffic, sharing radio resources between them creates its own unique challenges. We develop efficient multiplexing schemes between URLLC traffic and other mobile broadband traffic based on preemptive puncturing/superposition of the mobile broadband transmissions by URLLC transmissions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Next generation wireless networks will support a large number of applications with heterogeneous Quality of Experience (QoE) requirements, for example in 5G networks enhanced Mobile Broadband traffic. For example, some applications may demand low latency, others may demand both low latency and less variability in delays, and some may require only a high throughput over large time-scales. Exploring possible schedulers for downlink traffic (and uplink) in cellular Base Stations (BS) in such a heterogeneous setting is a challenging task for network designers and that will be the main focus of this thesis.

This thesis can be broadly divided into two parts. The first part addresses the design of QoE-aware schedulers for mobile broadband traffic. Note that the mobile broadband traffic is a very heterogeneous traffic class which includes traffic from diverse applications like web traffic, video streaming, file downloads, etc. We focus on achieving complex QoE trade-offs among various types of mobile broadband applications sharing a network using flow based schedulers. In the second part of the thesis, we identify and address the major wireless system design challenges in supporting URLLC traffic. Further we also design schedulers for 5G networks which can efficiently multiplex enhanced Mobile Broadband(eMBB) traffic and URLLC

traffic via superposition/puncturing of eMBB traffic. The two parts are explained in detail in Sections 1.1 and 1.2, respectively.

1.1 QoE-Aware Schedulers for Mobile Broadband Traffic

Traditional wireless schedulers have been driven by rate-based criteria, e.g., utility maximization e.g., proportionally fair allocations, which balance the average[1] rates allocated to users and/or queue-based schedulers, which monitor packet queue lengths and/or waiting times, see [2] for a survey. However, the major drawback associated with rate and queue-based schedulers is that they do not directly optimize QoE of users that are primarily sensitive to flow delays. For example, several studies have shown that QoE of users depend on the delay experienced in observing content/downloading files, see [3–6]. QoE is only indirectly related to average rate and packet delays. Therefore, to address this drawback we will explore various types of *flow-aware* schedulers and optimize delay-based metrics measured at the time-scale of *flows*. Let us first define a 'flow'.

We shall refer to a flow as the basic data unit whose reception drives the user perceived QoE. For example, for interactive web browsing a flow could be the content of a web page a user requested, or in the case of a file download a flow is associated with the reception of a file. Identifying flows from data streams and classifying users based on the QoE requirements has received substantial attention in literature, see e.g., [7,8]. Various parameters such as source/destination port numbers, IP addresses,

[1] Averages may be computed in an exponentially weighted or moving window ways and thus on different time scales.

inter-packet time intervals etc., can be used to classify packets associated with flows. In this proposal, we shall assume that the scheduler has the required information to identify the beginning and the end of a flow in the data stream of each user, i.e., it is flow-aware.

Network operators can ensure good QoE for all users by optimizing flow level delay metrics for the entire cell/system. The system-wide metric should be such that when optimized one achieves the optimal trade-offs in QoE among various applications/users for a range of system loads and traffic patterns. We illustrate trade-offs in QoE with the help of an example. Consider a situation in which a BS scheduler has to deal with different congestion levels, e.g., this situation may arise in a BS which serves a residential area. The system load could be higher in the nighttime as compared to the daytime. If the system load is very high, then delays experienced by all flows will be higher, and therefore, the scheduler will have to prioritize delay critical interactive applications over delay insensitive elastic traffic. However, if the system load is really low, then the delay performance for all interactive applications may be good and hence, improving their performance any further will only result in marginal improvement of QoE. Hence, the spare resources could be utilized to enhance the performance of other applications. In this example, the overall system metric should capture the desired trade-offs for a range of system load. To that end we will explore three different delay-based metrics in this proposal: 1) based on mean delays; 2) based on functions of mean delays; and, 3) based on the mean of functions of delays. We explain each of these in more detail.

Mean delay is the simplest metric that one might consider optimizing. Mean

delay optimal schedulers give priority to short flows over long flows, assuming the flow sizes are known to the scheduler, e.g., Shortest Remaining Processing Time (SRPT) scheduling policy. However, in many systems the flow sizes may not be known. Instead perhaps only the distribution for flow's sizes can be measured. In such a setting mean delay optimal scheduling can be complex and simple heuristics are desirable. We will focus on this aspect in Chapter 2.

Minimizing mean delay can ensure good QoE for users with delay sensitive traffic if they tend to generate shorter flows than delay insensitive applications. However, this approach does not address the following two aspects of QoE optimization:

1. QoE may be a non-linear function of the delay experienced by users. For example, for web browsing, it has been shown that users do not perceive any degradation in QoE if the flow delay is less than a certain threshold [9].

2. Different applications may have different sensitivities to delay. Also, it may not be the case that short flows are more sensitive to delay than larger ones. For example, machine-to-machine traffic might generate short flows, but may tolerate larger delays as opposed to stored video streaming (e.g. YouTube, Netflix etc.), which may generate comparatively larger flows but be less tolerant to delays.

To address these drawbacks we will consider minimizing cost functions of flow delays. We shall assume that the scheduler can classify flows into various classes, e.g., application types, with possibly different QoE models, i.e., we have a multi-class system. In this setting we can assume each type of application/user has an

associated cost which is a function of delays experienced by its flows. This can be set/designed by the network operator to reflect user perceived QoE models. The cost function can possibly be a non-linear function of delay. In fact, the cost function could be interpreted as the inverse of the QoE perceived by the user, i.e., lower the cost, the better the QoE. Since larger delays generally tend to result in a poorer user experience, cost functions would naturally be non-decreasing functions of delays. One can set cost functions for applications based on their sensitivity to delays, for example, for a delay sensitive application we can choose very 'steep' function of delay which increases sharply after tolerable delay is exceeded. One can then define a cost for the overall system by appropriately scaling and adding the cost functions for various applications/classes.

A natural question which arises when we minimize delay-based cost functions is whether one should minimize the *mean of functions of delays or functions of mean delays*. One way to answer this question is that functions of mean delays can be used when the user's QoE is primarily driven by the first moment of the user experienced delay distribution, whereas, mean of functions of delays would be useful for settings in which QoE depends on higher moments of the delay, for example, a user may be sensitive to both the mean and the variability of the delays experienced, or may care about delays exceeding a given threshold. In general, the setting where we consider functions of mean delays is more analytically tractable than that where we consider the mean of functions of delays. We will explore both. Next we will summarize the key contributions of this part of thesis.

1.1.1 Contributions

We will divide our work into three parts based on the metrics we choose to optimize, namely: 1) mean delay; 2) functions of mean delays; and, 3) mean of functions of delays.

1) Mean Delay: In this work we consider schedulers geared at minimizing the mean delay experienced by a typical flow in the network when the scheduler knows *only* the flow size *distribution* of the mix of traffic sharing the BS. This is a reasonable assumption when the scheduler does not have enough information about higher protocol layers like transport and application layers. The scheduler may be able to detect the beginning of a flow but may not know the total number of bits in the flow until the flow has been serviced to completion. We model the BS using a simple M/GI/1 queuing model. Using empirical data, we observe that the typical flow size distribution seen in wireless networks is NBUE + DHZ, i.e, it is a mixture of New Better than Used in Expectation (NBUE) and Decreasing Hazard Rate (DHZ) distributions. When the scheduler knows only the distribution of the flow sizes, then it is known that Gittins index scheduler is mean delay optimal, see [10]. Such schedulers, however can be somewhat complex to implement, so we propose a practical approximation for the Gittins index scheduler when the distribution of flow sizes belong to NBUE + DHZ class. Using a combination of analysis and simulation we explore the QoE trade-offs such a scheduler could achieve under different mixes of traffic, in particular: 1) mobile web browsing and small file delays; 2) stored streaming video quality vs re-buffering; 3) throughput of larger file downloads. The results suggest improved QoS/QoE trade-offs vs traditional proportionally fair schedulers

which are robust to changes in the network load. These results are presented in Chapter 2.

2) Functions of Mean Delays: Next we consider a multi-class M/GI/1 queuing system in which users/flows are classified based on their respective QoE sensitivity or application type. We minimize an overall system-wide cost function corresponding to a weighted sum of functions of mean delays of all classes. The weight of each class is assumed to be proportional to the class's flow arrival rate. Once again we will assume that the scheduler knows the flow size distribution but this on a per class basis. We develop a measurement-based scheduling policy which learns the arrival rates and the delays experienced by flows and adapts the scheduler so as to optimize the system-wide cost under the current load and traffic mix. We shall refer to the resulting scheduler as a Measurement Based Delay Optimal (MBDO) scheduler. We show that under mild assumptions, and in a stationary regime, that MBDO scheduling is asymptotically optimal. Our extensive simulations confirm the effectiveness at realizing trade-offs and performance of the proposed approach. We will describe MBDO scheduler in detail in Chapter 3.

3) Mean of Functions of Flow Delays: In this work we explore resource allocation strategies geared at optimizing the expected value of functions of delays. Similar to the previous work we consider a multi-class system with possibly different cost functions for different user classes. This final setting is more complex than the previous two cases so we will narrow our focus to the following two settings: 1) scheduler knows the flow size realizations; 2) flow sizes are drawn from an exponential distribution and scheduler knows the mean flow size. We further will begin

Metric Used	Information Available on Flow Sizes	Multi-Class System
Mean delay	Distribution for the entire mixture	No
Functions of mean delays	Distribution of each class	Yes
Mean of functions of delays	Realizations or Exponential with known mean	Yes

Table 1.1: Summary of our work on QoE-aware schedulers

by considering a transient setting where the number of users is fixed and there are no further user arrivals. In this setting the problem can be modeled as a Restless Multi-Armed Bandit (RMAB). The exact solution to this problem is unfortunately still analytically intractable. We thus develop a heuristic index policy, called as Opportunistic Delay Based Index Policy(ODIP), based on Whittle's relaxation, which is known to work well in practice for RMAB problems. With these simplifications, we are finally able to propose using this heuristic for dynamic settings which permit user arrivals to the system. Simulations confirm the effectiveness of the proposed approach. More details of this work and performance evaluation are given in Chapter 4. Table 1.1 summarizes the various settings considered in this section.

1.2 URLLC traffic: System Design Principles and Resource Sharing with eMBB Traffic

5G wireless networks are expected to support Ultra Reliable Low Latency Communications (URLLC) for applications like industrial automation, mission critical traffic, virtual reality, etc., see e.g., [11–16]. The design of wireless systems subject to the stringent requirements of URLLC traffic is a challenging task. In Chapter 5 we will answer the following key questions pertaining to a wireless system supporting URLLC traffic.

1. What is the impact of system bandwidth, packet sizes, SINR, and reliability and latency requirements on URLLC 'capacity'?

2. What are the optimal choices of bandwidth and transmission duration for URLLC transmissions?

3. What is the impact of Forward Error Correction (FEC) and Hybrid Automatic Repeat Request (HARQ) schemes on URLLC 'capacity'?

We will discuss about these questions briefly here. The first question is fundamental in nature and it helps network designers provision wireless systems appropriately. To elaborate on the second question, note that 5G networks are based on Orthogonal Frequency Division Multiple Access (OFDMA) based systems, where different transmissions are allocated different parts of a time-frequency plane. To send a URLLC packet, we can use a 'tall' transmission which uses a large bandwidth for a short duration or a 'wide' transmission, i.e., small bandwidth over a longer duration. If we use a 'tall' transmission, the number of concurrent transmissions possible will decrease which may affect the capacity for concurrent transmissions. However, a 'wide' transmission will take longer to complete and reduce the number of re-transmissions possible before the delay deadline expires. Hence, it may be desirable to implement a robust coding (with more redundancy bits) for 'wider' transmissions. We require an analytical framework to capture and optimize trade-offs between 'tall' and 'wide' transmissions.

A characterization of the impact of FEC and HARQ on URLLC capacity is important because one can then optimize the FEC and HARQ schemes to maximize

the URLLC capacity/spectral efficiency. For example, one can optimize the required number of re-transmissions to meet a reliability target and the probability of decoding failure in each transmission. The maximum number of re-transmissions is constrained by the deadline d. Once the target decoding failure probability is known for each stage one can then choose the coding rate appropriately which in turn affects the capacity of the system. To summarize, wireless system design for URLLC traffic has to tackle the complex dependencies between system bandwidth, SINR, reliability and latency constraints, resource allocation schemes, and FEC and HARQ mechanisms.

In many practical systems URLLC and eMBB traffic share a Base Station. Hence, it is of interest to develop efficient mutliplexing strategies for both URLLC and eMBB traffic. One possible solution is to have dedicated frequency bands for URLLC and eMBB traffic. However, few authors have observed that this approach leads to a low resource utilization, see [17, 18]. They have suggested a wide-band resource allocation for URLLC traffic where the entire system bandwidth is dynamically shared between eMBB and URLLC traffic without any dedicated bands for each traffic type. Further, the 3GPP standards body has proposed an innovative superposition/puncturing framework for multiplexing URLLC and eMBB traffic in a wide-band setting which is described briefly below and in a detailed manner in Chapter 6.

The proposed scheduling framework has the following structure [15]. As with current cellular systems, time is divided into slots, with proposed one millisecond (msec) slot duration. Within each slot, eMBB traffic can share the bandwidth over the time-frequency plane (see Figure 6.1). The eMBB shares are decided by the

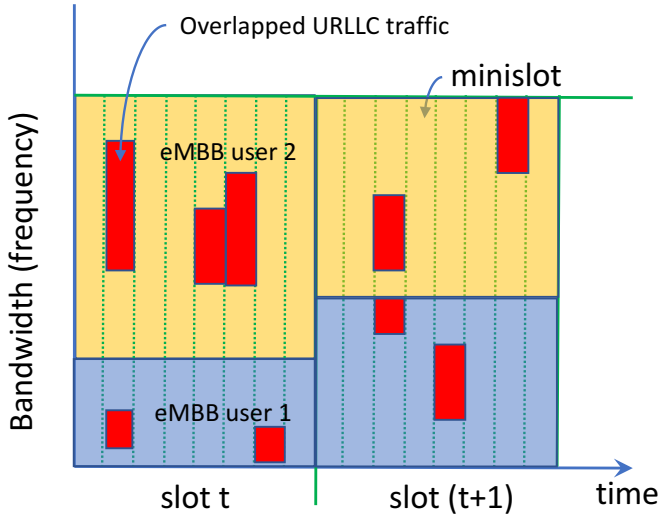


Figure 1.1: Illustration of superposition/puncturing approach for multiplexing eMBB and URLLC: Time is divided into slots, and further subdivided into minislots. eMBB traffic is scheduled at the beginning of slots (sharing frequency across two eMBB users), whereas URLLC traffic can be dynamically overlapped (superpose/puncture) at any minislot.

beginning, and fixed for the duration of a slot.

URLLC downlink traffic may arrive during an ongoing eMBB transmission; if tight latency constraints are to be satisfied, they cannot be queued until the next slot. Instead each eMBB slot is divided into minislots, each of which has a 0.125 msec duration. Thus upon arrival URLLC demand can be immediately scheduled in the next minislot *on top of the ongoing eMBB transmissions.* If the Base Station (BS) chooses non-zero transmission powers for both eMBB and overlapping URLLC traffic, then this is referred to as *superposition.* If eMBB transmissions are allocated zero power when URLLC traffic is overlapped, then it is referred to as *puncturing* of eMBB transmissions. The superposed/punctured URLLC traffic is sufficiently protected

(through coding and HARQ if necessary) to ensure that it is reliably transmitted. At the end of an eMBB slot, the BS can signal the eMBB users the locations, if any, of URLLC superposition/puncturing. The eMBB user can in turn use this information to decode transmissions, with some possible loss of rate depending on the amount of URLLC overlaps.

A key problem in this setting is thus the *joint scheduling of eMBB and URLLC traffic over two time-scales.* At the slot boundary, resources are allocated to eMBB users based on their channel states and utilities, in effect, allocating long term rates to optimize high-level goals (e.g. utility optimization). Meanwhile, at each minislot boundary, the (stochastic) URLLC demands are overlapped (superposed/punctured) onto previously allocated eMBB transmissions. Decisions on the placement of such overlaps across scheduled eMBB user(s) will impact the rates they will see on that slot. Thus we have a coupled problem of jointly optimizing the scheduling of eMBB users on slots with the placement of URLLC demands across minislots. Solutions to this joint scheduling problem are derived in Chapter 6. Next we shall summarize the major contributions of this part of the thesis.

1.2.1 Contributions

In Chapter 5, we consider a holistic approach towards the design of wireless systems supporting URLLC traffic where we study the impact of QoS requirements, resource allocation schemes and physical layers aspects like the choice of HARQ and FEC schemes on the URLLC 'capacity'. We develop an analytical model based on Jackson queuing networks which captures the essential properties of such a system.

The key contributions in this chapter are summarized below.

1. We derive the scaling results of URLLC 'capacity' with respect to system bandwidth, SINR, and QoS requirements.

2. We prove that 'wide' transmissions which spreads out the transmission as wide as possible in the time domain without violating latency constraints are better than 'tall' transmissions in terms of URLLC capacity.

3. We optimize FEC and HARQ schemes to maximize spectral efficiency. We show that at low URLLC loads, the optimal solution is a *one-shot* transmission meeting the desired reliability target without any further re-transmissions, and at high URLLC loads, the optimal solution permits re-transmissions if needed. Further, the maximum number of permitted re-transmissions is a non-increasing function of SINR.

In Chapter 6, we solve the joint eMBB/URLLC scheduling problem described previously with the dual objectives of maximizing utility for eMBB traffic while satisfying instantaneous URLLC demands. For a linear rate loss model (loss to eMBB is linear in the amount of superposition/puncturing), we derive an optimal joint scheduler. Somewhat counter-intuitively, our results show that our dual objectives can be met by an iterative gradient scheduler for eMBB traffic that anticipates the expected loss from URLLC traffic, along with an URLLC demand scheduler that is oblivious to eMBB channel states, utility functions and allocations decisions of the eMBB scheduler. Next we consider a more general class of (convex/threshold) loss

models and study optimal online joint eMBB/URLLC schedulers within the broad class of channel state dependent but time-homogeneous policies. We validate the characteristics and benefits of our schedulers via simulation.

Chapter 2

Mean Delay Minimization Using Context-Aware Schedulers

2.1 Introduction

In this chapter[1] we will discuss our approach of using a practical approximation of mean delay optimal scheduler to realize key QoE trade-offs between three different types of applications sharing the network: 1) interactive web browsing; 2) stored video streaming; and, 3)large file downloads. Each of these applications has its own specific QoE requirements, making this problem a challenging task.

There are three interrelated challenges in developing resource allocation strategies for such heterogeneous systems. First, the impact of resource allocation on an application's Quality of Service (QoS) or user's Quality of Experience (QoE) can be quite different, and in some cases may even be hard to characterize all together, e.g., video QoE. Second, wireless systems are subject to substantial temporal variability and spatial heterogeneity in capacity. Indeed, even for stationary users wireless channel capacity can fluctuate, while exhibiting drops of several orders of magnitude from the cell's 'center' to its 'edge.' Further, in practice the number of active users

[1]Publications based on this chapter: [19] A. Anand and G. de Veciana, "Invited paper: Context-aware schedulers: Realizing quality of service/experience trade-offs for heterogeneous traffic mixes", in Proceedings of WiOPT, 2016.

can change dramatically as they join, move and leave, and the overall network loads and traffic mixes can vary throughout the day. The third challenge is managing trade-offs amongst heterogeneous traffic mixes, particularly when the network becomes congested – i.e., how to optimize a *graceful degradation* in QoS/QoE when resources become scarce.

This third challenge associated with trade-offs is really the crux of the problem underlying scheduler design and yet is poorly understood and poorly reflected in state-of-the-art schedulers. Let us illustrate this via several examples:

1) Web browsing vs large file downloads. Web browsing sessions involve human interaction on the order of seconds, so the QoE metric of interest is maintaining responsiveness, i.e., delays on the order of seconds to download the typically small files associated with web content for mobile devices. By contrast, large files take a long time, so one might posit the relevant QoS metric is long term throughput. Clearly a scheduler that prioritizes small files associated with web browsing and other applications, over large files achieves a good QoS/QoE trade-off for the mix.

2) Video QoE management at congested base stations. Modern stored video streaming protocols, such DASH (Dynamic Adaptive Streaming over HTTP), are rate adaptive, i.e., they adapt the video rates, and associated quality, to network congestion an/or the risk of playback re-buffering. Consider a setting where a base station serves users with heterogeneous capacity (center/edge users) via a proportionally fair scheduler, i.e., allocations which are directly proportional to the user's capacity. For light to moderate base station loads edge users might see reduced video quality vs those at the cell center, which is reasonable. Under high

	Stored Video Streaming		**Web browsing/Small files**	**Large files**
Network load	Video quality	Re-buffering	Mean flow delay	Mean throughput
Low	High	Low	Low	High
Medium	Medium	Low	Low	High
High	Low	Low	Low/Medium	Medium

Table 2.1: Scheduler design objectives: QoS/QoE trade-offs across applications vs network loads.

loads, however, edge users will start to see playback re-buffering, i.e., QoE which is unacceptable. Thus for congested resources the scheduler should be more aggressive in shifting resources from cell center to edge users.

The above exemplify some of the complex trade-offs base station schedulers need to make across heterogeneous applications. Realizing such trade-offs through the design and analysis of *context-aware* schedulers is the focus of this chapter. This involves studying schedulers that realize QoS/QoE trade-offs objectives across applications for different traffic mixes and network loads. Table 2.1 exhibits an example of the high-level goals we aim to achieve for a mix of stored video streaming, web browsing, and file transfers.

We shall focus on the following natural QoS/QoE metrics which represent a simplification of the more complex models discussed further in the related work.

1) Mean delay for small flows. Most small flows are currently due to web traffic, for which the overall transfer delay (time to display) is the key goal. It is of interest to limit such delays to less than a second, to maintain interactivity, but further speedups are not of much value. Further, ideally these delays should not be too sensitive to other network loads, e.g., video streaming, large files etc.

2) Video quality and re-buffering for stored video streaming. The first

priority is to avoid client re-buffering, beyond this one would like to achieve good average video quality depending on the load and the users' channel condition.

3) Throughput for large files. It is reasonable for large files to see delays proportional to their size as such one would expect to the perceived throughput to be the relevant metric, though it might be affected by the overall system load and mix of traffic.

Before we discuss our work in more detail, let us put it into proper context based on the substantial previous work considering base station scheduling from different perspectives.

2.1.1 Related work.

Modeling QoS/QoE. Traditional QoS metrics such as throughput, packet delays and jitter, have been found to only poorly reflect user experience. For this reason there has been significant interest in better modeling user perceived QoE for various applications. For example, for interactive web browsing, QoE was found to be well modeled as a function of the delay of transactions, see [3, 20]. In particular [3], web browsing QoE as an S-shaped function of transaction delay, whereas [20], propose polynomial functions of transaction delays. These, and other, recent efforts reinforce the need to look at QoE metrics depending on flow (transaction) delays. Perhaps the simple lesson learned here is that one would like to see small transaction delays, below some level, but further reductions do not have a high marginal benefit. We shall embrace this principle. Similarly, there has been substantial recent interest in

modeling streaming video QoE including aspects of the quality of the reproduced video, possibly quality variability, re-buffering, and start up delays, see [21] and references therein. In general there is agreement that avoiding re-buffering is the first priority if one is to improve user perceived QoE, see [5].

Scheduling. Traditional work focused on scheduling for elastic traffic[2] focused on 'fair' rate allocation by using utility maximization approaches in the full buffer model, see e.g., [2,22,23] for detailed surveys. In general this fails to directly account for the dynamic nature of traffic and indeed the flow-level delays that translate to user perceived QoE.

There is also substantial work on queue-based schedulers addressing stability and/or QoS for real-time traffic, e.g, VoIP in LTE networks. Most of this work augments the utility-based schedulers such as proportionally fair (PF) with the current queue lengths of users, see e.g., [2]. A weakness of this work remains the lack of focus on flow level metrics and and ability to multiplex and control performance when there are user dynamics.

Another area of substantial research is network scheduling and transport for modern DASH-like video streaming,see e.g, [24–26]. In general these works starve to optimize the video client behavior as well BS/core network scheduling to video QoE with constraints on re-buffering time, or fraction of time low quality video is deliver. These works do do not fully address the impact of flow level dynamics and in particular the sharing of resources by heterogeneous applications. Still in the

[2]Traditionally interactive web browsing, large file downloads, emails etc are classified into a single category called best effort elastic traffic.

sequel we shall adopt [24] as a representative mechanism to assess our context-aware scheduler.

Finally, there has been some work on scheduling to address flow-level delays which draws from a rich body of work in queuing theory, see e.g., [27–34]. These works address the minimization of average flow delay for traffic having a a mix of small and large flows, so called mice and elephants. It is well known that if a scheduler knows the required processing time of flows, the Shortest Remaining Processing Time policy minimizes the mean delay, see e.g., [33]. If such information is not available, scheduler may infer this based on cumulative service to date and/or use prior knowledge of the flow size distribution. This is represented by schedulers such as the Foreground-Background (FB) or Least Attained Service (LAS), Multi-Level processor sharing, FCFS + FB, etc which are delay optimal in various settings depending on the flow-size distribution, see e.g. [28–30, 33]. This above work for the most part does not address wireless networks where different flows may see heterogeneous and/or changing wireless capacity.. Exceptions include downlink scheduling studied in [27, 35]. We will draw on this previous theoretical work in developing our own approach and in our effort to tackle QoS/QoE trade-offs across heterogeneous traffic.

2.1.2 Our Contributions

In this chapter we recognize that for many applications the QoS/QoE is tied to flow-level performance. For web browsing sessions, flows are associated web pages that are being downloaded. Similarly modern stored video streaming can be viewed

as a stream of 'flows' associated with video segments whose size is being adapted to network congestion. Thus the QoE for video depends on the delays/arrivals for the associated stream of flows. We propose a two-level framework for context-aware scheduling. The upper block, called the *flow classifier*, realizes context-aware decisions, regarding applications flows and possible trade-offs e.g, managing re-buffering amongst video streams. The lower block, implements a flow- and channel-aware scheduling algorithm, aimed at reducing delays for small flows without requiring prior knowledge of their size. To that end we study the characteristics of mean delay optimal Gittins index scheduler for an idealized model for a wireless BS serving users with heterogeneous capacity and for a class of distributions found on today's networks. Extensive simulations are used to compare our context-aware scheduler to traditional proportional fair scheduler. In particular we show that our approach is able to achieve the desired trade-offs (see Table 2.1) in QoS/QoE amongst streaming video, web browsing and large file transfers and do so robustly over a range of network loads.

2.1.3 Organization

The chapter is organized as follows. In Sec. 2.2 we present the architecture of our context-aware scheduler. Its design and analysis are explained in Sec. 2.3. In Sec. 2.4 we discuss some practical implementation aspects of using TCP like transport protocols with our scheduler. Performance analysis through simulations are explained in Sec. 2.5, followed our conclusions in Sec. 5.5.

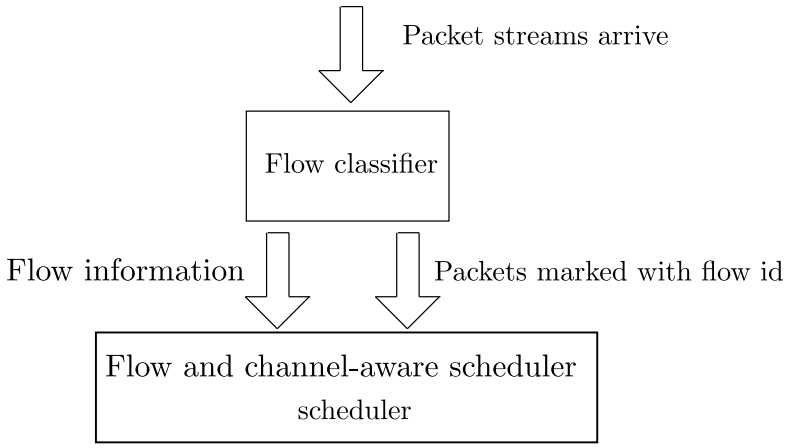


Figure 2.1: The block diagram for our context-aware scheduler.

2.2 Context-Aware scheduler

Our context-aware scheduler consists of two modules, namely, the flow classifier and the flow and channel-aware scheduler. The block diagram is shown in Figure 2.1. We describe the two blocks in detail.

2.2.1 Flow classifier

Packet streams arrive to the flow classifier block which realizes context-aware decisions. This block may be implemented at the BS itself or in the core network. Its main functions are:

1) Manage flow information. It distinguishes flows based on their application type and marks the packets of a flow with a unique flow id. This information is later used by the scheduler block. Also, it may decide when a flow has completed based on a threshold for the gaps in inter-packet arrivals. It exchanges control signals and flow level information with the scheduler, for example, to signal the initiation

of a new flow. It may also gather meta-data associated with the flows which may be shared with the scheduler, e.g., video segment playback duration.

2) Ensure video QoE. We envisage a flow-classifier that is actively managing video QoE. In particular, it has to ensure sustained playback for all video clients without re-buffering. To that end, it has ensure that video streams are not starved of resources by the scheduler block. We assume that all video users are continuously watching the video. Otherwise, the video clients stop requesting new segments and our flow classifier detects that streaming has completed using inter-packet delays. We consider a simple strategy to prevent re-buffering. The flow classifier samples the *deficit* of video streams whenever a flow completes service.[3] Let $\mathcal{N}$ be the set of video streams in the system. Let τ_i, $i = 1, 2, \ldots$ be the instants at which flows complete service. If $s_i\left(t_1, t_2\right]$ is the total number of segments downloaded by video stream i between time t_1 and t_2, then the deficit for the ith stream $d_i\left(\tau_k\right)$ is defined as

$$d_i\left(\tau_k\right) := \max\left\{d_i\left(\tau_{k-1}\right) + \tau_k - \tau_{k-1} - \tau_{\text{seg}}\ s_i\left(\tau_{k-1}, \tau_k\right], \overline{\gamma}\right\}, \tag{2.1}$$

where τ_{seg} is the video playback duration of a segment and $\overline{\gamma} \leq 0$ is a suitably chosen threshold. A positive deficit at any time means that the number of segments downloaded until then is not sufficient for sustained playback, and the video client is in re-buffering state. A negative $\overline{\gamma}$ puts a more stringent constraint on re-buffering. Let $\mathcal{D}_i(\tau_k)$ be the set of flows for which the deficit is strictly greater than $\overline{\gamma}$ at time τ_k. If $\mathcal{D}_i(\tau_k)$ is non-empty, then the flow classifier block disables the set of flows

[3]Video segments are marked as flows by flow classifier.

$\mathcal{N} \setminus \mathcal{D}_i(\tau_k)$ till τ_{k+1}, i.e., the flows in the set $\mathcal{N} \setminus \mathcal{D}_i(\tau_k)$ do not contend for the radio resources in the next $\tau_{k+1} - \tau_k$ seconds. This ensures that the deficient video streams are given priority over the streams which have sufficient segments in the playback buffer.

2.2.2 Flow and channel-aware scheduler

This block allocates the radio resources to flows. The scheduling policy specifies which flows are to be served at each slot. It may use the current Channel Quality Indicator metric (CQI) of users with active flows and/or the flow state information, e.g., the amount of service given to a flow. We discuss its design and analysis in the next section.

2.3 Design and analysis of flow and channel-aware scheduler

2.3.1 Idealized queuing model

To devise our flow and channel-aware scheduler we shall revisit an idealized queuing model based on the multi-class M/GI/1 queue. If the slot duration at which the BS makes scheduling decisions is quite small when compared to the transmission time of a typical flow, then a continuous time queuing system is a good approximation for the BS.

Arrival process. Given the possibility of a large diverse set of independent active flows, we shall model the arrival process of flows to the system as a Poisson process of appropriate rate. These flows are associated with users having different channel strengths and/or Signal to Interference and Noise (SINR) ratios. In many

wireless systems like LTE-Advanced, IEEE 802.11 ac etc., the BS can support only a pre-determined discrete set of transmission rates for users. We classify users into K distinct classes based on their current transmission rates. The rate of arrival for each class is given by λ_i, $i = 1, 2, \ldots, K$. Let c_i be the transmission rate for the ith class and let $c_1 < c_2 \ldots, c_K$. We assume, for now, that a flow's transmission rate remains fixed throughout its lifetime. However, class changes can be easily incorporated into our scheduling algorithm – this is addressed in Sec. 3.5.

Flow size distribution. Our scheduler sees a heterogeneous mix of flows associated with interactive web traffic and small to large file downloads. Therefore, from a statistical point of view, the scheduler sees a concentration short and medium sized flows and few large flows. This property is very well captured by the NBUE + DHZ (β) class of flow size distributions. We will explain this in detail.

Let X denote the random variable (r.v.) associated with the flow size. Let $G_X(x)$, $g_X(x)$, and $\overline{G}_X(x)$ be the cumulative distribution function (c.d.f.), probability density function (p.d.f.), and complementary c.d.f. (c.c.d.f.) of the flow size, respectively. We assume that the c.d.f. is a continuous function of the flow size. Define hazard rate function $h_X(x) := \frac{g_X(x)}{\overline{G}_X(x)}$. A distribution is said to be of type NBUE + DHZ (β) if:

1. When the flow size is less than β bits, then the distribution is of the type New Better Than Used in Expectation (NBUE), i.e., the expected residual size of a flow which has attained service less than β bits is less than the original

expected size of the flow. This implies that $\forall a \leq \beta$,

$$\mathbb{E}[X] \geq \mathbb{E}[X - a | X > a]. \tag{2.2}$$

2. When the flow size is more than β bits, then the flow size distribution has Decreasing Hazard Rate (DHZ). This means that $h_X(x)$ is decreasing function of x for $x > \beta$. The DHZ property is a sufficient condition for a distribution to have an increasing mean residual file size.

An example of a distribution which is NBUE + DHZ (β) is the *Exp. + Pareto* distribution which is given below:

$$\overline{G}_X(x) = \begin{cases} \exp(-\mu x), & x < \beta, \\ \exp(-\mu\beta)\left(\frac{\beta}{x}\right)^{\alpha}, & x \geq \beta, \end{cases} \tag{2.3}$$

where $\mu > 0$ and $\alpha > 1$. If $\mu = 0$, then (2.3) reduces to normal Pareto distribution. More examples are given in [28].

Our preliminary exploration of measured data in [36] shows that it is nicely modeled using distributions NBUE + DHZ (β) distributions with Pareto tail and they are analytically tractable. Figures 2.2 and 2.3 plot the cumulative distribution function (c.d.f.) of the flow sizes obtained from Google and Flickr, respectively, by the authors in [36]. The c.d.f. in Fig. 2.2 is curve fitted by Pareto distribution with parameters $\alpha = 1.01$ and $\beta = 0.6$ KBytes, whereas the c.d.f. in Fig. 2.3 is curve fitted by Exponential + Pareto distribution with parameters $\alpha = 0.55$, $\beta = 2.742$ KBytes, and $\mu = 1.02$. Therefore, in this chapter we mainly consider distributions with Pareto tail and we call them NBUE + Pareto (α, β).

Next we discuss about mean delay optimal scheduling policy.

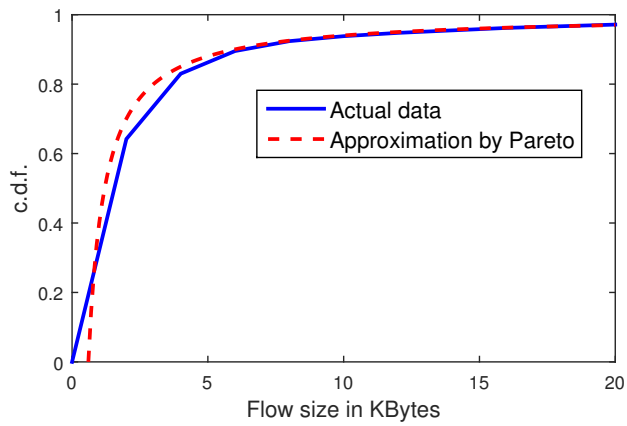


Figure 2.2: The c.d.f. of flow size for traffic data obtained from Google. The c.d.f. is closely approximated by Pareto distribution with parameters $\alpha = 1.01$ and $\beta = 0.6$ KBytes.

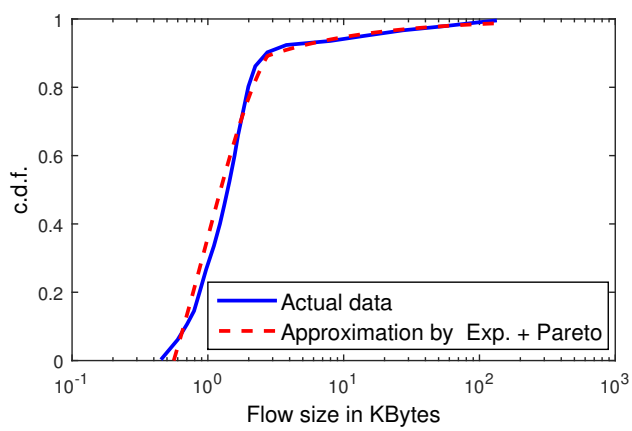


Figure 2.3: The c.d.f. of flow size for traffic data obtained from Flickr. The c.d.f. is closely approximated by Exponential + Pareto distribution with parameters $\alpha = 0.55$, $\beta = 2.742$ KBytes, and $\mu = 1.02$.

2.3.2 Mean delay optimal policy

When flow sizes are not directly available, the Gittins index scheduling policy minimizes the expected delay in an M/GI/1 queuing system [32]. Below we shall introduce the Gittins index and discuss some of its important properties derived in [28, 29]. We use these properties to derive the optimal scheduling policy for our multi-class wireless setting which we consider in this chapter.

Gittins Index. Consider an M/GI/1 queuing system which serves flows at unit rate. This means that a flow of size x bits will take x seconds to complete service. Consider a flow which has already been served a bits. Define $J(a, \Delta)$ for $\Delta \geq 0$ as

$$J(a, \Delta) := \begin{cases} \frac{\overline{G}_X(a) - \overline{G}_X(a+\Delta)}{\int_0^\Delta \overline{G}_X(a+t)dt}, & \text{if } \Delta > 0, \\ h_X(a), & \text{if } \Delta = 0. \end{cases} \tag{2.4}$$

The above expression is the ratio of the probability that a flow which has attained service of a bits will complete and the expected additional time required by the flow to complete when it is given a service time of Δ seconds. Therefore, $J(a, \Delta)$ is the ratio of expected *reward* to the expected *cost* of giving a service of Δ seconds to a flow that has already attained a bits of service until now.

The Gittins index for such a queuing system is defined in [10] and given by

$$\mathcal{G}_X(a) = \sup_{\Delta \geq 0} J(a, \Delta). \tag{2.5}$$

There may be many values of Δ that maximize the above expression with a possible value of $+\infty$ too. We define $\Delta^*(a)$ as

$$\Delta^*(a) = \sup_{\Delta \geq 0} \{\Delta : J(a, \Delta) = \mathcal{G}_X(a)\}. \tag{2.6}$$

If a scheduler is such that it serves the flow achieving the highest Gittins index at all times, then such a scheduler is called as the *Gittins index scheduler*.

We summarize the important properties of the Gittins index for distributions of NBUE + DHZ (β) type which were derived in [28, 29].

Proposition 2.3.1. *Properties of $\mathcal{G}_X(\cdot)$ for NBUE + DHZ (β) distribution are:*

(a) $\Delta^*(0) \geq \beta$.

(b) For all $a < \Delta^*(0)$, $\mathcal{G}_X(a) \geq \mathcal{G}_X(0)$.

(c) For all $a \geq \beta$, $\mathcal{G}_X(a)$ *is decreasing and* $\mathcal{G}_X(a) = h(a)$.

(d) If $h_X(x)$ *is continuous and* $0 < \Delta^*(0) < \infty$, *then* $\mathcal{G}_X(0) = \mathcal{G}_X(\Delta^*(0)) = h(\Delta^*(0))$.

Comments. The points (a) and (b) above imply that if a flow which has not received any prior service is selected for service, it would receive $\Delta^*(0) \geq \beta$ seconds of server time. Once it begins service, other flows in the system which have not received any service previously would not preempt it. Property (c) implies that the Gittins index is a decreasing function of x, for $x > \beta$. This is because of the DHZ tail which makes it less beneficial for the system to serve large flows.

Next we discuss the Gittins index scheduler for our wireless BS model based on multi-class class M/GI/1 queue.

2.3.3 Optimal scheduler for multi-class M/GI/1 queuing system

Consider the multi-class M/GI/1 queuing model for the BS. A flow of size x bits in i^{th} class requires x/c_i seconds of server time. Therefore, the mean service time associated with a flow in i^{th} class is $\mathbb{E}[X]/c_i$. For now we shall assume that at any time $t \geq 0$ only one flow is scheduled for transmission using the entire bandwidth available. A scheduling policy specifies which flow is to be scheduled at each slot for any sample path of the arrival process.

Before we derive the optimal Gittins index scheduler for this model, we consider the Gittins index for our multi-class system. The Gittins index in this setting depends on both the class of the flow and the attained service by the flow. This is because when the server allocates Δ seconds of service time to a flow, the probability that it completes service within the Δ seconds and the expected time it takes to complete service depend on the transmission rate of its class. We shall express the Gittins index of a flow in i^{th} class, $\mathcal{G}_i(\cdot)$, in terms of the Gittins index $\mathcal{G}_X(\cdot)$ associated with an M/GI/1 system where flows are served at unit rate.

Lemma 2.3.2. *Suppose a flow of class i has attained x bits of service, then its Gittins index $\mathcal{G}_i(\cdot)$ is given by:*

$$\mathcal{G}_i(x) = c_i \mathcal{G}_X(x). \tag{2.7}$$

Proof. By the definition of Gittins index, we have

$$\mathcal{G}_i(x) = \sup_{\Delta>0} \frac{\overline{G}_X(x) - \overline{G}_X(x + c_i\Delta)}{\int_0^{\Delta} \overline{G}_X(x + c_i t)\, dt}, \tag{2.8}$$

$$= c_i \sup_{\tilde{\Delta}>0} \frac{\overline{G}_X(x) - \overline{G}_X\left(x + \tilde{\Delta}\right)}{\int_0^{\tilde{\Delta}} \overline{G}_X(x + \tau)\, d\tau}, \tag{2.9}$$

$$= c_i \mathcal{G}_X(x), \tag{2.10}$$

where $\tilde{\Delta} = c_i\Delta$. □

The Gittins index scheduler requires the exact knowledge of the index as a function of the service given to a flow. Thus in order to compute the Gittins index we require the knowledge of the distribution of flow sizes. This information may not available in practice. Therefore, we require a robust approximation to the Gittins index scheduler which is based on easily measurable statistical properties like the mean flow size. In the sequel we discuss some of the key characteristics of the Gittins index scheduler which will be used to motivate our design approximations to the optimal Gittins index scheduler.

2.3.4 Qualitative characteristics of the optimal scheduler

Figure 2.4 shows a typical plot of the Gittins index curves for a system with three different classes of users. Define $\theta := \Delta^*(0)$. We call θ as the *cross-over threshold*. Later in this section, we will see that the Gittins index policy treats flows that have received less than θ bits of service and more than θ bits of service differently. We use this property to develop our approximation to Gittins index scheduler.

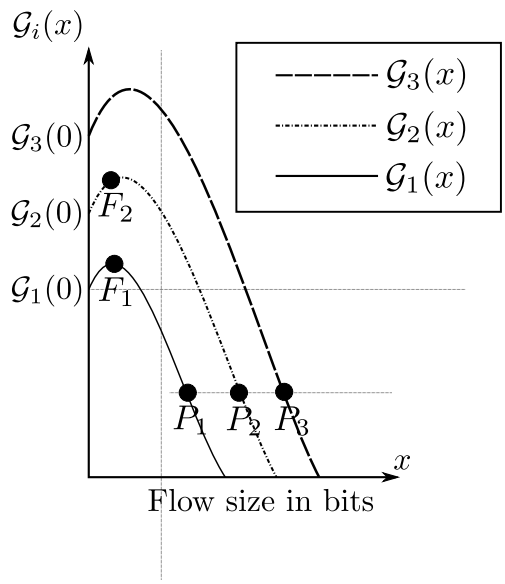


Figure 2.4: Illustration of Gittins index curves as function of the flow size for a multi-class M/GI/1 queuing system.

Fig. 2.4 illustrates all the properties of Gittins index mentioned in Prop. 2.3.1 and in Lemma 2.3.2. At any given time, the states of flows present in the system can be visualized as points on the Gittins index curves based on the service they have attained. The x-axis of a point represents the number of bits served for that flow, and y-axis is its Gittins index based on its class, for example, a new flow arriving to class i is represented by the point $(0, \mathcal{G}_i(0))$. As the flows get served they move along the Gittins index curve.

Consider the characteristics of the optimal scheduling policy when all the flows in the system have received less than θ bits of service. The flow which is in state F_1 on the Gittins index curve in Fig. 2.4 has received less than θ bits of service. Its Gittins index is greater than $\mathcal{G}_1(0)$. This means it enjoys a higher priority over

new arrivals to class 1 and over the flows in class 1 which have not been served till now. Therefore, the scheduling policy is First Come First Serve (FCFS) among the class 1 flows which have received service less than θ bits. This is true for other classes too. This FCFS policy is a result of the NBUE property of the flow size distribution when flow sizes are less than β bits (which is less than θ). Due to the NBUE property, a flow which has received strictly positive service and less than θ bits is more likely to complete soon rather than a newly arriving flow. Therefore, such a flow has a higher Gittins index than any newly arriving flow to its class, and hence, is scheduled ahead of the later arrivals to its class.

In scenarios where the capacities of various classes are widely separated, if $i > j$, then $\mathcal{G}_i(x_i) > \mathcal{G}_j(x_j), \forall x_i, x_j \leq \theta$. Therefore, among the flows which have attained service less than θ bits, flows with higher transmission rates should preempt the flows with lower transmission rates. For example, the flow in state F_2 should preempt a flow at F_1. This implies that the scheduling policy is *multi-class preemptive FCFS* for all flows which have attained service less than θ bits i.e., the policy is FCFS for flows in a class and flows in classes with higher transmission rates can preempt flows with lower transmission rates.

Next we discuss the characteristics of the optimal scheduling policy for really long flows which have received a large amount of service. Consider points P_1, P_2, and P_3 on the Gittins index curves in Fig. 2.4. They all have the same value for their Gittins index. Let M be the total number of flows in these states. If we consider distributions with Pareto tails, i.e., the tail probability decays as $1/x^\alpha$, $\alpha > 1$, then it is clear that the Gittins index scheduler serves these M flows according to the

Processor Sharing (PS) discipline with equal fraction of time given to all the flows, see [27]. Since each flow receives an equal fraction of time, they get rates proportional to their channel capacities, i.e., allocation is Proportionally Fair (PF).

Another key observation is that all really long flows in the system which already received a large amount of service have a lower Gittins index than new arrivals and the flows which have received service less than θ bits. This is due to the DHZ property of the tail. From Prop. 2.3.1 and Lemma 2.3.2, $\mathcal{G}_i(x) = c_i h_X(x)$, $\forall x > \beta$. Since $h_X(x)$ is decreasing in x, $\mathcal{G}_i(x)$ eventually is lower than the Gittins index of new arrivals and that of the flows which have received service less than θ bits.

To summarize, we have the following key characteristics of the optimal scheduler

1. All flows with given cumulative service less than θ bits are served based on preemptive priority for classes with higher c_i and FCFS within classes.

2. Flows which have received a large cumulative service are eventually served using PF scheduling.

3. Flows with received service less than θ bits have priority over those which have already seen a large cumulative service.

The above characteristics motivate an approximation to the optimal Gittins index scheduler. This is explained next.

Let f be an active flow. Its time of arrival is given by $f.t$. At any point in time flows in a given class i are partitioned into two sets: $\mathcal{L}_i$ denoting those that have received less than or equal to θ bits and $\mathcal{H}_i$ the remaining flows. Define $\mathcal{L} := \cup_{i=1}^{K} \mathcal{L}_i$ and $\mathcal{H} := \cup_{i=1}^{K} \mathcal{H}_i$. The sets $\mathcal{L}$ and $\mathcal{H}$ consist of all active flows which have received less than θ bits of service and more than θ bits of service, respectively. If $\mathcal{A}$ and $\mathcal{B}$ are two sets, then $\mathcal{A} \succ \mathcal{B}$ implies that the flows of $\mathcal{A}$ are given preemptive priority over the flows of $\mathcal{B}$. Next we introduce our approximation to Gittins index scheduler which we denote by p-FCFS + PF (θ).

2.3.5 p-FCFS + PF (θ)

To specify a scheduling policy, we need to specify how flows are prioritized among the sets $\{\mathcal{L}_i\}_{i=1}^{K}$ and $\{\mathcal{H}_i\}_{i=1}^{K}$. Once we decide the priority between sets, we specify how resources are allocated to flows within these sets. We shall give priority to various sets in the following manner – $\mathcal{L}_K \succ \mathcal{L}_{K-1} \ldots \succ \mathcal{L}_2 \succ \mathcal{L}_1 \succ \mathcal{H}$. In $\mathcal{L}_i$, the flow which has the earliest arrival time has the highest priority. In $\mathcal{H}$, all flows have the same priority. At each slot we implement Algorithm 1.

This is a simple low complexity scheduling policy which approximates the optimal Gittins index scheduler for small and really large flows. It only requires knowledge of one parameter– the cross-over threshold θ. Below we show that θ is a solution to a fixed point equation. We derive an approximate expression for θ which depends on two easily measurable properties – the mean flow size and the exponent of decay of the tail probability of flow size distribution.

Proposition 2.3.3. *For NBUE + Pareto (α, β) distribution, θ is obtained by solving*

Algorithm 1 p-FCFS + PF (θ)

```
{L_i, H_i} ← FLOW MANAGEMENT( θ)
if L ≠ φ then
    i* = argmax_i {i|L_i ≠ φ}
    Serve flow f* = argmin_f {f.t|f ∈ L_{i*}}
else
    if H ≠ φ then
        Serve all flows in H according to PF scheduling policy.
    end if
end if
procedure FLOW MANAGEMENT(θ)
    Update each L_i with new arrivals.
    Move flows with attained more than θ bit of service from the corresponding L_i
to H_i.
    Remove flows that have completed service.
end procedure
```

the following fixed point equation:

$$\theta = \alpha \left[\frac{\mathbb{E}[X] - P(X > \theta) \frac{\alpha\theta}{\alpha - 1}}{P(X \leq \theta)} \right], \tag{2.11}$$

where X is the random variable denoting the flow size. For large enough values of α, $\theta \approx \alpha \mathbb{E}[X]$.

Proof. Proof is given in Appendix 2.7. □

For $\alpha > 2$, our approximation is quite close to θ. Detailed comparisons between θ and its approximation are given in Table 2.2. In the sequel we give the expressions for the mean delay as a function of the flow size for our p-FCFS + PF (θ) scheduler.

	$\beta = 3$ KBytes, $\mu = 0\,\text{KByte}^{-1}$		$\beta = 3$ KBytes, $\mu = 0.5\,\text{KByte}^{-1}$		$\beta = 3$ KBytes, $\mu = 1\,\text{KByte}^{-1}$	
α	θ	Approximation	θ	Approximation	θ	Approximation
1.5	9.8	16.9	2.9	4.9	3.0	1.9
1.9	10.2	13.5	2.9	4.3	3.0	1.9
2.3	11.8	12.2	4.5	4.8	3.0	2.4
2.7	12.8	12.9	5.1	5.2	3.0	2.8
4.0	16.0	16.0	7.1	7.1	4.0	4.0

Table 2.2: Comparison between θ and approximation for Exp. + Pareto distribution

2.3.6 Mean delay analysis for p-FCFS + PF (θ)

Before we discuss the derivations in detail, we introduce further notations.

1. Let $G(\cdot)$ be the c.d.f. of flow size. Then $G^{(x)}(\cdot)$ denotes the truncated version of $G(\cdot)$ at x, this is given by

$$G^{(x)}(y) = \begin{cases} G(y), & y < x, \\ 1, & y \geq x. \end{cases} \tag{2.12}$$

2. Expectation of flow size with respect to $G^{(x)}(\cdot)$ is denoted by $\mathbb{E}^{(x)}[X]$.

3. We denote the load arriving to class i for truncated flow sizes at x bits by $\rho_i^{(x)} = \lambda_i \frac{\mathbb{E}^{(x)}[X]}{c_i}$.

4. The overall load arriving to class i, which is denoted by ρ_i, is given by $\rho_i = \lambda_i \frac{\mathbb{E}[X]}{c_i}$. The overall load arriving to the system is denoted by $\rho = \sum_{i=1}^{K} \rho_i$.

5. Let $T(x)$ be the expected delay of a typical flow of size x. Similarly $T_i(x)$ be the expected delay of a typical flow of size x that belongs to to class i.

6. Let $W^{(\theta)}_{(\text{P-FCFS},i)}$ be the stationary workload in the system seen by a flow arriving to class i. It includes the time to serve flows of ith class or higher which are

already present in the system when class i flow arrives. It is given by

$$W^{(\theta)}_{(\text{P-FCFS},i)} = \frac{\frac{1}{2}\sum_{l=i}^{K}\lambda_l \frac{\mathbb{E}^{(\theta)}\left[X^2\right]}{c_l^2}}{1-\sum_{l=i}^{K}\rho_l^{(\theta)}}. \tag{2.13}$$

See [34] for its derivation.

Next we derive the mean delay expressions for p-FCFS + PF (θ). We give exact expression when $x \leq \theta$. For $x > \theta$, the system can be modeled as a PS system with batch arrivals. We use the analysis in [31] to obtain upper and lower bounds for delay. Finally we conclude this section with the insights obtained from the delay expressions.

Theorem 2.3.4. *If $\rho < 1$, then the mean delay for a multi-class M/GI/1 queuing system under p-FCFS + PF (θ) service policy satisfies following:*

For flows of size $x \leq \theta$:

$$T_i(x) = \frac{W^{(\theta)}_{(P\text{-}FCFS,i)}}{\left(1-\sum_{l=i+1}^{K}\rho_l^{(\theta)}\right)} + \frac{x}{c_i} + \sum_{l=i+1}^{K}\frac{x}{c_i}\left(\frac{\rho_l^{(\theta)}}{1-\sum_{j=l}^{K}\rho_j^{(\theta)}}\right), \tag{2.14}$$

For flows of size $x > \theta$:

$$T_i(x) = \frac{W^{(\theta)}_{(P\text{-}FCFS,1)} + \theta/c_i}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}} + \frac{T_{BPF(x-\theta),i}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}, \tag{2.15}$$

where $c_1(x-\theta) \leq T_{BPF(x-\theta),i} \leq c_2(x-\theta)$ with constants $c_1, c_2 > 1$.

Proof. Proof is given in Appendix 2.8. □

The expression for $T(x)$ is easily obtained from $T_i(x)$ as $T(x) = \sum_{i=1}^{K}\frac{\lambda_i}{\lambda}T_i(x)$.

Remarks:

1. Mean delay of small flows is less sensitive to the flow size as compared to PF scheduler. For the PF scheduler, the mean delay seen by a class i flow of size x bits is given by $T_i^{\mathrm{PF}}(x) = \frac{x/c_i}{1-\sum_{l=1}^{K}\rho_l}$. It can be shown that for $x \leq \theta$

$$\frac{dT_i(x)}{dx} < \frac{dT_i^{\mathrm{PF}}(x)}{dx}, \quad x \leq \theta. \tag{2.16}$$

   This is a desirable characteristic for small flows generated by interactive web traffic. For web browsing, users care about the time to display web pages, irrespective of the size of web pages.

2. Mean delay for flows of size $x > \theta$ is more sensitive to flow size as compared to the PF scheduler. For $x > \theta$, we have

$$\frac{dT_i(x)}{dx} > \frac{dT_i^{\mathrm{PF}}(x)}{dx}, \quad x > \theta. \tag{2.17}$$

   Therefore, when $x \to \infty$, $T_i(x) > T_i^{\mathrm{PF}}(x)$. However, $T_i(x)$ still increases linearly with x.

2.4 TCP Based Implementation

In this chapter we have so far assumed that if a flow is active, then the data to be transmitted is always available to the BS. However, this may not be the case when the data packets are sent over a TCP connection. Due to the congestion and flow control mechanisms of TCP, all the packets of a flow may not have reached the BS. In order to address this issue, we modify Algorithm 1.

For each active flow we maintain a queue for its packets. The flows themselves form a queuing system. Therefore, we consider a "queue of queues". This is shown

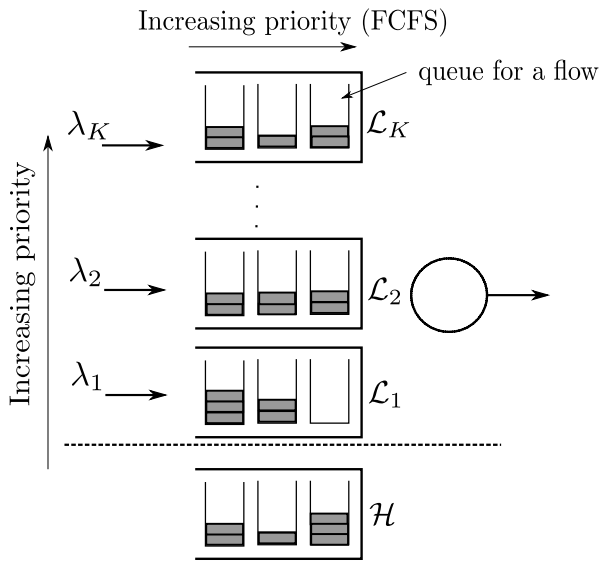


Figure 2.5: Extension of Algorithm 1 for TCP based flows.

in Fig. 2.5. We consider a straight forward extension of Algorithm 1 in which we schedule the flow with highest priority and non-empty queue for transmission at each slot. The priority across classes and between flows are same as in Algorithm 1. This is also illustrated in Fig. 2.5.

2.5 Performance Evaluation

In this section we present the results obtained from simulations. First we present the simulation results for the idealized queuing model. This is followed by a study of the performance of our context-aware scheduler under heterogeneous traffic conditions. We compare its performance with that of the PF scheduler.

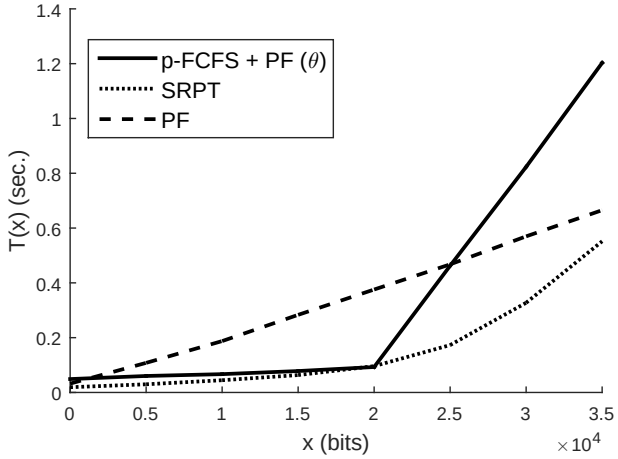


Figure 2.6: The mean flow delay $T(x)$ is plotted as a function of the flows size for p-FCFS + PF (θ), PF and SRPT schedulers

2.5.1 Idealized queuing model

In Fig. 2.6, we compare the mean delay as a function of the flow size for p-FCFS + PF (θ), SRPT, and PF schedulers. We simulate an M/GI/1 queue with five different classes of users based on their transmission rates. The transmission rates are time invariant in this setup. The flow size distribution is Exp. + Pareto with parameters $\mu = 0.125\,\text{Kbits}^{-1}$, $\beta = 21.6\,\text{Kbits}$, and $\alpha = 4$. We choose $\theta = 22\,\text{Kbits}$ for simulation results with good confidence intervals. We observe that the p-FCFS + PF (θ) approximates SRPT scheduler closely for flows less than θ Kbits. For really small flows (less than 1 Kbits), PF scheduler does slightly better than p-FCFS + PF (θ) scheduler. This is because in p-FCFS + PF (θ), a new flow, however small, has to wait for the workload ahead of it to be completed, whereas in PF scheduler, they get served immediately. For flow sizes between 1 and 25 Kbits, p-FCFS + PF (θ) has a much lower mean delay than the PF scheduler. Figure 2.6 also validates

our analysis for the mean delay. For $x \leq \theta$ Kbits, the mean delay for p-FCFS + PF (θ) is less sensitive to the variation in the value of x as compared to PF scheduler. For $x > \theta$ Kbits, the mean delay for p-FCFS + PF (θ) is much more sensitive to the variation in the value of x and as x increases, eventually, it is higher than the mean delay curve for PF.

2.5.2 Context-Aware scheduler

We consider a single BS serving 9 video streaming users and a dynamic number of active web browsing sessions and file downloads. The BS uses slotted time with slot duration $\tau_{\text{slot}} = 0.01$ sec. It makes scheduling decisions at the beginning of each slot. At any time instant, the users are located at varying distances from the base station and therefore, have heterogeneous channel strengths. The channel variations due to mobility are modeled by Markov Chain. The marginal distribution of this Markov chain is same as appropriately scaled versions of the channel strength distribution obtained from an HSDPA system. See [24] for more details on the generation of channel realizations. We classify the users into 10 different classes based on their channel strengths at each slot. Due to the time varying nature of wireless channels, the users may move from one class to another.

The flow sizes of the mix of web browsing and file downloads are modeled as a Pareto distribution with the parameters $\beta = 40$ Kbits and $\alpha = 5$. These flows arrive to the system as a Poisson process with suitable rate, independent of the video traffic in the system. We classify the flows less than 100 Kbits size as interactive web traffic and the rest as file downloads.

The stored video delivery model which we simulate mimics the DASH framework. Similar simulation model for video has also been studied in [24]. The video users view different parts of three open source movies, namely, Oceania, Route 66, and Valkama. The video segments sent are of one second playback duration. Each video segment has 6 different representations of varying quality and segment sizes. The sizes of various representations in the increasing order of quality are 100, 200, 300, 500, 900, and 1500 Kbits/segment. We use MSSSIM-Y metric (see [37]) for video segments to measure the mean quality of the video stream delivered.

The video client application with each user requests the next video segment only after the previous segment is delivered. The video client can buffer at most ten video segments. When the buffer is not full the client requests the next segment using the state-of-the-art algorithm QNOVA proposed in [24]. QNOVA is a client application which takes into account mean-variability trade-offs in quality, pricing constraints and re-buffering constraints to request appropriate representation for next video segment. In our simulation we adjust QNOVA such that it does not consider variability in quality across video segments nor pricing constraints. We also relax the re-buffering constraints in QNOVA because our context-aware scheduler takes care of the re-buffering events.

Figures 2.7 and 2.8 plot the mean quality of video streams and the average re-buffering time as a function of the normalized load of web traffic and file downloads, respectively. Normalized load is defined as the total data rate of web traffic and file downloads arriving to the system divided by the mean transmission rate for flows. It is a proxy for the fraction of system utilization by web traffic and file

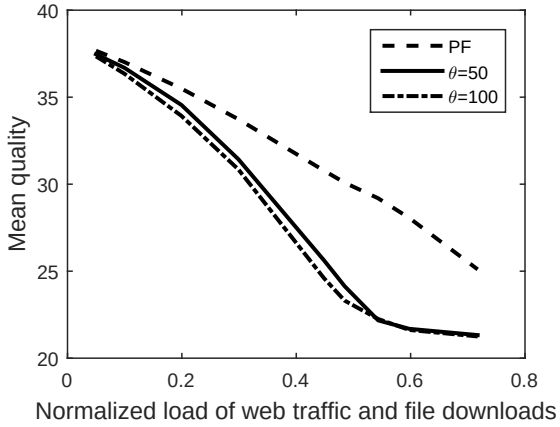


Figure 2.7: Mean quality vs normalized load of web traffic and file downloads.

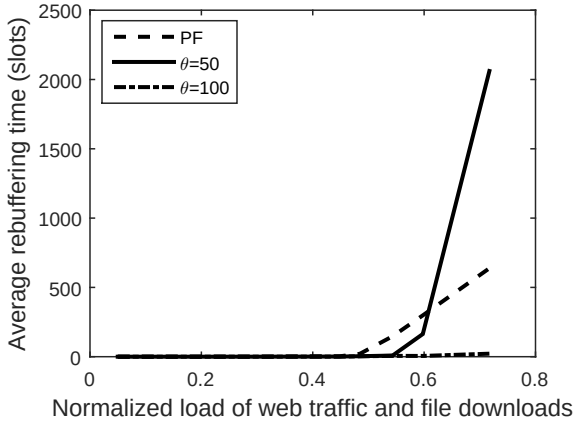


Figure 2.8: Mean re-buffering time vs normalized load of web traffic and file downloads.

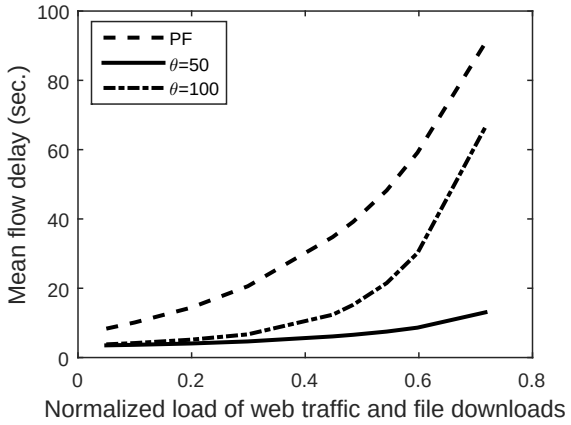


Figure 2.9: Mean delay for flows less than 100 Kbits vs normalized load of web traffic and file downloads.

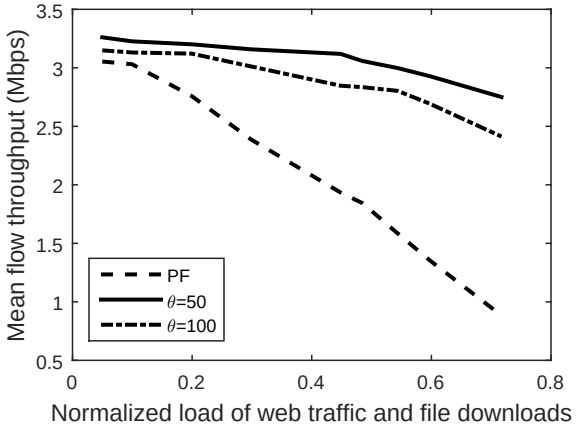


Figure 2.10: Mean throughput for flows greater than 100 Kbits vs normalized load of the interfering web traffic and file downloads.

downloads. Figures 2.9 and 2.10 plot the mean flow delay for interactive web traffic and the mean throughput for file downloads as a function of its normalized load. We compare our context-aware scheduler with the PF scheduler which does not use contextual information. Through simulations we found that θ between 50 Kbits and 100 Kbits give good results. The key results are:

1) Trade-off between mean quality and mean delay at lower loads. In Fig. 2.9, we observe that our scheduler improved the mean delay for interactive web traffic by atleast 54% for loads less 0.4, when compared to the PF scheduler. This is because it expedites flows of size less than θ via the flow and channel-aware scheduler block in our context-aware scheduler. Thus there is slight reduction in the mean video quality for system loads less 0.4. Since the lowest quality representation of video is 100 Kbits, $\theta = 50$ Kbits selectively expedites short flows over video and file downloads much more than $\theta = 100$ Kbits. Therefore, $\theta = 50$ gives better mean delay performance.

2) Robustness to loads. In Fig. 2.9, we observe that for our scheduler the mean delays for flows less than 100 Kbits size do not vary much for loads less than 0.4. For example, when $\theta = 50$, the mean flow delay increases by 85% when the load increases from 0.1 to 0.4. However, for the PF scheduler, the mean delay increases by 323.42% for the same range of loads. This robustness is a result of our scheduler favoring short flows and forcing the video clients to request lower representations as the load increases. Therefore, the video streams adapt better to the changing system load in our context-aware scheduler than under PF scheduler.

3) Trade-off between mean quality and re-buffering at higher loads.

Figure 2.8 shows that our scheduler accommodates a much higher load of interfering web traffic and file downloads without any re-buffering. For $\theta = 50$ Kbits, our context-aware scheduler can sustain video playback without re-buffering till a load of 0.55. This is 14.6% gain over PF scheduler which has non-zero re-buffering time at a load of 0.48. Similarly for $\theta = 100$ Kbits we see a gain of 45.8%. For $\theta = 100$ Kbits the gain is higher because we give priority to all flows less than 100 Kbits, which include the lowest quality video segments. The price we pay for avoiding re-buffering is the reduction in mean quality at higher loads, say between 0.4 to 0.6. There are two reasons for this reduction in mean quality. First, our scheduler favors flows of size less than θ. Second, when the system is congested, the re-buffering avoidance mechanism in the BS prevents users which have sufficient segments in their playback buffers from obtaining the radio resources.

4) Increased throughput. Figure 2.10 shows that our schedulers have a higher mean throughput for flows of size exceeding 100 Kbits. For a load of 0.4, our scheduler has atleast a gain of 45.8%. As we have seen in Fig. 2.6, our flow and context-aware scheduler significantly reduces the delay for flows of size slightly larger than θ Kbits. This results in the increased throughput for our scheduler. However, we note that for really large flows the mean throughput in our scheduler could be less than that of PF scheduler, but such events occur very rarely.

2.6 Conclusions

In this chapter, we aimed to design and study scheduler achieving robust QoS/QoE trade-offs amongst heterogeneous applications/users sharing a Base Sta-

tion. Robustness here corresponds in part to the possibility of changing the nature of the trade-offs as the network loads increase so as to better address the sensitivity of various applications/users to congestion. Through a combination of analysis and extensive simulations we have evaluated our proposed framework and believe that it has met the objectives we set for mixes of streaming video, web browsing, and file transfers which are the lions share of today's wireless data traffic.

Appendix

2.7 Proof of Proposition 2.3.3

From Prop. 2.3.1 we know that $\mathcal{G}_X(0) = h_X(\theta)$. For NBUE + Pareto (α, β) distributions, $h_X(\theta) = \alpha/\theta$. Using the definition of Gittins index and the fact that $\theta = \Delta^*(0)$, we get that

$$\mathcal{G}_X(0) = J(0, \theta) = h_X(\theta) = \frac{\alpha}{\theta}. \tag{2.18}$$

The expression for $J(0, \theta)$ in (2.4) could be re-written as

$$J(0, \theta) = \frac{P(X \leq \theta)}{\mathbb{E}[X] - P(X > \theta)\,\mathbb{E}[X|X > \theta]}. \tag{2.19}$$

For NBUE + Pareto (α, β) distributions $\mathbb{E}[X|X > \theta] = \frac{\alpha\theta}{\alpha-1}$ and $P(X > \theta) = (\beta/\theta)^\alpha$. Substituting these expressions in (2.18), we get the fixed point equation (2.11). For large values of α, $P(X \leq \theta) \approx 1$. Using this approximation in (2.11), we get $\theta \approx \alpha\mathbb{E}[X]$.

2.8 Proof of Theorem 2.3.4

We condition on the arrival of a flow of size x bits and derive the expressions for mean delay. We consider two separate cases, namely, $x < \theta$ and $x \geq \theta$.

2.8.1 Case-I, $x < \theta$

Recall that in p-FCFS + PF (θ), the new arrivals go into a multi-class priority FCFS system till they attain θ bits of service. After that they go into a low priority queue, where everyone is served according to PF discipline when multi-class priority FCFS system is empty. Therefore, flows already present in the system with service attained greater than θ bits do not interfere with the multi-class priority based FCFS part. Also, the newly arriving flows of size greater than θ bits are equivalent to flows of size θ bits for the multi-class priority FCFS system. Hence, we use the truncated c.d.f. $G^{(\theta)}(x)$ as the flow size distribution.

The mean delay for a typical flow of size x bits of class i has the following components

1. The stationary workload seen in the system due to classes $i, i+1, \ldots, K$.
2. The new arrivals to classes $i+1, i+2, \ldots, K$ while the flow is waiting for service.
3. The server time taken to serve the flow, which is equal to x/c_i.
4. The time spent in preemption due to newly arriving flows in classes $i+1, i+2, \ldots, K$ while the flow is being served.

The stationary workload ahead of a new arrival of class i is obtained from the *Pollazeck-Khinchin* formula applied to this system. It is given by

$$W^{(\theta)}_{(\text{P-FCFS},i)} = \frac{\frac{1}{2}\sum_{l=i}^{K} \lambda_i \frac{\mathbb{E}^{(\theta)}\left[X^2\right]}{c_i^2}}{1 - \sum_{l=i}^{K} \rho_l^{(\theta)}} \tag{2.20}$$

Due to the new arrivals to classes $i+1, i+2, \ldots, K$ while the flow in class i is waiting for service, the workload seen gets inflated to $W^{(\theta)}_{(\text{P-FCFS},i)} / \left(1 - \sum_{l=i+1}^{K} \rho_l^{(\theta)}\right)$.

Now we have to compute the time spent in preemption. Suppose that our class i flow of size x bits is in service and has received τ seconds of service till now. Consider the next infinitesimal time $d\tau$. During this time $d\tau$ a new arrival could occur in any of the classes $i+1, i+2, \ldots, K$. These new arrivals could preempt the flow of class i in service. An arrival to l^{th} class, $l > i$ occurs with probability $\lambda_l d\tau + o(d\tau)$. Since we consider the infinitesimal interval $d\tau$, more than one arrival occurs with negligible probability. Each time a new arrival occurs to any of the classes $i+1, i+2, \ldots, K$, it starts a new busy cycle. Duration of a busy cycle started by l^{th} class, $l > i$ is given by $\frac{\mathbb{E}^{(\theta)}[X]}{c_l\left(1-\sum_{j=l}^{K} \rho_j^{(\theta)}\right)}$. Note that the inflation by the factor $\left(1 - \sum_{j=l}^{K} \rho_j^{(\theta)}\right)$ is because the new arrivals to any of the classes $l, l+1, \ldots, K$ could extend the busy cycle started by an arrival to the lth class. Therefore, the total time spent in preemption by new arrivals to the lth class is given by

$$\text{Total time spent in preemption due to } l^{\text{th}}\text{class} \tag{2.21}$$

$$= \int_0^{x/c_i} \frac{\mathbb{E}^{(\theta)}[X]}{c_l\left(1 - \sum_{j=l}^{K} \rho_j^{(\theta)}\right)} \lambda_l d\tau, \tag{2.22}$$

$$= \frac{x}{c_i}\left(\frac{\rho_l^{(\theta)}}{1 - \sum_{j=l}^{K} \rho_j^{(\theta)}}\right). \tag{2.23}$$

Using (2.20) and (2.23), we get the expression for total expected delay of a typical arrival of size x bits to class i as

$$T_i(x) = \frac{W^{(\theta)}_{(\text{P-FCFS},i)}}{\left(1 - \sum_{l=i+1}^{K} \rho_l^{(\theta)}\right)} + \frac{x}{c_i} + \sum_{l=i+1}^{K} \frac{x}{c_i}\left(\frac{\rho_l^{(\theta)}}{1 - \sum_{j=l}^{K} \rho_j^{(\theta)}}\right), \quad x < \theta. \tag{2.24}$$

2.8.2 Case-II, $x > \theta$

A flow of size $x > \theta$ bits of class i first enters the multi-class priority based FCFS system, obtains service of θ bits, and then is served using PF discipline if the multi-class priority based FCFS system is empty. For the purpose of analysis, we split the mean delay in two components.

1. The mean time spent before it is served for the first time using PF discipline. This includes the time spent in the multi-class priority FCFS system and the waiting time before it is served according to PF discipline.

2. The mean time spent in the system after it starts service in PF queue.

Consider the first component of mean delay. A flow with size $x > \theta$ bits has to first finish its service of θ bits in multi-class priority FCFS system, then has to wait for the busy cycle of the multi-class priority FCFS system to be over before it is served using PF discipline for the first time. The stationary workload seen by an arriving flow in the priority FCFS system is given by the *Pollazeck-Khinchin* formula $\frac{\frac{1}{2}\sum_{l=1}^{K}\lambda_i \frac{\mathbb{E}^{(\theta)}\left[X^2\right]}{c_i^2}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}$. The service time taken by the flow is θ/c_i. This total workload plus the service time would be inflated by the factor $1-\sum_{l=1}^{K}\rho_l^{(\theta)}$ due to the new arrivals into system during their service. Therefore, we have

Mean time spent before being served by PF discipline

$$= \frac{\frac{\frac{1}{2}\sum_{l=1}^{K}\lambda_i \frac{\mathbb{E}^{(\theta)}\left[X^2\right]}{c_i^2}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}} + \theta/c_i}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}. \quad (2.25)$$

After the busy period of priority FCFS, our tagged flow of size x bits and all the other flows of size greater than θ bits present in the system are served using PF discipline. They are served till the next new arrival into the priority FCFS queue. Once a new arrival enters priority FCFS queue it starts a new busy cycle. This cycle repeats again with alternating periods of busy cycles of multi-class priority FCFS system and the PF queue. To analyze the delay we consider a virtual time axis with only the intervals in which the PF queue is served. At the beginning of each interval a batch of arrivals enter into the PF queue. The interval durations are i.i.d. exponentially distributed with the parameter $\sum_{i=1}^{K} \lambda_i$. Therefore, this system could be modeled as an M/GI/1 queuing system with batch arrivals. The delay expression for such a system is given as a solution of an integro-differential equation in [34]. In [31], the authors have derived upper and lower bounds for mean delay as function of the flow size. the flow has already received service of θ bits. The residual service left to be done by the PF queue is $x - \theta$ bits. Let the $T_{\text{BPF}}(x-\theta, i)$ be the mean virtual time for a flow of size $x - \theta$ bits of class i in PF queue. Using Theorem 3 in [31], we have

$$\frac{(x-\theta)/c_i}{1-\tilde{\rho}} \leq T_{\text{BPF}(x-\theta),i} \leq \min\left\{\frac{(b+1)(x-\theta)}{c_i(1-\tilde{\rho})}, \frac{(x-\theta)/c_i}{1-\tilde{\rho}} + \frac{bS(2-\tilde{\rho})}{2(1-\tilde{\rho})^2}\right\}, \quad (2.26)$$

where $\tilde{\rho} = \lambda \mathbb{E}[N]\mathbb{E}[S]$, with $\mathbb{E}[S]$ being the mean flow service time in PF queue and is given by $\mathbb{E}[S] = \sum_{i=1}^{K} \frac{\lambda_i}{\lambda} \frac{\mathbb{E}[X-\theta|X>\theta]}{c_i}$, $\mathbb{E}[N]$ being the mean batch size which is equal to $\frac{1-F_X(\theta)}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}$. The parameter b is given by the expression

$$b = 2\lambda(1-F_X(\theta))\frac{W_{(\text{P-FCFS},1)}^{(\theta)} + \frac{\theta}{\lambda}\sum_{l=1}^{K}\frac{\lambda_l}{c_l}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}. \quad (2.27)$$

During the virtual time spent in the PF queue, which is $T_{\mathrm{BPF}(x-\theta),i}$, the PF queue is preempted by several busy cycles of the multi-class priority FCFS system. These busy cycles are due to new arrivals to all the classes of the multi-class priority FCFS system. Hence the virtual time $T_{\mathrm{BPF}(x-\theta),i}$ is inflated by a factor of $1-\sum_{l=1}^{K}\rho_l^{(\theta)}$. Therefore, the mean delay of a class i flow of size x bits is given by

$$T_i\left(x\right)=\frac{\frac{\frac{1}{2}\sum_{l=1}^{K}\lambda_i\frac{\mathbb{E}^{(\theta)}\left[X^2\right]}{c_i^2}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}+\theta/c_i}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}+\frac{T_{\mathrm{BPF}(x-\theta),i}}{1-\sum_{l=1}^{K}\rho_l^{(\theta)}}. \tag{2.28}$$

Chapter 3

Minimizing Functions of Mean Delays: A Measurement Based Scheduler

3.1 Introduction

In this chapter [1] we will focus on minimizing functions of mean delays in a multi-class queuing system which models a cellular base station. Traditional wireless schedulers have been driven by rate-based criteria, e.g., utility maximization or proportionally fair allocations, which balance the average[2] rates allocated to users and/or queue-based schedulers, which monitor packet queue lengths and/or waiting times. In particular the utility of user/application i is represented via a function $u_i(\cdot)$ of the user's average rate r_i. In the simplest and stationary instance of this framework, scheduling is performed so as to solve the following optimization problem:

$$\max_{\mathbf{r}} \ \{ \sum_{i=1}^{n} u_i(r_i) \mid \mathbf{r} \in \mathcal{R}\}, \tag{3.1}$$

where n is the number of active users, $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$ and $\mathcal{R}$ is the achievable rate region. In this setting one often assumes users always have data to transmit,

[1] Publications based on this chapter: [38] A. Anand and G. de Veciana, "Measurement-based scheduler for multi-class QoE optimization in wireless networks", in Proceedings of INFOCOM, 2017.

[2] Averages may be computed in an exponentially weighted or moving window ways and thus on different time scales.

i.e., so called *full buffer* model, see e.g. [39–41]. This approach clearly does not capture the dynamic nature of transaction/flow based traffic wherein the number of active users changes over time, and wherein QoE is driven by flow-based performance metrics and only indirectly associated with mean rate and/or packet-level delays.

Specifically we shall refer to a flow as the basic data unit whose reception drives the user perceived QoE. In particular, for interactive web browsing a flow could be the content of a web page a user requested, or in the case of a file download the associated with reception of the file. In the context of modern stored video streaming, the video is partitioned into a sequence of small files (video segment) each of which might be considered a flow that should arrive in a timely manner. Several studies have shown that users perceived QoE should be modeled as a non-linear function of the *flow-level delay*, see e.g., [6, 9]. This non-linearity gives us more flexibility in scheduling users' data. For example, for web browsing, it has been shown that users do not perceive any degradation in QoE if the flow delay is less than a certain threshold [9]. So, depending on the system loads, one may not need to be aggressive in allocating resources to web browsing users, possibly to the benefit of others.

In this chapter we consider a stochastic model where flows arrive to the system, each with a service requirement in terms of the total amount of bits to be transmitted and they depart after they have been served. We shall assume that there are C classes of users corresponding to different application or service types. Flows arrive at a rate λ_c for class c and we let d_c denote the mean delay experienced by class c flows. We model the end user's QoE through a cost which is an increas-

ing convex function of mean flow delay. The lower the cost, the better the user's QoE. The cost function may depend on the application type allowing one to capture different user/application QoE sensitivities to mean flow delays. The cost function of class c will be denoted by $f_c(\cdot)$. By contrast with rate-based scheduling, we will consider the design of a scheduling policy which solves the following optimization problem:

$$\mathcal{OP}_1 : \qquad \inf_{\mathbf{d}^\pi} \ \{ \ \sum_{c=1}^{C} \lambda_c f_c(d_c) \mid \mathbf{d}^\pi \in \mathcal{D} \ \} \tag{3.2}$$

where $\mathbf{d}^\pi := (d_1^\pi, d_2^\pi, \ldots, d_C^\pi)^T$ is the mean delay vector realized by policy π and $\mathcal{D}$ is the set of achievable mean delay vectors by all finite mean delay work conserving policies. Note that a work conserving policy need not in general have finite mean delay vector[3]. In $\mathcal{OP}_1$, we scale the cost function of a class with its arrival rate. This is a natural way to represent performance in a dynamic system where one should capture not only high costs, but the number of flows that experience high costs.

In addition to addressing drawbacks associated with the conventional approaches, our model also addresses the need to capture and realize trade-offs in how resources are allocated amongst classes. Our premise is that network operators will want to make QoE trade-offs among applications and that these may be different depending on the system loads. In other words, one should consider optimizing resource allocation for systems not only for heavy loads where such trade-offs are critical, but also for moderate to light loads. As mentioned earlier the trade-offs to be realized

[3]If the service time distribution has a finite mean but infinite second moment, then an M/GI/1 queue served according to a non-preemptive work conserving discipline has an infinite mean delay

can be quite different depending on the load and mix of traffic the system is supporting. For example, when the system is congested, it might be better to give more resources to interactive applications vs large file downloads, so that delay sensitive applications are given priority. However, for lightly loaded systems, allocating more resources to interactive applications will improve their QoE only marginally, once the mean delay is less than a threshold. Therefore, spare resources can be allocated to large file downloads.

In our framework, trade-offs are captured by specifying cost functions for each application. The delay sensitive applications have 'steeper' cost functions after the tolerable delay, as compared to delay tolerant applications. In general, as the system load increases, the mean delays seen by all classes of traffic increase. However, the delay sensitive applications get higher priority because of their steeper cost functions. Therefore, for a range of system loads, a solution to $\mathcal{OP}_1$ will achieve the necessary trade-offs. Next we discuss the related work in flow-level scheduling.

3.1.1 Related Work

Flow-level scheduling has been extensively studied in the literature, see [9, 27, 35, 42–47]. Some of the works focus only on stability of the system and do not consider delay metrics, see [42, 43]. Several other works target minimization of mean flow delay [27, 35, 46, 47]. However, as mentioned earlier the users' QoE may not be a linear function of mean delays.

The works most closely related to our work are [9, 44], and [48]. In [9], the authors show that the problem of QoE optimization in wireless networks can be mod-

eled as a Linear Programming problem. However, solving the LP is computationally expensive. Therefore, they develop a heuristic which works well. This chapter does not provide any analytical performance results for the heuristic. In [44], the authors develop scheduling policies to satisfy delay based deadlines for various applications. Using simple policies, they achieve the minimum possible deadline violation probability in systems with large amounts of resources (bandwidth and time). The cost functions which we use in our approach can be used to approximate the deadlines and give us more flexibility in allocating resources. In [48], the authors consider an approach which uses cost functions based on delay, however, their work is restricted to only non-pre-emptive scheduling. To the best of our knowledge, this is the first work which considers the minimization of cost functions of the mean delay for general flow size distributions while considering both pre-emptive and non-pre-emptive policies. However, we assume the knowledge (perhaps measured) of flow size distributions which is not assumed in [44, 48]. We deem this a strength since in principle our approach can capture measurable and base station specific characteristics of the offered loads.

Several works such as [42, 46, 47] consider wireless channel models with fast fading. Such a channel model invites the use of opportunistic scheduling policies based on the instantaneous channel conditions. However, in this chapter we focus a time invariant channel model. This model is justified when the users are relatively stationary as compared to the time scale of flow dynamics and/or when there is a *channel hardening* effect. Channel hardening occurs when many diverse paths between transmitter and receiver diminishes the effect of fast fading, see [49]. In the

sequel we will however incorporate heterogeneous channel strengths as seen by users that have very different channel characteristics due to their different locations, e.g., far or close by, relative to a base station.

3.1.2 Our Contributions

In this chapter we introduce a Measurement-based Delay Optimal (MBDO) scheduler which minimizes a non-linear cost function of the mean delays experienced in a multi-class system. Starting from a fairly general multi-class M/GI/1 queuing model for a base station we make the following contributions.

1) Extension of Gittins index scheduler: We propose and show a simple extension to the results in [10]. In particular, we show that a weighted Gittins index scheduler (**w**-GittinsScheduler) will minimize a weighted linear combination of mean delays in a multi-class system. This **w**-GittinsScheduler scheduler, serves as the workhorse for our MBDO scheduler.

2) MBDO scheduling: We propose the MBDO scheduler which based on system measurements adapts to the system characteristics so as to eventually optimize system performance. In particular, at the end of each queue busy cycle, the MBDO scheduler adapts the weights for a **w**-GittinsScheduler based scheduler based on measurements to date. Such measurements allow the scheduler to learn the loads on the system, and possibly also to the flow size statistics and optimize scheduling decisions to the specific load and mix the base station is supporting. MBDO scheduler can thus track slow variations in traffic characteristics which might change on the time-scales of few hours in wireless networks, see [50]. The scheduler can in

principle also track slow variations in flow size distributions, however, in this chapter we assume the knowledge of flow size distributions.

3) Optimality results : Under mild assumptions on flow size distributions and the knowledge of the minimum of the fraction of total traffic that might arrive to a class, we show that the mean delay vector achieved by our MBDO scheduler converges to the optimal solution of $\mathcal{OP}_1$ in probability.

Overall this approach is quite novel. We are not aware of any proposed measurement-based wireless scheduler able to optimize flow-level delays/trade-offs for a multi-class system. In addition the possibility of tuning scheduling to the traffic characteristics, e.g., flow-size distributions, which may depend on usage patterns in given locations (e.g,, university vs financial district), is novel and intriguing.

3.1.3 Organization

This chapter is organized as follows. In Section 3.2, we present a simple M/GI/1 queuing model where all flows are served at unit rate. In Section 3.3, we explain about MBDO in detail and prove the asymptotic optimality of our proposed scheme. In Section 3.4, we extend our scheme for a wireless BS, where different users could have different channel rates. Performance evaluation through simulations is given in Section 3.5.

Notation: In the sequel we denote vectors by bold faced letters and random variables by capital letters. All vectors are column vectors of length C, the number of classes in the system. The components of vectors are represented by normal faced letters, for example, $\mathbf{D}$ denotes a random vector given by $(D_1, D_2, \ldots, D_C)^T$, where

T is the transpose operator. Continuous time random processes are written as a function of time, for example, $\{\mathbf{D}(t),\, t \geq 0\}$ is a continuous time vector-valued random process. Discrete time random processes are indexed as follows $\left\{\mathbf{D}^{(k)},\, k \in \mathbb{N}\right\}$. The expectation operator is denoted by $\mathbb{E}\left[\cdot\right]$ and the probability of an event A is given by $P\left(A\right)$.

3.2 System Model

Throughout this chapter we will develop our scheduler based on a basic multi-class M/GI/1 queuing model, but expect it to be robust to the underlying assumptions. Poisson arrivals are a reasonable model for flow-based transactions and even interactive, i.e, on-off type web browsing, when viewed as an aggregate of reasonably large population. The flow service requirements are generally distributed and again it is reasonable to assume independence amongst flows. We assume that the system supports C classes of flows. Flows of class c arrive as a Poisson process of rate λ_c. The flow sizes are modeled as random variables which are i.i.d. for each class and independent of the flow sizes of other classes. Flow sizes for class c have a distribution function $G_c(\cdot)$ with a mean value of m_c bits. The scheduler does not have prior knowledge of the size of individual flows, however, it does have knowledge of the size distributions, and of the cumulative service each flow has received. Initially we assume that all flows are served at unit rate by the server, thus the stability of queue is assured if $\rho := \sum_{c=1}^{C} \lambda_c m_c < 1$. This will subsequently be relaxed in Section 3.4.

As mentioned in the introduction we associate a cost function $f_c\left(.\right)$ to each

class c, which depends on mean flow-delay d_c experienced by flows in that class. We assume that f_c is strictly convex, continuous, and differentiable. Also, f_c is non-decreasing and bounded from below. Let d_c^π be the mean delay of class c under a scheduling policy π. The overall mean delay vector for policy π is denoted by $\mathbf{d}^\pi = (d_1^\pi, d_2^\pi, \ldots, d_C^\pi)^T$. Let $\mathcal{D}$ be the set of mean delay vectors that can be achieved by finite mean delay work conserving policies. We call this as the set of *feasible mean delays*.

3.3 Cost Minimization

We are interested in finding a scheduling policy π such that $\mathbf{d}^\pi$ solves the following optimization problem.

$$\mathcal{OP}_1 : \quad \inf_{\mathbf{d}} \ \{ \ f(\mathbf{d}) := \sum_{c=1}^{C} \lambda_c f_c(d_c) \mid \mathbf{d} \in \mathcal{D} \ \}. \tag{3.3}$$

Note we will show that there is indeed a policy which achieves the infimum.

To solve $\mathcal{OP}_1$, we first consider the following optimization problem:

$$\mathcal{OP}_2 : \inf_{\mathbf{d}} \ \{ \ \sum_{c=1}^{C} \lambda_c w_c d_c \mid \mathbf{d} \in \mathcal{D} \ \}, \tag{3.4}$$

where the weights $w_c, c = 1, 2, \ldots C$ are positive real numbers.

The following corollary, which is a natural consequence of Theorem 5.6 in [10] shows that a Gittins' index based scheduler optimizes $\mathcal{OP}_2$. Below we state the result and then detail the characteristics of such schedulers.

Corollary 3.3.1. *A (**w**-GittinsScheduler) achieves the optimal delays for $\mathcal{OP}_2$. In such a scheduler the Gittins index of a flow is simply scaled by its class weight,*

and at each time instant the flow with the highest weighted index is scheduled for transmission.

Proof of this result is given in the Appendix 3.7. Next introduce **w**-GittinsScheduler and Gittins indices in detail.

<u>**w**-GittinsScheduler:</u> Let $\mathcal{A}(t)$ be the set of active flows at time t. For each flow $l \in \mathcal{A}(t)$, we associate a positive real number known as the Gittins index, which is a function of the cumulative service the flow has received, in bits. For a flow l, let its Gittins index be denoted by $\mathcal{G}_l(\cdot)$. At each time t, we scale the Gittins index of a flow by its class weight w_c. We shall refer to this as the *weighted Gittins index*. We schedule the flow with the highest weighted Gittins index at all times. If there are two or more flows with the highest weighted Gittins index, we choose one of the flows at random. Note that the **w**-GittinsScheduler with a given weight vector **w** is same as the **w**-GittinsScheduler with weight vector $\kappa\mathbf{w}$, where $\kappa > 0$. Only the relative weights across classes matter in **w**-GittinsScheduler. Therefore, in this chapter we will assume that the weights are normalized to one for **w**-GittinsScheduler. Next we review the Gittins indices for such dynamic systems given in [10, 28].

<u>*Gittins index:*</u> Consider a flow which has received a bits of cumulative service. Let $G(\cdot)$ and $\overline{G}(\cdot)$ be the cumulative density function (c.d.f.) and complementary

c.d.f. of the flow, respectively. For $\Delta \geq 0$, we define the following

$$
\begin{aligned}
R(a,\Delta) &:= \left(\overline{G}\left(a\right) - \overline{G}\left(a+\Delta\right)\right)/\overline{G}\left(a\right), \\
C(a,\Delta) &:= \left(\int_0^{\Delta} \overline{G}\left(a+t\right) dt\right)/\overline{G}\left(a\right), \\
J\left(a,\Delta\right) &:= \frac{R(a,\Delta)}{C(a,\Delta)}.
\end{aligned}
$$

Here $R(a,\Delta)$ and $C(a,\Delta)$ correspond to probability that a flow which has received a bits of service will complete, and the expected time the flow would be busy if it were allocated Δ seconds of service. Therefore, $J\left(a,\Delta\right)$ is the ratio of expected *reward* to the expected *cost* of allocating Δ seconds to a flow which has received a bits of service. The Gittins index for an active flow in our queuing model as defined in [10] is given by

$$\mathcal{G}\left(a\right) = \sup_{\Delta \geq 0} J\left(a,\Delta\right) \tag{3.5}$$

i.e., the best reward/cost trade-off over all time horizons Δ Computing the Gittins index requires knowledge of flow size distribution. In our setting, different classes of traffic may have different flow size distributions depending on the applications types in the network, and how they are grouped together into classes. Such information can in principle be easily collected by monitoring traffic on the network.

Next we discuss an approach to optimize $\mathcal{OP}_1$ based on a **w**-GittinsScheduler. The following two lemmas show that $\mathcal{OP}_1$ is a convex problem with a unique minimum which can be realized via a **w**-GittinsScheduler with appropriate weights.

Lemma 3.3.2. *If $\rho < 1$, then the achievable delay region for work conserving finite mean delay policies $\mathcal{D}$ is a non-empty convex set.*

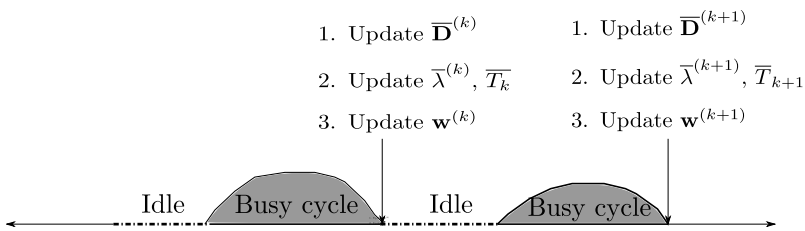


Figure 3.1: Sample path of MBDO scheduler

Lemma 3.3.3. *There exists an unique minimizer* $\mathbf{d}^*$ *for the optimization problem* $\mathcal{OP}_1$ *and it can be achieved by a weighted Gittins index policy with suitable weights.*

Proof. Proofs are given in Appendices 3.8 and 3.9, respectively. □

In the next sub-section, we will describe our policy in detail.

3.3.1 Measurement-Based Delay Optimal (MBDO) Scheduler

The idea underlying MBDO scheduling is to learn an optimal weights setting for **w**-GittinsScheduler such that optimal delays for $\mathcal{OP}_2$ are also optimizing for $\mathcal{OP}_1$.

We shall decompose the system evolution based on its renewal periods, where each period consists of an idle period and a busy cycle. The weights for the **w**-GittinsScheduler are fixed for each renewal period but adapted at the end of each renewal cycle based on measurements seen to date. This is exhibited in Figure 3.1.

Pseudo-code for our MBDO scheduler is given in the Algorithm 2 panel. The variables used and their meanings are summarized in Table **??**. The procedure W-GittinsScheduler simply implements a weighted Gittins index policy during a

Algorithm 2 Measurement-Based Delay Optimal Scheduling (MBDO)

Initialize: $\overline{\mathbf{D}}^{(0)}, \overline{\boldsymbol{\lambda}}^{(0)}, \overline{T}^{(0)}, \mathbf{w}^{(1)}$ with some non-zero positive values.
Track the following:

- The amount of service given to each flow
- The number of active flows of class c at time t, say $N_c(t)$.

for each renewal cycle k **do**
Run w-GittinsScheduler($\mathbf{w}^{(k)}$).
$\mathbf{D}^{(k)} \leftarrow$ DelayEstimate($\overline{\boldsymbol{\lambda}}^{(k-1)}, \overline{T}^{(k-1)}, \{N_c(\cdot)\}$) **Updates**:

$$\overline{\mathbf{D}}^{(k)} \leftarrow \overline{\mathbf{D}}^{(k-1)} + \epsilon_k \left(\mathbf{D}^{(k)} - \overline{\mathbf{D}}^{(k-1)}\right)$$

$$\mathbf{w}^{(k+1)} = \gamma\left(\overline{\mathbf{D}}^{(k)}\right)\left(\frac{\partial f_1}{\partial d_1}\Big|_{\overline{D}_1^{(k)}}, \frac{\partial f_2}{\partial d_2}\Big|_{\overline{D}_2^{(k)}}, \ldots \frac{\partial f_C}{\partial d_C}\Big|_{\overline{D}_C^{(k)}}\right)^T$$

$$\overline{\boldsymbol{\lambda}}^{(k)} = \frac{(k-1)\,\overline{T}^{(k-1)}}{(k-1)\,\overline{T}^{(k-1)} + T_k}\overline{\boldsymbol{\lambda}}^{(k-1)} + \frac{1}{(k-1)\,\overline{T}^{(k-1)} + T_k}\mathbf{N}^{(k)},$$

$$\overline{T}^{(k)} = \frac{k-1}{k}\overline{T}^{(k-1)} + \frac{1}{k}T_k,$$

end for

---

procedure DelayEstimate($\overline{\boldsymbol{\lambda}}^{(k-1)}, \overline{T}^{(k-1)}, \{N_c(\cdot)\}$)
return

$$D_c^{(k)} = z\left(\frac{1}{\overline{\lambda}_c^{(k-1)}\overline{T}^{(k-1)}}\right)\int_{k^{\text{th}}\text{ renewal cycle}} N_c\,(t)\,dt \quad c = 1, 2, \ldots, C$$

end procedure

---

procedure w-GittinsScheduler($\overline{\mathbf{w}}$)
  for Each time slot **do**
    for Each flow in the system **do**
      Compute Gittins index for each flow.
      Scale the Gittins index by its class weight w_c.
    end for
    Schedule the flow with highest w_c.
  end for
end procedure

Table 3.1: Variables used in MBDO

Name	Description
$\overline{\mathbf{D}}^{(k)}$	Estimate of mean delay upto and including k^{th} renewal cycle.
$\mathbf{D}^{(k)}$	Estimate of mean delay for the policy used in k^{th} renewal cycle.
$\overline{\lambda_c}^{(k)}$	Estimate of mean arrival into class c rate upto and including k^{th} renewal cycle.
$\overline{T}^{(k)}$	Estimate of mean renewal cycle duration upto and including k^{th} renewal cycle.
$N_c(t)$	Number of active flows of class c at time t.
$N_c^{(k)}$	Total number of flows that arrived to class c in k^{th} renewal cycle.
$\mathbf{w}^{(k)}$	Weights used by $\mathbf{w}$-GittinsScheduler in k^{th} renewal cycle.

busy cycle. For simplicity we further divide the time into slots and assume scheduling decisions are made at the beginning of every slot. The slot duration is assumed to be very small as compared to the flow transmission times. The computations performed at the end of a renewal cycle are discussed below.

(1) Delay measurement. We shall estimate the mean delay seen by each class in the kth renewal cycle. it is denoted by $\mathbf{D}^{(k)}$. Further we let $\overline{\mathbf{D}}^{(k)} := \left(\overline{D}_1^{(k)}, \overline{D}_2^{(k)}, \ldots, \overline{D}_C^{(k)}\right)^T$ denote the time-averaged delay vector averaged across renewal cycles up to and including the kth one. Specifically $\overline{\mathbf{D}}^{(k)}$ at the end of kth renewal cycle is updated using the new estimate $\mathbf{D}^{(k)}$ as follows:

$$\overline{\mathbf{D}}^{(k)} \leftarrow \overline{\mathbf{D}}^{(k-1)} + \epsilon_k \left(\mathbf{D}^{(k)} - \overline{\mathbf{D}}^{(k-1)}\right), \tag{3.6}$$

where $(\epsilon_k \,|\, k \in \mathbb{N})$ is a non-increasing sequence of positive real numbers such that

$$\sum_k \epsilon_k = \infty \quad \text{and} \quad \sum_k \epsilon_k^2 < \infty. \tag{3.7}$$

For technical reasons, the delay estimate $\mathbf{D}^{(k)}$ is obtained using the procedure DelayEstimate, where the function $z(\cdot)$ is defined as follows:

$$z(x) := \min\left(x, \lambda/\lambda_{c^*}\right), \tag{3.8}$$

where $\lambda_{c^*} = \min\{\lambda_i | i = 1, 2, \ldots, C\}$. The reasoning behind the choice of this estimator is discussed in § 3.3.2 and we assume the knowledge of the minimum fraction of traffic (λ_{c^*}/λ) that may arrive to any class.

(2) Updating $\overline{\boldsymbol{\lambda}}^{(k)}$ and $\overline{T}^{(k)}$. The estimate of mean arrival rate vector up to and including k^{th} cycle is given by $\overline{\boldsymbol{\lambda}}^{(k)} := \left(\overline{\lambda_1}^{(k)}, \overline{\lambda_2}^{(k)}, \ldots, \overline{\lambda_C}^{(k)}\right)$. Similarly, the estimator for the mean renewal cycle duration up to and including the kth cycle is given $\overline{T}^{(k)}$. They are updated as follows:

$$\overline{\boldsymbol{\lambda}}^{(k)} = \frac{(k-1)\,\overline{T}^{(k-1)}}{(k-1)\,\overline{T}^{(k-1)} + T_k}\overline{\boldsymbol{\lambda}}^{(k-1)} + \frac{1}{(k-1)\,\overline{T}^{(k-1)} + T_k}\mathbf{N}^{(k)}, \tag{3.9}$$

$$\overline{T}^{(k)} = \frac{k-1}{k}\overline{T}^{(k-1)} + \frac{1}{k}T_k, \tag{3.10}$$

where T_k is the random variable denoting the length of the kth renewal cycle and $\mathbf{N}^{(k)} := \left(N_1^{(k)}, N_2^{(k)}, \ldots, N_C^{(k)}\right)$ is the random vector denoting the total number of flow arrivals during that cycle for each class.

(3) Adaptation of weights: For the next $(k+1)^{\text{th}}$ renewal cycle, run **w**-GittinsScheduler with weights given by

$$w_c^{(k+1)} = \gamma\left(\overline{\mathbf{D}}^{(k)}\right)\frac{\partial f_c}{\partial d_c}\Big|_{\overline{D}_c^{(k)}}, \text{ for } c = 1, 2, \ldots, C,$$

where $\gamma\left(\overline{\mathbf{D}}^{(k)}\right)$ is the normalizing factor so that $\mathbf{w}^{(k+1)}$ has unit norm.

The procedure DelayEstimate is crucial to the optimality of MBDO, so we shall discuss it in detail next.

3.3.2 Delay Estimates

In our model the duration of renewal periods T_k's are i.i.d. since arrivals are Poisson and service times are i.i.d. Furthermore, the distribution of T_k's is independent of scheduling policy because we are considering only work conserving policies. Note that the delay estimator for class c in the kth renewal period used in the procedure DelayEstimate can be re-written as

$$D_c^{(k)} = U_c^{(k)} + B_c^{(k)} \tag{3.11}$$

where

$$U_c^{(k)} := \frac{\int_{k^{\text{th}}\text{renewal cycle}} N_c\left(t\right) dt}{\lambda_c \mathbb{E}\left[T_k\right]}, \tag{3.12}$$

$$B_c^{(k)} := \left[z\left(\frac{1}{\overline{\lambda_c}^{(k-1)}\overline{T}^{(k-1)}}\right) - \frac{1}{\lambda_c \mathbb{E}\left[T_k\right]}\right] \tag{3.13}$$

$$\times \int_{k^{\text{th}}\text{renewal cycle}} N_c\left(t\right) dt. \tag{3.14}$$

In Lemma 3.3.4, we prove that $U_c^{(k)}$ is an unbiased estimator for mean delay in k^{th} renewal cycle. We also require that this term have finite second moment to prove the convergence results of MBDO. This is proved in Lemma 3.3.5.

Lemma 3.3.4. *Let* $U_c^{(k)} = \frac{\int_{k^{th}renewal\ cycle} N_c(t)dt}{\lambda_c \mathbb{E}[T_k]}$, $c = 1, 2, \ldots, C$, *then* $U_c^{(k)}$ *is an unbiased estimator for the mean delay seen by a typical flow in* c^{th} *class for the scheduling policy in the* k^{th} *renewal cycle.*

Proof. The proof is given in Appendix 3.10. □

Lemma 3.3.5. *Let the fourth moment of flow size distribution for class c be denoted by h_c. If $h_c < \infty$, $c = 1, 2, \ldots, C$, then $\mathbb{E}\left[\left(U_c^{(k)}\right)^2\right] < \infty$, $c = 1, 2, \ldots, C$.*

Proof. The proof is given in Appendix 3.11. □

The term $B_c^{(k)}$ in (3.11) represents the bias in the estimator. The function $z(\cdot)$ truncates the value of $1/(\overline{\lambda}^{(k-1)}\overline{T}^{(k-1)})$ to λ/λ_{c^*} which ensures that the term $B_c^{(k)}$ has a finite first moment. We have chosen the value λ/λ_{c^*} for truncation to obtain asymptotic unbiased estimates as indicated in the following lemma.

Lemma 3.3.6. *If $h_c < \infty$ and $z(\cdot)$ is as defined in* (3.8)*, then* $\lim_{k\to\infty} \mathbb{E}\left[|B_c^{(k)}|\right] \to 0$.

The above lemma shows that $B_c^{(k)}$ converges to zero in expectation. The main idea is to show the almost sure convergence of the sequence $\left(|B_c^{(k)}| | k \in \mathbb{N}\right)$ to 0 as well as its uniformly integrability. Proof is given in Appendix 3.3.6. This result is necessary for the convergence result given in the next section.

3.3.3 Optimality Results

In this sub-section, we will show the asymptotic optimality of MBDO scheduling and the main result of this chapter.

Theorem 3.3.7. *Assuming flow size distributions for all classes have finite fourth moments and $(\epsilon_k \,|\, k \in \mathbb{N})$ satisfies* (3.7)*, then the MBDO scheduler is such that $\overline{\mathbf{D}}^{(k)}$ converges to $\mathbf{d}^*$ in probability, i.e., for any $\epsilon > 0$*

$$\lim_{k\to\infty} P\left(|\overline{\mathbf{D}}^{(k)} - \mathbf{d}^*| > \epsilon\right) = 0. \tag{3.15}$$

where $\mathbf{d}^*$ *is the unique minimizer of* $\mathcal{OP}_1$.

Let us outline the key steps of the proof of this theorem, an leave the details to appendix. Consider the piece-wise constant random process $\overline{\mathbf{D}}(t)$ which is defined as:

$$\overline{\mathbf{D}}(t) = \begin{cases} \overline{\mathbf{D}}^{(k)} & \text{if} \quad t \in \left[\sum_{i=1}^{k-1} \epsilon_i, \sum_{i=1}^{k} \epsilon_i\right) \\ 0 & \text{if} \quad t < 0. \end{cases} \tag{3.16}$$

The key ideas come from stochastic approximation algorithms wherein as ϵ_k become small, i.e., for large t, the trajectories of the sequence $\overline{\mathbf{D}}(t)$ can be approximated by the trajectories of an associated differential equation for a variable $\overline{\mathbf{x}}(t)$ given by

$$\frac{d\overline{\mathbf{x}}(t)}{dt} = \mathbf{g}^*\left(\overline{\mathbf{x}}(t)\right) - \overline{\mathbf{x}}(t), \tag{3.17}$$

where $\mathbf{g}^*\left(\overline{\mathbf{x}}\right) := \underset{\mathbf{d}\in\mathcal{D}}{\operatorname{argmin}} \nabla f\left(\overline{\mathbf{x}}\right)^T \mathbf{d}$ and

$$\nabla f\left(\mathbf{x}\right) := \left(\lambda_1 \frac{\partial f_1}{\partial d_1}\Big|_{x_1}, \lambda_2 \frac{\partial f_2}{\partial d_2}\Big|_{x_2}, \ldots, \lambda_C \frac{\partial f_C}{\partial d_C}\Big|_{x_C}\right)^T.$$

This can be seen noting that the update equation (3.6) can be re-arranged as

$$\frac{\overline{\mathbf{D}}^{(k)} - \overline{\mathbf{D}}^{(k-1)}}{\epsilon_k} = \mathbf{D}^{(k)} - \overline{\mathbf{D}}^{(k-1)}. \tag{3.18}$$

In the above equation, the L.H.S. approximates $\frac{d\overline{\mathbf{D}}(t)}{dt}$ when ϵ_k is small. The term $\mathbf{D}^{(k)}$ should be viewed as an asymptotically unbiased estimate of $\mathbf{g}^*\left(\overline{\mathbf{D}}^{(k-1)}\right)$. Indeed this follows observing that:

1. It follows from Lemmas 3.3.4 and 3.3.6, we have shown that $\mathbf{D}^{(k)}$ is an asymptotically unbiased estimate of mean delay under the policy used in k^{th} renewal cycle.

2. Our choice of weights for k^{th} renewal cycle is such that it minimizes $\underset{\mathbf{d}\in\mathcal{D}}{\text{argmin:}}\ \nabla f\left(\overline{\mathbf{D}}^{(k-1)}\right)^T \mathbf{d}$ in k^{th} renewal cycle.

Therefore, for small ϵ_k we can approximate (3.18) by (6.28).

One can then show that the differential equation (6.28) is globally asymptotically stable and its trajectories converge to the optimal point of $\mathcal{OP}_1$. This follows from the following lemma.

Lemma 3.3.8. *The differential equation given by* (6.28) *is globally asymptotically stable and its asymptotically stable point is* $\mathbf{d}^*$.

Proof. Proof is given in the Appendix 3.13. □

Finally one can use Theorem 2.1 in Chapter 7, [51] to conclude the convergence of $\overline{\mathbf{D}}^{(k)}$ to $\mathbf{d}^*$. The conditions necessary for the theorem are satisfied as a result of Lemmas 3.3.4, 3.3.5, 3.3.6, and 3.3.8

3.4 Modifications to model wireless networks

In the previous section, we considered a multi-class M/GI/1 queue and assumed all flows were served at unit rate. In a wireless network flows destined to different users may experience different service rates due to the heterogeneity in channel conditions they see. For simplicity in this chapter we assume (mean) service rates may be heterogenous but are fixed for the duration of a flow. Further modifications can be considered to address opportunism and/or user mobility.

For systems with heterogeneous service rates, one can show that the Gittins index for a flow f need only be scaled by its mean service rate r_f, see [35]. If a flow f has received a cumulative service of x bits and the service rate for the flow is r_f, then its Gittins index $\mathcal{G}_f(\cdot)$ is given by:

$$\mathcal{G}_f(x) = r_f \mathcal{G}(x). \tag{3.19}$$

where $\mathcal{G}(x)$ is the Gittins index of the flow if it is served at unit rate in a queue. In summary then, the effective weight for a flow would be the product of its class weight w_c and its service rate.

3.5 Performance Evaluation

In this section, we study the performance of MBDO scheduling through discrete event simulation.

Simulation setup: We consider an M/GI/1 queue with three idealized traffic classes, so as to best understand how MDBO scheduling is performing. We will assume the total service rate is normalized to one bit/second so flow sizes are given in terms of required service time (in seconds). The service rate and the service requirements can be scaled appropriately to study other scenarios too. The three service classes are described in detail below.

1. *Small flows:* Flow sizes for this class are uniformly distributed between 0.1 and 0.3 seconds. The cost function for this class is given by $f_1(d_1) = \frac{1}{2}d_1^2$. This might model web traffic or other interactive applications which are delay sensitive. However, for mean delays upto 1 second, the cost is low.

2. *Medium sized flows:* Flow sizes are uniformly distributed between 0.3 and 0.5 seconds. The cost function used is given by $f_2(d_2) = \frac{1}{3}\left(\frac{d_2}{0.6}\right)^3$, i.e., the delay cost increases steeply after 0.6 seconds. This class represents medium sized flows with tight delay constraints. This could model the segments in a HTTP adaptive video streaming service.

3. *Large files:* Flow sizes have a Pareto distribution with c.c.d.f $\overline{G}(x) = \left(\frac{4}{x+4}\right)^5$, $x \geq 0$. They have a mean service time of 1 second. The Pareto distribution is a heavy-tailed thus this class includes a mix of small and large flows. This could be used to model a variety of file downloads. This class has a cost function $f_3(d_3) = 0.1d_3$. This class is the least sensitive to delay, i.e., most elastic or delay-adaptive.

There are closed form expression for the Gittins index of Uniform and Pareto distributions. For Uniform distribution, the Gittins index is the inverse of the mean residual service time, see [28]. If the flow size is uniformly distributed in the interval $[p, q]$, we have

$$\mathcal{G}_U(a) = \begin{cases} \frac{2}{p+q-2a} & \text{if} \quad 0 \leq a \leq p, \\ \frac{2}{q-a} & \text{if} \quad p < a < q. \end{cases} \tag{3.20}$$

For Pareto distribution, the Gittins index is equal to its hazard rate, where hazard rate is the ratio of p.d.f. to c.c.d.f., see [28]. Therefore, for our setting, the Gittins index is given by

$$\mathcal{G}_P(a) = \frac{5}{a+4}. \tag{3.21}$$

In order to see how MBDO realizes trade-offs, we shall fix the arrival rates of two classes and sweep increase that of the third class. Therefore, we study three

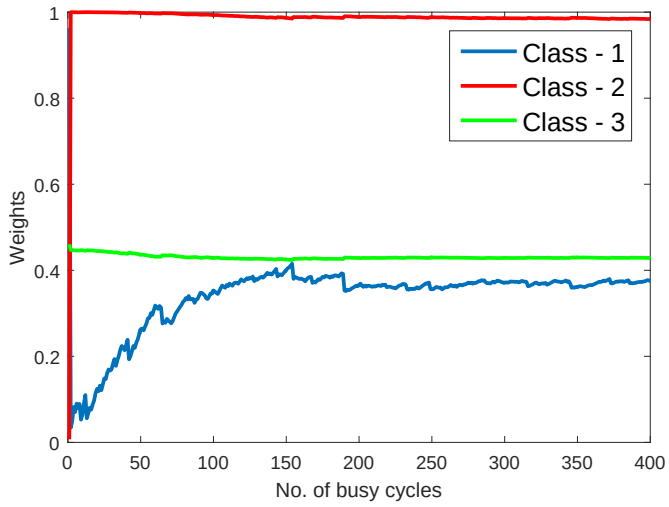


Figure 3.2: Weights for all classes as a function of the number of busy cycles for $\lambda_1 = 0.5$, when $\lambda_2 = 1$ and $\lambda_3 = 0.2$ flows/sec.

different cases based on the class for which we sweep the arrival rate. All simulations statistics were obtained based on 4×10^5 flows have been served to completion, giving trends with negligible confidence intervals. In Fig. 3.2, we have shown the convergence of weights in MBDO for. After about 100 busy cycles, the weights converge. We have used the sequence $\epsilon_k = 1/k, k \geq 1$ to average the delay in MBDO.

We shall mainly compare our scheduler with a mean delay Gittins index scheduler, i.e., **w**-GittinsScheduler with equal weights. The **w**-GittinsScheduler with equal weights is known as Gittins index scheduler in the literature and we will use the same terminology in the sequel. We have also compared our scheduler with the Processor Sharing (PS) scheduler. Note that PS is similar to Proportional Fair when the channels do not change much. However the delay performance for PS (a rate

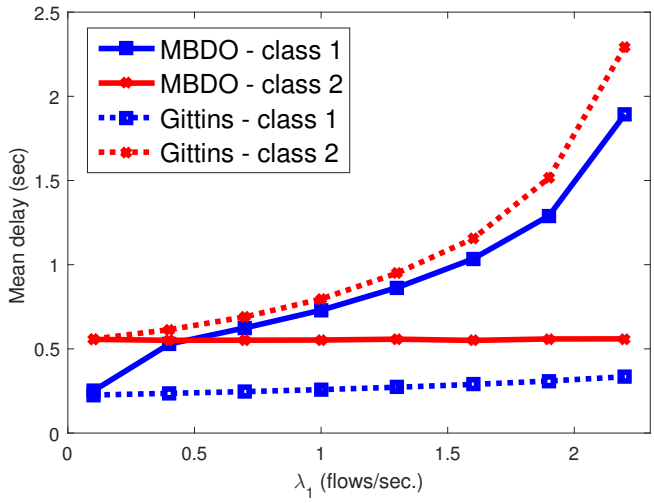


Figure 3.3: Mean delays for classes 1 and 2 as a function of λ_1, when $\lambda_2 = 1$ and $\lambda_3 = 0.1$.

based scheduler) is much worse than MBDO, since it is not geared towards minimizing delays of flow and hence, we cannot illustrate the trade-offs in resource allocation achieved by MBDO when we compare it to PF. The comparison with PS for sweeping the arrival rates of small flows is given in Fig. 3.4.

1) Sweep arrival rate of small flows: In this scenario, we fix the arrival rates of Classes 2 and 3 (λ_2 and λ_3) at 1 and 0.1 flows/second, respectively. We sweep the arrival rate of Class 1 (λ_1) from 0.1 to 2.2 flows/second. We have plotted the mean delays of Classes 1 and 2 vs λ_1 in Fig. 3.3. We have not shown the delay performance for the third class in this Fig.3.3 because the delay of third class is much larger in both the cases and finer details will be missed. The plot with all three classes is shown in 3.5. The two key observations obtained from Fig. 3.3 are as follows:

1. The mean delay for Class 1 flows in MBDO increases much more with λ_1 than

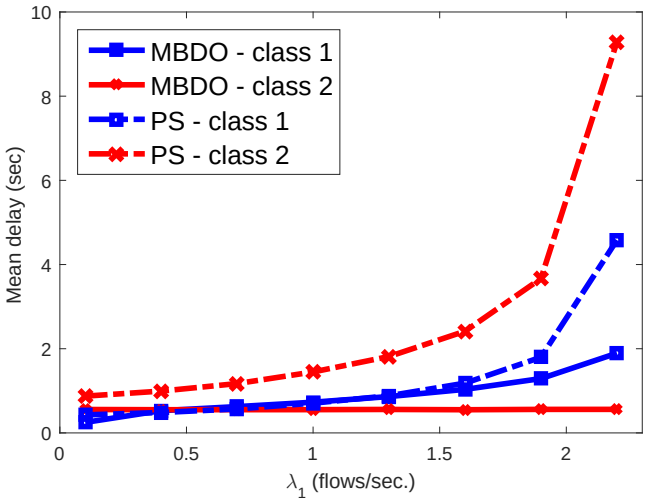


Figure 3.4: Mean delays for classes 1 and 2 as a function of λ_1, when $\lambda_2 = 1$ and $\lambda_3 = 0.1$.

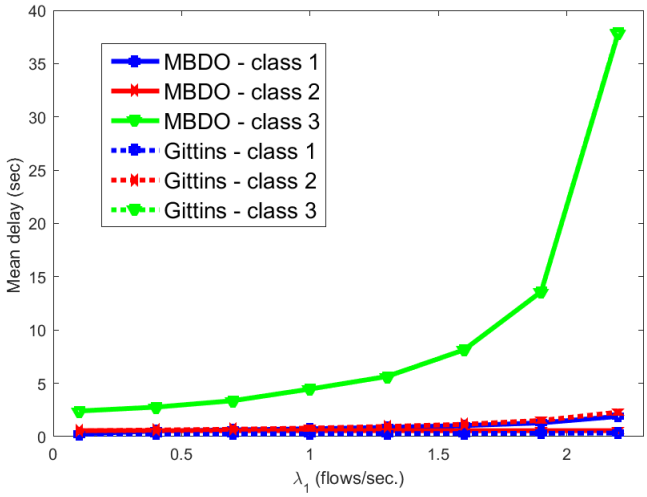


Figure 3.5: Mean delays for all classes as a function of λ_1, when $\lambda_2 = 1$ and $\lambda_3 = 0.1$.

for the Gittins index scheduler.

2. The mean delay for Class 2 flows in MBDO stays close to 0.55 sec. even with increasing λ_1, whereas for the Gittins index scheduler it increases by a factor of four on increasing λ_1 from 0.1 to 2.2 flows/sec.

Note that the Gittins index scheduler minimizes the overall mean delay of a typical arrival, i.e., solves $\mathcal{OP}_2$ with equal weights. In other words, by Little's law, it minimizes the mean number of flows in the system. Thus the Gittins index scheduler gives priority to the shorter Class 1 flows at the expense of Class 2 and Class 3 flows. However, for the MBDO scheduler, Class 2 traffic has a very steep cost function after the mean delay of 0.6. As more Class 1 flows arrive into the system, the steep cost function of Class 2 will ensure that the class 2 traffic will get more priority over the Class 1 traffic. Note that the Class 1 traffic can tolerate a mean delay up to 1 sec. without paying too much penalty. Hence, the mean delay of Class 2 does not vary much with λ_1 under MBDO scheduling. Class 3 has lower priority than both Class 1 and 2 as it has the least sensitivity to delay. Therefore, the MBDO is able to protect the most delay sensitive Class 2 traffic from both Class 1 and Class 3 traffic.

2) Sweep arrival rate of medium-sized flows: In this scenario, we keep the arrival rates of Classes 1 and 3 fixed at 1 and 0.1 flows/sec., respectively and the arrival rate of Class 2 is swept from 0.1 to 1.6 flows/sec. We show the mean delays for Classes 1 and 2 vs λ_2 in Fig. 3.6. An interesting observation is that the mean delay for Class 1 first decreases and then increases on increasing λ_2 in MBDO scheduler. Recall the objective function of $\mathcal{OP}_1$. The overall cost function $f(\cdot)$ increases with λ_2 due to

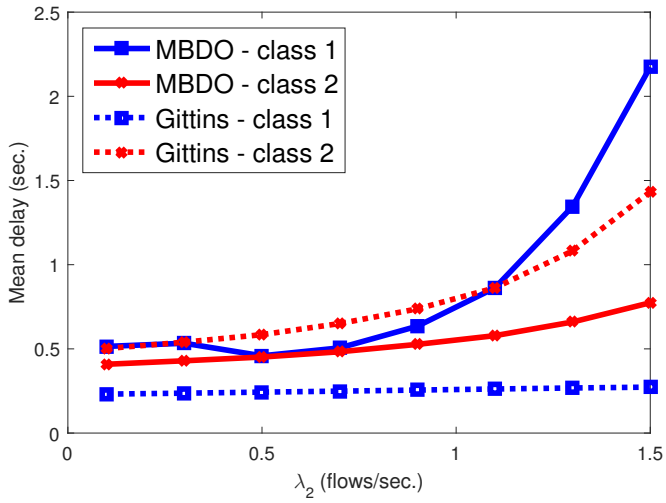


Figure 3.6: Mean delays for classes 1 and 2 as a function of λ_2, when $\lambda_1 = 1$ and $\lambda_3 = 0.1$.

increase in $\lambda_2 f_2(\cdot)$. If the mean delay for Class 2 is less than 0.6 seconds, then value of f_2 does not change much. The only way MBDO can compensate for increasing λ_2 is to decrease the cost of a traffic class, i.e., decrease the mean delay for Class 1. Note that decreasing the mean delay of Class 3 does not help much as it is not so sensitive to delay. Once the mean delay for Class 2 is close to 0.6 seconds, then it dominates the total cost and MBDO stabilizes its delay at the expense of Class 1 traffic. This is in sharp contrast to the Gittins index scheduler which always gives lower mean delays to small flows. Therefore, the mean delay for highly delay sensitive Class 2 traffic is made robust to the changes in its arrival rate in MBDO scheduler.

3) Sweep arrival rate of large flows: Here λ_1 and λ_2 are fixed at 0.5 and 1 flows/sec., respectively, while λ_3 is swept from 0.01 to 0.45 flows/sec. We exhibit the mean delay for classes 1 and 2 vs λ_3 in Fig. 3.7 and the mean delay for class 3 vs λ_3

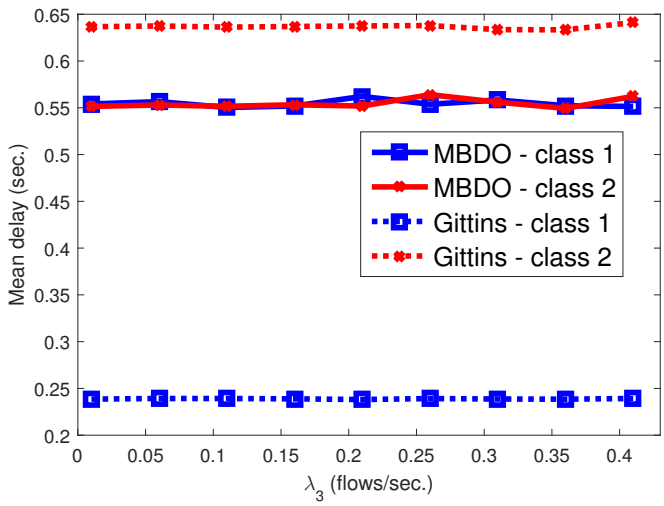


Figure 3.7: Mean delays for classes 1 and 32 as a function of λ_3, when $\lambda_1 = 0.5$ and $\lambda_2 = 1$.

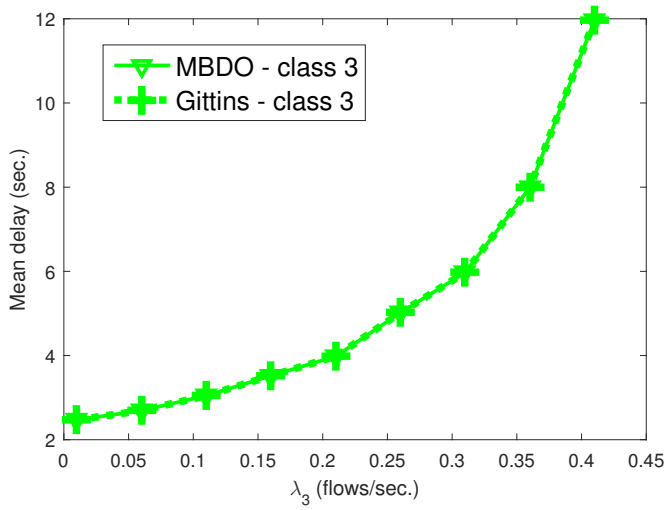


Figure 3.8: Mean delays for class 3 as a function of λ_3, when $\lambda_1 = 0.5$ and $\lambda_2 = 1$.

in Fig. 3.8. Note that the mean delays of classes 1 and 2 are not affected by increase in λ_3. There are two reasons for this.

1. For the Pareto distribution, the Gittins index decreases with the cumulative service given to the flow, whereas, for Uniform distribution it increases. For the parameters which we have used for the Pareto and Uniform distributions, $\mathcal{G}_U(0) > \mathcal{G}_P(0)$. Therefore, Classes 1 and 2 always have higher Gittins indices than class 3. This ensures that Classes 1 and 2 get absolute pre-emptive priority over class 3. Hence, the mean delays of Classes 1 and 2 are not affected by class 3 in Gittins index scheduler.

2. In MBDO, in addition to the Gittins index, we also have the weights associated with the classes. Due to the fact that class 3 is least sensitive to delay, the weights used for classes 1 and 2 are higher than class 3. This along with the characteristics of Gittins indices ensure that classes 1 and 2 get absolute pre-emptive priority over class 3 and hence, they are not affected by class 3.

Due the above mentioned effects, the class 3 always has the least priority in Gittins and MBDO schedulers and therefore, has the same mean delay in both these schedulers. Even though mean delays for classes 1 and 2 are unaffected by class 3 under the Gittins index and MBDO schedulers, they differ in their treatment of Classes 1 and 2. This is because of the effect of cost functions in MBDO which tolerates higher delays for class 1 traffic. Therefore, delay sensitive applications are protected from the changing loads of a delay insensitive application.

3.6 Conclusions

In this chapter we have proposed a novel delay based approach for QoE optimization in wireless networks. Our proposed scheme MBDO is measurement based and can adapt to slowly varying traffic statistics at the BS. It also achieves optimal trade-offs in resource allocation between application types at various system loads based on their sensitivities to mean delay. Through simulations we have shown that MBDO performs better than mean delay optimal Gittins index and Processor Sharing schedulers.

Appendix

3.7 Proof of Corollary 3.3.1

It is shown in Theorem 5.6, [10] that the weighted Gittins index scheduler minimizes the mean expected weighted flow delay in a busy cycle. Using Renewal Reward Theorem (RRT)(see [33]) and the fact that the renewal cycles are identical for all work conserving policies, it can be shown that minimizing the expected weighted flow time in a busy cycle with weights $w_c, c = 1, 2, \ldots, C$ is same as $\mathcal{OP}_2$.

3.8 Proof of Lemma 3.3.2

First we will show that the region is convex. Let $\overline{\mathbf{d}}_1$ and $\overline{\mathbf{d}}_2$ be the two mean delay vectors achieved by the two finite mean delay policies π_1 and π_2, respectively. To achieve the mean delay vector $\phi\overline{\mathbf{d}}_1 + (1-\phi)\overline{\mathbf{d}}_2$, $\phi > 0$, use the policy π_1 with probability ϕ and π_2 with probability $1-\phi$, i.i.d. across busy cycles. The set of achievable finite delay vectors is non-empty because the mean delay for a Processor Sharing discipline is finite as long as the first moments of the service times exist and the load ρ is less than one. See [33] for the proof.

3.9 Proof of Lemma 3.3.3

For brevity, we define

$$\nabla f(\mathbf{d}) := \left(\lambda_1 \frac{\partial f_1(d_1)}{\partial d_1}, \lambda_2 \frac{\partial f_2(d_2)}{\partial d_2}, \ldots, \lambda_C \frac{\partial f_C(d_C)}{\partial d_C}\right)^T$$

.

In $\mathcal{OP}_1$ we take infimum of a continuous, differentiable, strictly convex, lower bounded, increasing (in all coordinates) function over a convex set $\mathcal{D}$ which is a subset of the positive orthant of $\mathbb{R}^C$. Therefore, the objective function of $\mathcal{OP}_1$ has an infimum and it is uniquely achieved by a vector. Let this infimum achieving vector be denoted by $\overline{\mathbf{d}}^*$. Next we show that there exists a work conserving policy which has its mean delay vector same as $\overline{\mathbf{d}}^*$.

The vector $\overline{\mathbf{d}}^*$ is the optimal solution to $\mathcal{OP}_1$ if and only if the following condition is satisfied.

$$\nabla f\left(\overline{\mathbf{d}}^*\right)^T\left(\overline{\mathbf{d}}-\overline{\mathbf{d}}^*\right) \geq 0, \quad \forall \overline{\mathbf{d}} \in \mathcal{D}. \tag{3.22}$$

We have to show that the delay vector $\overline{\mathbf{d}}^*$ can be achieved, i.e. it is in $\mathcal{D}$. We have shown in Corollary 3.3.1 that we can minimize the linear combination of the delays using weighted Gittins index scheduler. Hence, we can minimize $\nabla f\left(\overline{\mathbf{d}}^*\right)^T\overline{\mathbf{d}}$ using weighted Gittins index scheduler using weights as $\nabla f\left(\overline{\mathbf{d}}^*\right)$. From (3.22), we know that this is a necessary and sufficient condition for optimality. This proves that $\overline{\mathbf{d}}^*$ is in $\mathcal{D}$.

3.10 Proof of Lemma 3.3.4

Assume that a given scheduling policy π is used in all busy cycles. Let $N_c(t)$ be the number of flows of class c present in the system at time t. Let N_c be the random variable which denotes the number of customer in class c when the system is stationary. If time instants t_1 and t_2 belong to different busy cycles, then $N_c(t_1)$

is independent of $N_c(t_2)$ because of the independent increment property of Poisson arrivals and the assumption that flow sizes are i.i.d. Therefore we can consider renewal cycles which consist of the idle period and the busy cycle. From the Reward-Renewal theorem, we get that

$$\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} N_i(t)dt = \frac{\mathbb{E}\left[\int_{\text{busy cycle}} N_i(t)dt\right]}{\mathbb{E}\left[T\right]} \quad \text{w.p.1}, \tag{3.23}$$

where T is a random variable denoting the renewal duration for a typical cycle. Note that the mean renewal cycle duration is same irrespective of the scheduling policy, as long as the policy is work conserving. For stationary queues, we have

$$\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} N_i(t)dt = \mathbb{E}\left[N_i\right] \quad \text{w.p.1}. \tag{3.24}$$

Using Little's law, we get $\mathbb{E}\left[N_i\right] = \lambda_i \mathbb{E}\left[D_i\right]$, where D_i is the random variable which denotes the stationary mean delay seen by a typical arriving customer. Substituting this in (3.23), we get that

$$\mathbb{E}\left[D_i\right] = \frac{\mathbb{E}\left[\int_0^T N_i(t)dt\right]}{\lambda_i \mathbb{E}\left[T\right]}. \tag{3.25}$$

If we define $D_i := \frac{\int_0^T N_i(t)dt}{\lambda_i \mathbb{E}[T]}$, then from the above expression D_i is an unbiased estimator for the delay of class i flows under the policy used in the given renewal cycle.

3.11 Proof of Lemma 3.3.5

Let us look at k^{th} busy cycle. Let $N^{(k)}$ be the total number of jobs that arrived in the busy cycle. Then

$$\mathbb{E}\left[\left[\int_{k^{\text{th}}\text{ busy cycle}} N_i(t)dt\right]^2\right] \leq \mathbb{E}\left[\left(N^{(k)}T_k\right)^2\right]. \tag{3.26}$$

From Cauchy–Schwarz inequality, we get that

$$\mathbb{E}\left[\left(N^{(k)}T_k\right)^2\right] \leq \sqrt{\mathbb{E}\left[\left(N^{(k)}\right)^4\right]\mathbb{E}\left[T_k^4\right]}. \tag{3.27}$$

Based on the analysis of the distribution of busy cycle duration in Chapter 27, [33], it can be shown that $\mathbb{E}\left[\left(N^{(k)}\right)^4\right]$ and $\mathbb{E}\left[T_k^4\right]$ are finite when $h_c < \infty, c = 1, 2, \ldots, C$ and $\rho < 1$.

3.12 Proof of Lemma 3.3.6

From the definition of $\overline{\lambda_c}^{(k)}$ and $\overline{T}^{(k)}$ in (3.9) it is true that

$$\lim_{k\to\infty} \overline{\lambda_c}^{(k)} = \lambda_c, \quad \text{w.p.1.} \tag{3.28}$$

$$\lim_{k\to\infty} \overline{T}^{(k)} = \mathbb{E}\left[T\right] \quad \text{w.p.1.} \tag{3.29}$$

Since the function $1/x$ is continuous when $x > 0$, the above results would ensure that

$$\lim_{k\to\infty} \left|\frac{1}{\overline{\lambda_c}^{(k)}\overline{T}^{(k)}} - \frac{1}{\lambda_c\mathbb{E}\left[T\right]}\right| = 0 \quad \text{w.p.1.}, \tag{3.30}$$

However, to prove that the above term converges to zero in expectation, we have to show uniform integrability of the sequence $\left\{\left|\frac{1}{\overline{\lambda_c}^{(k)}\overline{T}^{(k)}} - \frac{1}{\lambda_c\mathbb{E}[T]}\right| \mid k \in \mathbb{N}\right\}$. Therefore, we introduce the thresholding function $z\left(\cdot\right)$.

Let us consider a threshold $\theta > 0$ and define $z(\cdot)$ as follows

$$z(x) := \min(\theta, x), \quad x \geq 0. \tag{3.31}$$

If the value of θ is such that $\theta \geq \max\left\{\frac{1}{\lambda_c \mathbb{E}[T]} \mid c = 1, 2, \ldots, C\right\}$, then we have that

$$\lim_{k\to\infty} |z\left(\frac{1}{\overline{\lambda_c}^{(k)}\overline{T}^{(k)}}\right) - \frac{1}{\lambda_c \mathbb{E}[T]}| = 0 \, \forall c \quad \text{w.p.1.} \tag{3.32}$$

This is because we have chosen θ such that $z\left(\frac{1}{\lambda_c \mathbb{E}[T]}\right) = \frac{1}{\lambda_c \mathbb{E}[T]} \forall c$. Since $|z\left(\frac{1}{\overline{\lambda_c}^{(k)}\overline{T}^{(k)}}\right) - \frac{1}{\lambda_c \mathbb{E}[T]}|$ is bounded for all k, there exists a constant $B < \infty$ such that

$$\mathbb{E}\left[|z\left(\frac{1}{\overline{\lambda_c}^{(k)}\overline{T}^{(k)}}\right) - \frac{1}{\lambda_c \mathbb{E}[T]}|^2\right] < B \quad \forall k. \tag{3.33}$$

This is a sufficient condition for uniform integrability of the sequence.

Next we will show that $\theta = \lambda/\lambda_c^*$ is a good choice for the threshold. We require that $\theta \geq \max\left\{\frac{1}{\lambda_c \mathbb{E}[T]} \mid c = 1, 2, \ldots, C\right\}$. This is ensured if

$$\theta \geq \frac{1}{\lambda_{c^*} \mathbb{E}[T]}. \tag{3.34}$$

From [33], we know that

$$\mathbb{E}[T] = 1/\lambda + \frac{\rho/\lambda}{1-\rho}. \tag{3.35}$$

Substituting the above expression into the inequality (3.34), and using the fact that $\frac{\rho/\lambda}{1-\rho} \geq 0$, we get that

$$\theta \geq \lambda/\lambda_{c^*}. \tag{3.36}$$

3.13 Proof of Lemma 3.3.8

A differential equation is globally asymptotically stable if for any initial condition, eventually it converges to the equilibrium point. Here the equilibrium point is the place where $\frac{d\overline{\mathbf{x}}(t)}{dt} = 0$. From (6.28), the equilibrium point $\mathbf{x}^*$ satisfies:

$$\mathbf{g}^*(\mathbf{x}^*) = \mathbf{x}^*. \tag{3.37}$$

From the definition of $\mathbf{g}^*(\cdot)$, this implies that

$$\nabla f(\mathbf{x}^*)^T \mathbf{x}^* \leq \nabla f(\mathbf{x}^*)^T \mathbf{d} \quad \forall \mathbf{d} \in \mathcal{D}. \tag{3.38}$$

From (3.22), this implies that $\mathbf{x}^*$ is the optimal solution for $\mathcal{OP}_1$, which we proved is unique.

To prove that the differential equation (6.28) is globally asymptotically stable, it is enough to show that we can construct a Lyapunov function $L(\mathbf{d})$ which has a negative drift. Let $L(\mathbf{d}) := f(\mathbf{d}) - f(\mathbf{x}^*)$.

$$\frac{dL(\mathbf{d})}{dt} = \frac{df(\mathbf{d})}{dt}, \tag{3.39}$$

$$= \nabla f(\mathbf{d})^T \frac{d\mathbf{d}}{dt}, \tag{3.40}$$

$$= \nabla f(\mathbf{d})^T \mathbf{g}^*(\mathbf{d}) - \nabla f(\mathbf{d})^T \mathbf{d}, \tag{3.41}$$

$$\leq 0. \tag{3.42}$$

The last inequality follows from the definition of $\mathbf{g}^*(\cdot)$. However, we have to show a strict negative drift for the Lyapunov function. In the RHS of (3.41), $\nabla f(\mathbf{d}) > \mathbf{0}$, $\forall \mathbf{d} \in \mathcal{D}$. This is because of the assumption of strict convex and increasing cost functions. Therefore, for the R.H.S. of (3.41) to be zero, the only possibility is

$\mathbf{g}^*(\mathbf{d}) = \mathbf{d}$. This happens only at the equilibrium point $\mathbf{x}^*$, which is same as the unique solution to $\mathcal{OP}_1$. Hence, the drift is strictly negative when the delay vector $\mathbf{d}$ is away from the equilibrium point.

Chapter 4

Minimizing Mean of Functions of Delays: A Whittle's Index Based Approach

4.1 Introduction

In this chapter[1] we will focus on optimizing mean of functions of delays which can possibly take into higher moments of delays. We will also consider the most general system model for flow based schedulers in this chapter. Traditional work on delay minimization, see e.g. [33, 34], has not simultaneously addressed the following aspects of user experience and resource allocation in wireless networks:

1. QoE of a user may be a non-linear function of the delay to download a file. For example, for many applications users can tolerate delays up to a certain threshold and beyond that the user experience deteriorates gradually [3].

2. Applications may have different sensitivities to delay. Some applications could be more delay tolerant than others, e.g., a simple file download vs interactive web browsing, thus a scheduler can exploit this heterogeneity in delay sensitivity to realize appropriate QoE trade-offs among applications for a range of system loads.

[1]Publications based on this chapter: [52] A. Anand and G. de Veciana, "A Whittle's Index Based Approach for QoE Optimization in Wireless Networks", in Proceedings of ACM SIGMETRICS, 2018 (accepted).

3. User service rates may change with time due to variations in wireless channel characteristics and different users may have different service rates at any given time.

In this chapter we explore addressing the above mentioned issues simultaneously. To that end, we consider a setting in which each user in the system has a job to be served by the BS and it has an associated cost function which is a non-decreasing function of the delay to complete its service. Our aim is to study how to minimize the total expected cost in serving all types of jobs in the system.

The cost function models the QoE of a user as a function of the delay it experiences. The larger the cost, the poorer the QoE perceived by the user. Since the cost function could be non-linear and possibly be different for different jobs this approach takes into account both the non-linearity and the heterogeneity in users' QoE with respect to the delay experienced. Using this approach we can model several useful cost functions, for example, one could consider polynomial functions of delay to model the user's QoE [3] for applications like web browsing and FTP. QoE for stored video streaming (DASH framework) is slightly more complex as it is a function of several parameters like the amount of re-buffering, initial delay and variations in quality of video segments [5]. However, our notion of flow is flexible to accommodate this setting. Indeed current video streaming protocols essentially transfer a sequence of flows associated with video segments. The QoE can then be tied to the delays of these flows/files and/or variability associated with transferring them to the receiver. Cost functions can be obtained through offline studies which collect Mean Opinion

Scores (MOS) from users, see for e.g. [6] and [5]. Henceforth, we shall use cost as a measure of a user's QoE.

An important challenge which is specific to systems with time-varying service rates is realizing the right trade-off between *opportunism and minimizing cost*. If we schedule the user with the highest service rate at all times, then we may increase the overall rate at which the jobs are served. However, this *opportunistic* selection of jobs for service may not be cost optimal, as delay critical jobs with low service rates may see poor cost performance. At the other extreme, if we schedule jobs solely based on their current marginal costs, then we may schedule users when their service rates are low and hence the overall rate at which jobs are processed goes down and overall jobs are delayed, resulting in poor overall cost. Therefore, one needs to find the right balance between being opportunistic and giving priority based on cost. This is explored in this chapter by studying directly how to minimize the expected system cost.

4.1.1 Related Work

We classify the related work into two categories based on the underlying model for job arrivals to the system, namely: 1) *Dynamic* system in which jobs arrive according to a stochastic process (typically a Poisson process) and leave once they are serviced; and 2) *Transient* system in which there is a finite number of jobs at the beginning and no additional arrivals enter the system. We will make further classifications based on the information on job sizes available to the scheduler, for example, some works assume that the job sizes are known to the scheduler whereas

others assume that there is perfect or partial knowledge of job size distributions. Another characteristic which distinguishes various works in the literature is whether they consider a system with time-varying service rates.

4.1.1.1 Dynamic Systems

Many authors have considered mean delay minimization in dynamic systems which process jobs at a constant service rate, see for e.g., [28–30, 32, 34]. If the job sizes are known to the scheduler, it has been shown that the Shortest Remaining Processing Time (SRPT) scheduler is the mean delay optimal scheduler [53]. Under the SRPT policy, the job with the least remaining processing time is scheduled for service at all times. If only the job size distributions are known and the job arrivals form a Poisson process, then it has been proved that the Gittins index scheduler is mean delay optimal [10]. Gittins index schedulers assign a priority to jobs depending on the service received to date and job size distributions. Properties of Gittins index based schedulers for different job size distributions have been studied extensively, see [27–29, 47]. There are few works which consider, however, time-varying service rates in a dynamic system, see [35, 42, 43]. These works either focus on establishing system stability rather than delay-based performance metrics, or propose heuristics which are based on schedulers developed for constant service rate systems.

An interesting line of work which focuses on non-linear cost functions of the *mean* file/job delay in multi-class systems is explored in [38,48]. However, these works deal with cost functions of expected delays rather than expectation of cost functions of the delays experienced by users. This difference is crucial since minimizing the

expectation of the cost functions of delay accounts for higher moments of the delay distribution, whereas, minimizing a metric based on functions of expected delays only accounts for the first moments. Our approach therefore, can model scenarios where the users are sensitive to both the mean and the variability in delay distributions seen by the users. Also, [38, 48] do not consider time-varying job service rates which are typical in wireless settings.

Another line of work which focuses on optimizing non-linear cost functions of delay and queue lengths in multi-class systems includes [54–59]. They consider generalizations of $c\mu$ rule and prove its optimality in heavy traffic regime for various settings. They differ from our work in the following ways.

1. The above works except [59] do not consider time-varying service rates.

2. They do not use the job size information for scheduling, instead, use only the average job size of each class. Using knowledge of job sizes or distribution of actual size is beneficial as it helps us further discriminate jobs based on their sizes.

3. They allow preemption among jobs of different classes but do not allow preemption among jobs of the same class. In wireless systems the jobs sizes could have large variations in their size. Therefore, if we do not allow preemption among jobs of the same class, then the system might suffer from high delays due to a big Head-of-the-line (HOL) job. Also in systems with time-varying service rates one should be able to switch between jobs quickly to opportunis-

tically schedule users. In our work, we allow both preemption within a class and across classes.

In [60], the authors consider optimization of average cost under convex holding costs functions of the number of users in the system. This is different from our setting where we associate a cost with the delay experienced by each user.

4.1.1.2 Transient Systems

Unfortunately, many problems are analytically intractable in the dynamic setting. In particular there is no known optimal solution to the problem of minimizing mean delay in a dynamic system with time-varying service rates [61]. Therefore, many authors have focused on scheduling policies which optimize the relevant metrics in transient systems and propose such solutions as a heuristic for dynamic systems. The effectiveness of these policies are then studied through simulation. Our problem is also analytically intractable in a dynamic system and hence, we shall also consider transient systems. Next we will discuss related work focused on transient systems.

The authors of [46,62] have considered minimizing mean delay in the transient setting where they assume that there is a time-scale separation between service-rate variations and job service times. This means that service rate variations occur at a time-scale which is much smaller than the overall time taken to serve a job. They also assume that the service rate fluctuations are statistically identical and independent across users. These assumptions are valid in situations where the job sizes are large and/or when the service rate variations are due to fast fading. Under this assumption, they have combined opportunistic scheduling with a SRPT like policy to minimize

mean delay. The main issue with this approach is that the assumption of statistically identical service rate variations across users may not be valid in scenarios where there are users with heterogeneous mobility patterns. Also, the assumption of a time-scale separation may not hold when there are many short files to be transmitted.

Minimizing delay based metrics in a transient system with time-varying rates for jobs and without the assumption of time-scale separation between service-rate variations and job service times is unfortunately still analytically intractable due to the associated large state spaces. Recently there have been many works which leverage Whittle's indices to explore the optimization of delay performance in wireless networks in a transient setting [61, 63–65]. However, this line of work has focused only on minimizing weighted linear functions of delay and does not address non-linear cost functions of delay. In [63], the authors have shown that the problem of minimizing mean delay is indexable and derived the Whittle's index when job sizes are geometrically distributed with i.i.d. service rate variations across time. This result was extended to the case with Markovian service rate variations in [64], however, they do not show whether the problem is indexable. In [65], the authors consider a system model where the job sizes are not known but only the job size distributions are known. They derived index policies based on solving a Markov Decision Process, however, they consider only ON-OFF channel model. The approach used in [61] is closely related to our work. They approximate job sizes using shifted Pascal distributions, i.e., a phase-type distribution where each phase has an i.i.d. geometric distribution. They have also derived Whittle's indices when users have heterogeneous two-state i.i.d. channel variations.

Users' States	Parameter of Interest	Priority given to
User $i \rightarrow$ best possible rate, user $j \rightarrow$ lowest possible rate	Service rate	User i
Both users in their lowest possible rate	Residual file sizes	User with the largest residual file size
Both users in their best possible rate	Residual file sizes	Depends on the cost function
Both users in their best possible rate	Probability of best possible rate	User with the lower probability
Users i and j have the same rate	$c_i(t) \leq c_j(t) \quad \forall t$	User j

Table 4.1: Summary of structural properties of ODIP.

4.1.2 Our Contributions

In this chapter we focus on resource allocation strategies to minimize the expectation of possibly non-linear cost functions of job delays in a transient setting with time-varying service rates. To the best of our knowledge, this is the first chapter which simultaneously addresses the challenges of 1) non-linearity and heterogeneity in users' experiences as a function of delay, and 2), time-varying service rates for jobs in a non-heavy traffic regime. To that end, we develop a Whittle's index based scheduling policy, which we denote as Opportunistic Delay Based Index Policy (ODIP), for a transient system. ODIP is simple and easy to implement. At any given time, each user has an index based on its residual file size, service rate and its cost function. In any slot we schedule a user based on the indices. The main results of this chapter are as follows:

1) *Indexability:* We show that our delay/cost minimization problem is *indexable*. This means that we can associate a well-defined index with each possible state. These indices can then be used to assign priorities to active users.

2) *Opportunistic Delay Based Index Policy:* We derive structural properties of the ODIP index for the case of phase-type job size distributions, convex cost functions of delay, and i.i.d. (possibly heterogeneous) two-state service rates for

each user. In particular we show that when a user's instantaneous channel has the best possible rate, then the user has a higher priority than users whose channels are not currently in their respective best possible rates. We then show the following structural properties of the Whittle's index:

1. Given two users with the same holding cost function and identical and independent channel statistics. If both the users are in their respective lowest possible rates, then the user with the *longest remaining service* time gets higher priority. However, if both users are in their respective best possible channel rates, then the priority order between the two users depends on both the cost function and their respective residual file sizes. These properties should be contrasted with the SRPT scheduling policy which gives the highest priority to the user with the smallest residual file size.

2. If there are two users which differ only in the probability of their channel being in the best possible rate, then the user with the lowest probability of being at the best rate gets a higher priority. Therefore, ODIP is opportunistic and gives a higher priority to users likely to be in good rates.

3. If there are two users which differ only in their cost functions and the cost function of one user strictly dominates the other, then the ODIP gives a higher priority to the user with the higher cost function.

These properties are summarized in Table 4.1 where we have characterized the priority order between two users when we vary one parameter of interest while the other parameters are kept the same.

	Information on Jobs	Service Rate
1	Sizes known	Fixed across time slots
2	Geometric distribution and mean job size known	i.i.d. across time, two states
3	Sizes known	i.i.d. across time, multiple states

Table 4.2: Various scenarios for which Whittle's indices are obtained.

Leveraging these structural properties, we derive expressions for the Whittle's index for a few special cases. Each case is characterized by two elements of the system model: 1) information on job size distribution available to the scheduler; and 2) service rate model. The cases considered in this chapter are summarized in Table 4.2. In the scenario where job sizes are known to the scheduler, we shall approximate job sizes using an appropriate phase-type distribution. In all the scenarios, we assume that service rates are independent across users, however, they may not have to be statistically identical.

3) *Simulation Study:* For dynamic systems, we use the results from [66] to show that ODIP is maximally stable, i.e., ODIP ensures system stability if there exists a policy which stabilizes the system for the given system load. We then compare the performance of applying ODIP in a dynamic setting with other policies through simulation. We establish that ODIP makes trade-offs which cannot be achieved by policies which do not take into account the non-linearity of users' QoE in file/job delays. We also show that simple priority based policies perform poorly as compared to ODIP when we consider higher moments of delays in the cost function.

4.1.3 Organization

The remainder of the chapter is organized as follows. In Sec. 4.2, we describe our system model. In Sec. 4.3, we develop our Whittle's index based approach. In Sec. 4.4 we derive the structural properties of ODIP. Expressions for Whittle's index are provided in Sec. 4.5. Performance evaluation results based on simulation are presented in Sec. 4.6.

4.2 System Model

We consider a transient setting where N users are present in the system at time $t = 0$, each with a single job to be served. Since there is a one-to-one correspondence between a user and a job, we shall use the terms user and job interchangeably. Time is assumed to be slotted and is indexed by $t = 0, 1, 2, \ldots$. For simplicity we assume that the scheduler can schedule only one user in a given slot and this decision has to be made at the beginning of the slot. Users leave the system after their jobs are served to completion, and there are no further arrivals.

If a user i is scheduled at time t, then it is served at its current service/channel rate $R_i(t)$ measured in bits/slot. We shall assume that the service rate processes $(R_i(t), t \in \mathbb{Z}^+)$, $i = 1, 2, \ldots, N$ are

1. i.i.d. across time slots and independent across users

2. We assume that $R_i(t) \in \{r_{i,1}, r_{i,2}, \ldots, r_{i,L}\}$, and $R_i(t)$ can take the value $r_{i,l}$ with probability $q_{i,l}$. Without loss of generality we assume that $r_{i,1} > r_{i,2} > \ldots r_{i,L}$ and for all l, $q_{i,l} \neq 0$. Let R_i denote an r.v. with the above distribution.

We call it as multi-state channel model. A restriction of this model to the case with $L = 2$ is called as a *two-state* channel model.

Independence of service rate across users is a reasonable assumption as the user mobilities are generally independent of each other, and hence, they experience independent and heterogeneous wireless channel variations. We can also account for the heterogeneity in long term channel variations like shadowing and path loss variations by selecting different mean service rates for different users. Small time-scale fast fading experienced by mobile users are taken care by the i.i.d. service rate variations across slots.

Further we assume that the job sizes are drawn from a phase-type distribution as in [61]. Thus the job size of user i is modeled by a random variable S_i given by:

$$S_i = \sum_{j=1}^{j_i} S_{i,j}, \tag{4.1}$$

where j_i is the number of phases, and $S_{i,j}, j = 1, 2, \ldots, j_i$ are i.i.d. geometric random variables with mean $1/\mu_i$ bits. We use such phase-type distributions to model the following two cases:

1. If $j_i = 1$, then the phase-type distribution reduces to a geometric distribution. We consider geometric distributions in the second case in Table 4.2.

2. If j_i is large we can model known deterministic file sizes by phase type distributions. For example, if the job size of user i is known to be s_i bits, then one can choose μ_i and j_i such that

$$s_i = j_i/\mu_i. \tag{4.2}$$

For a given value of s_i, as j_i increases, the phase-type approximation of a deterministic/known job size is more accurate. We will use this approximation to study the first and third cases in Table 4.2.

Next we explain how we model the effect of time varying service rates on the service time of a user. Let us first consider an example where the service rate of user i has a constant value of $r_{i,l}$ bits/slot. If $\mu_i r_{i,l} \leq 1$, then the average number of slots to complete the transmission of a phase of user i can be approximated by $1/\mu_i r_{i,l}$. Therefore, if the service rate is fixed at $r_{i,l}$, then the average number of slots to complete a phase has a geometric distribution with parameter $\mu_i r_{i,l}$. From (4.2), we require that $j_i \leq s_i/r_{i,l}$ for the condition $\mu_i r_{i,l} \leq 1$ to be true. To ensure that for all j we have $\mu_i r_{i,l} \leq 1$, we assume that for a given value of s_i, we choose j_i and μ_i such that (4.2) is satisfied and $j_i \leq s_i/r_{i,1}$. We shall assume that s_i is much larger than the number of bits that can be transmitted in a slot, and hence, j_i is large enough to closely approximate s_i with j_i phases.

This idea has a natural extension to time-varying service rates. If the current service rate of user i is $r_{i,l}$, and user i is scheduled for transmission in the current slot, then the probability that its current phase completes in this slot is given by $\mu_i r_{i,l}$. Therefore, the service rate of a user in a given slot modulates the probability of successful completion of the current phase. When all the phases of a user are serviced, then the user leaves the system.

In summary, we shall assume that the scheduler either has knowledge of the exact job sizes or the job size distribution, depending on the case being considered,

see Table 4.2. When we assume that the scheduler has the knowledge of job sizes, we will use phase-type distributions to approximate job sizes. In this setting knowledge of job sizes would imply that the scheduler knows the parameters $\mu_i, i = 1, 2, \ldots, N$ and the number of remaining phases for each user. By contrast when we consider job sizes with geometric distributions, we will assume that the scheduler knows only the parameters of the distributions which are memoryless. We shall also assume that the scheduler knows the service rates of all the users in the next time slot for which a scheduling decision has to be made, and the service rate statistics of all the users.

Let us now introduce the objective function to be optimized:

$$\mathcal{OP}_1 : \quad \min_{\pi \in \Pi} \; \mathbb{E}^{\pi} \left[\sum_{t=1}^{\infty} \sum_{i=1}^{N} c_i(t) \mathbf{1} \left\{ Y_i^{\pi}(t) > 0 \right\} \right], \tag{4.3}$$

where Π is the set of causal and feasible scheduling policies. Here $Y_i^{\pi}(t)$ is a random variable corresponding to the residual file size of user i at time t under policy π and $c_i(t)$ is the holding cost at slot t. A policy is said to be *causal* if it does not assume knowledge of future service rate realizations. A policy is feasible if only one user is scheduled per slot. For a feasible policy π we have that for all i and t:

$$\sum_{i=1}^{N} A_i^{\pi}(t) = 1, \quad A_i^{\pi}(t) \in \{0, 1\}, \quad \text{a.s.}, \tag{4.4}$$

where $A_i^{\pi}(t)$ is a random variable which is equal to one if user i is scheduled for transmission in slot t and zero otherwise.

The holding cost function $c_i(\cdot)$, is a function of time, that captures the sensitivity of user i's QoE to the delay. Suppose the user leaves the system at time d, then the overall accumulated cost, which we denote by $C_i(\cdot)$, is given by $C_i(d) = \sum_{t=0}^{d} c_i(t)$.

Therefore, $c_i(\cdot)$ can be viewed as the marginal cost for a job staying an additional t^{th} slot in the system. The following assumption will be made on these functions.

4.2.1 Assumption on holding cost functions

1. *Monotonicity:* For any user i, $c_i(\cdot)$ is a positive, non-decreasing function of time.

2. *Bounded by polynomials:* There exist real numbers $\delta > 0$, $\zeta > 0$, and $t' \in \mathbb{Z}^+$ such that for $t > t'$ and $i = 1, 2, \ldots, N$, $c_i(t) < \delta t^{\zeta}$.

3. *Non-zero:* For any user i, $c_i(t)$ is not equal to zero for all t.

The monotonicity assumption ensures that a properly interpolated $C_i(\cdot)$ would be a convex function of the holding time. The boundedness assumption is a technical assumption to ensure finiteness of indices for the policy to be discussed in the sequel. The last assumption rules out trivial solutions to $\mathcal{OP}_1$. Note that if for all t and user i $c_i(t) = c$, then $\mathcal{OP}_1$ reduces to the minimization of the overall mean delay.

The remainder of this chapter is focused on exploring resource allocation strategies to solve $\mathcal{OP}_1$.

4.3 Problem Formulation

The minimization problem $\mathcal{OP}_1$ can be viewed as a Markov Decision Process (MDP) when the channel rate variations are Markovian or i.i.d. across time. However, due to the large state space, in general it is not analytically tractable. Therefore, we will consider the so called Whittle's relaxation of $\mathcal{OP}_1$ [67].

The main idea underlying Whittle's relaxation is to relax the constraint of scheduling exactly one user per slot. Instead we add a cost ν for scheduling a user on a given slot, and we minimize a new total cost function which is given by:

$$\mathcal{OP}_2: \quad \min_{\pi \in \tilde{\Pi}}: \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N} c_i(t)\mathbf{1}\left\{Y_i^{\pi}(t) > 0\right\} + \nu \sum_{t=0}^{\infty}\sum_{i=1}^{N} A_i^{\pi}(t)\right], \tag{4.5}$$

where $\tilde{\Pi}$ is the set of causal policies, which may no longer satisfy (4.4). This relaxed problem can now be de-coupled into sub-problems associated with each user i as follows:

$$\mathcal{SP}_i(\nu): \quad \min_{\pi \in \tilde{\Pi}}: \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} c_i(t)\mathbf{1}\left\{Y_i^{\pi}(t) > 0\right\} + \nu \sum_{t=0}^{\infty} A_i^{\pi}(t)\right]. \tag{4.6}$$

Using Whittle's relaxation one can obtain a feasible policy for $\mathcal{OP}_1$ based on the solutions to $\mathcal{SP}_i(\nu), i = 1, 2, \ldots, N$. To that end we first explore the solution to the MDP associated with $\mathcal{SP}_i(\nu)$.

Consider $\mathcal{SP}_i(\nu)$. User i's state is specified by three variables: j the number of remaining phases including the current phase; r the current service rate; and, t the current time. There are two possible actions in a state, to Transmit (T) or Not to Transmit (NT). Let $P\left((j, r, t), (j', r', t'); a\right)$ be the transition probability from the state (j, r, t) to (j', r', t') under the action a. The transition probabilities under the two possible actions are summarized in Table 4.3. Let us consider an example to illustrate how they are obtained: a transition from (j, r, t) to $(j, r_{i,1}, t+1)$ occurs under the action T, if the transmission does not succeed in completing a phase in slot t, which happens with probability $(1 - \mu_i r)$ and the service rate in slot $t+1$ is $r_{i,1}$, which happens with probability $q_{i,1}$. Since these are independent events, we have

Transition Probability	Expression
$P\left((j,r,t),(j,r_{i,l},t+1);T\right)$	$q_{i,l}(1-\mu_i r)$
$P\left((j,r,t),(j-1,r_{i,l},t+1);T\right)$	$q_{i,l}\mu_i r$
$P\left((j,r,t),(j,r_{i,l},t+1);NT\right)$	$q_{i,l}$

Table 4.3: Transition probabilities in state (j,r,t)

$P\left((j,r,t),(j,r_{i,1},t+1);T\right) = q_{i,1}(1-\mu_i r)$. One can similarly define other transition probabilities. The transition probabilities from (j,r,t) to states other than those specified in Table 4.3 are zero.

Based on standard results for MDPs, it can be shown that there exists a time-varying Markov policy which is optimal for $\mathcal{SP}_i(\nu)$, see [68]. Therefore, we shall restrict ourselves to Markov policies. Let $V_i^*(j,r,t;\nu)$ be the total cost under the optimal policy for $\mathcal{SP}_i(\nu)$ starting from the state (j,r,t) for a transmission cost of ν. From the Bellman equations for MDPs, we have that

$$V_i^*(j,r,t;\nu) = \min\Big\{c_i(t) + \overline{V}_i^*(j,t+1;\nu), \\ c_i(t) + \nu + \mu_i r\overline{V}_i^*(j-1,t;\nu) + (1-\mu_i r)\overline{V}_i^*(j,t+1;\nu)\Big\}, \quad (4.7)$$

where $j \in \{1,2,\ldots j_i\}$, $t \in \{0,1,2,3,\ldots\}$, $r \in \{r_{i,1},r_{i,2}\}$, and $\overline{V}_i^*(j,t+1;\nu)$ is defined as follows:

$$\overline{V}_i^*(j,t+1;\nu) := \mathbb{E}\left[V_i^*(j,R_i,t+1;\nu)\right]. \quad (4.8)$$

$\overline{V}_i^*(j,t+1;\nu)$ is the optimal value function averaged over the service rates. Note that a holding cost $c_i(t)$ is incurred for slot t irrespective of the action taken in slot t. From (4.7) and the definition of $\overline{V}_i^*(j,t+1;\nu)$, it is clear that the optimal policy

will transmit in (j, r, t) if and only if the following inequality holds:

$$\nu \leq \mu_i r \Delta_i^* (j, t+1, \nu), \tag{4.9}$$

where $\Delta_i^* (j, t, \nu)$ is defined as follows:

$$\Delta_i^* (j, t, \nu) := \begin{cases} \overline{V}_i^* (j, t; \nu) - \overline{V}_i^* (j-1, t; \nu), & \text{if } j > 1, \\ \overline{V}_i^* (j, t; \nu), & \text{if } j = 1. \end{cases} \tag{4.10}$$

Indeed this policy minimizes the value functions by choosing the function minimizing the R. H. S. in (4.7). The inequality (4.9) is central to the main results of this chapter. It implies it is optimal to transmit in a given state if and only if the marginal decrease in the future cost due to the transmission in the given state is more than the cost ν of transmission.

To develop a feasible solution for $\mathcal{OP}_1$ from $\mathcal{SP}_i(\nu)$, for $i = 1, 2, \ldots, N$, we first show that the problem is indexable. The indexability property, defined in [67] is re-stated here:

Definition The optimization problem $\mathcal{SP}_i(\nu)$ is indexable if for any $j \in \{1, 2, \ldots, j_i\}$, $r \in \{r_{i,1}, r_{i,2}, \ldots, r_{i,L}\}$, and $t \in \{0, 1, 2, \ldots\}$, there exists a value $\nu_i^* (j, r, t)$ such that

1. It is optimal to transmit in (j, r, t) if $\nu < \nu_i^* (j, r, t)$:
2. It is optimal not to transmit in (j, r, t) if $\nu > \nu_i^* (j, r, t)$.
3. It is optimal to either transmit or not to transmit in (j, r, t) if $\nu = \nu_i^* (j, r, t)$.

The value $\nu_i^* (j, r, t)$ is known as the *Whittle's* index.

The indexability property ensures that the optimal action in a given state has a threshold structure in ν. Note that some problems are not indexable, see [67] for examples. However, $\mathcal{SP}_i(\nu)$ is indexable and this result is stated next with a proof given in Appendix 4.8.

Theorem 4.3.1. *Under Assumption 4.2.1, phase-type distribution for file sizes and i.i.d multi-state channel model, $\mathcal{SP}_i(\nu)$ is indexable.*

To construct a feasible solution for $\mathcal{OP}_1$ based on $\mathcal{SP}_i(\nu)$, $i = 1, 2, \ldots, N$, we schedule the user with the highest Whittle's index in each slot. We can interpret the Whittle's index as the lowest price at which it is optimal not to transmit in a given state. A higher Whittle's index means that the state is better suited for transmission. This is a natural heuristic which arises from the relaxation of $\mathcal{OP}_2$. Whittle's index based policies are known to have good performance in practice, see [61, 67]. The remainder of this chapter will focus on the derivation and characteristics of the Whittle's index for various scenarios mentioned in Table 4.2.

4.4 Whittle's Index

In this section we will characterize key structural properties of the Whittle's Index for $\mathcal{SP}_i(\nu)$. The first main result is given in the following theorem, which is proved in Appendix 4.11.

Theorem 4.4.1. *Under Assumption 4.2.1, phase-type distribution for file sizes and i.i.d multi-state channel model, the Whittle's index for any user i in phase $j \in$*

$\{1, 2, \ldots, j_i\}$ *is such that*

$$\nu_i^* \left(j, r_{i,1}, t\right) = \infty, \tag{4.11}$$

$$\nu_i^* \left(j, r_{i,l}, t\right) < \infty \quad l \neq 1. \tag{4.12}$$

Theorem 4.4.1 implies that for any finite value of ν, it is optimal to transmit when the current rate is $r_{i,1}$. Since the lowest price at which it is optimal not to transmit in $(j, r_{i,1}, t)$ is ∞. Since the Whittle's index for users experiencing their lowest possible rate is finite, they will have a lower priority than users experiencing their best possible channel rate. A similar result was proved in [61] in the setting of constant holding costs. Theorem 4.4.1 is thus a generalization of that result to convex holding costs.

Since the Whittle's index is ∞ for all users currently experiencing their highest possible service rates, scheduling users based on the Whittle's index policy alone is not feasible. We require a further tie-breaking rule to obtain a feasible policy. We will refer to (4.11) and (4.12) as the *primary* indices and the tie-breaking rule which we will derive next will be based on *secondary* indices. The secondary index is defined based on the discounted version of the problem and determined as the asymptotic behavior of the Whittle's index as the discount factor approaches one. The discounted version of $\mathcal{OP}_2$ is given by:

$$\mathcal{OP}_2^{\beta}: \quad \min_{\pi \in \Pi} \; \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t \left(\sum_{i=1}^{N} c_i(t) \mathbf{1} \left\{ Y_i^{\pi}(t) > 0 \right\} \right) + \nu \sum_{t=0}^{\infty} \beta^t \left(\sum_{i=1}^{N} A_i^{\pi}(t) \right) \right], \tag{4.13}$$

where $\beta \in [0, 1)$ is the discount factor. The discounted sub-problem for user i is in

turn given by:

$$\mathcal{SP}_i^{\beta}(\nu): \quad \min_{\pi\in\Pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\beta^t c_i(t)\mathbf{1}\left\{Y_i^{\pi}(t)>0\right\}+\nu\sum_{t=0}^{\infty}\beta^t A_i^{\pi}(t)\right]. \tag{4.14}$$

We can define the Whittle's index for the discounted version of the problem as follows:

Definition Let $\mathcal{P}(j,r,t)$ denote the set of prices such that for $\nu' \in \mathcal{P}(j,r,t)$ it is optimal not to transmit in (j,r,t) when $\nu > \nu'$. We let the Whittle's index for the discounted problem for a user i in state (j,r,t), denoted by $\nu^*_{i,\beta}(j,r,t)$, be $\nu^*_{i,\beta}(j,r,t) := \inf\{\nu' : \nu' \in P(j,r,t)\}$.

The above definition differs from that of the un-discounted case since we do not show or require that the discounted problem be indexable.

The tie-breaking rule for users in their respective best possible service rate is based on the observation that for any $j \in \{1,2,\ldots,j_i\}$ and $r \in \{r_{i,1}, r_{i,2}, \ldots, r_{i,L}\}$

$$\lim_{\beta\to 1}\nu^*_{i,\beta}(j,r,t) = \nu^*_i(j,r,t). \tag{4.15}$$

The tie-breaking rule for user i is obtained by considering the asymptote of $\nu^*_{i,\beta}(j,r_{i,1},t)$ as $\beta \to 1$ which we shall call the *secondary* index. This is the same terminology as used in [61]. We define the secondary index for state $(j,r_{i,1},t)$ as given by

$$\xi^*_i(j,r_{i,1},t) := \lim_{\beta\to 1}(1-\beta)\,\nu^*_{i,\beta}(j,r_{i,1},t). \tag{4.16}$$

Since we have defined the secondary index in terms of a limit we have to show that the limit exists and it is finite. This is given by the next result which is proved in Appendix 4.12.1.

Theorem 4.4.2. *Under Assumption 4.2.1, phase-type distribution for job sizes and i.i.d. multi-state channel model, we have that for any* $j \in \{1, 2, \ldots, j_i\}$ *and* $t \geq 0$*, the secondary index* $\xi_i^* (j, r_{i,1}, t)$ *is finite and* $\xi_i^* (j, r_{i,1}, t) < \infty$*.*

With these in hand we can now describe our Whittle's index based policy, which we shall refer to as Opportunistic Delay Based Index Policy (ODIP).

4.4.1 Opportunistic Delay Based Index Policy (ODIP)

In any time-slot t, we will schedule a user based on the flow-chart exhibited in Fig. 4.1. We first check if there is any user whose current service rate is the best possible. If there is at least one such user, then we schedule the user with the highest secondary index for transmission. If there is no such user, then we will schedule the user with the highest primary index. The selected user in that case will have a finite primary index as guaranteed by Thm. 4.4.6.

The computation of indices in ODIP requires cost functions of various applications, channel statistics of users, and flow sizes. When a new user joins the network, there many not enough channel measurements to get reliable channel statistics. Hence, when a new user joins the system, one has to use the typical channel state distribution observed in the network. This can be obtained through offline data collection. As time evolves, one can then update the channel statistics from the channel measurements at the Base Station (BS). Below we develop some qualitative results on the primary and secondary indices, which characterize the scheduling policy.

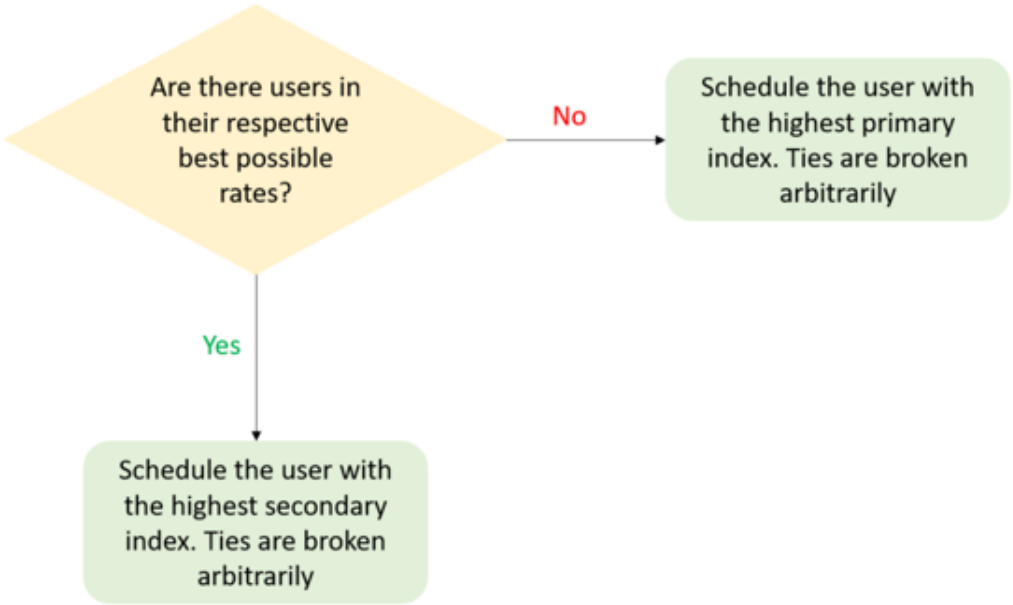


Figure 4.1: Flow-chart for ODIP.

4.4.2 Qualitative Results for Two-state Channel Model

In this section for simplicity we shall restrict ourselves to a two-state channel model, i.e., $L = 2$. First we compare the indices of two users where the cost function of one user dominates that of the other user. The proof of this result is given in Appendix 4.14.3.

Theorem 4.4.3. *Suppose users i and l have i.i.d. two-state service rate variations. If their holding cost functions are such that for all $t \geq 0$, $c_i(t) \leq c_l(t)$, then for any $j \in \{1, 2, \ldots, j_i\}$, $r \in \{r_{i,1}, r_{i,2}\}$ and $t \geq 0$, we have that $\Delta_i^*(j, t, \nu) \leq \Delta_l^*(j, t, \nu)$.*

The above theorem is used to prove the following two important corollaries.

Corollary 4.4.4. *Suppose users i and l have i.i.d. two-state service rate variations. If their holding cost functions are such that for all $t \geq 0$, $c_i(t) \leq c_l(t)$, then for any $j \in \{1, 2, \ldots, j_i\}$, $r \in \{r_{i,1}, r_{i,2}\}$ and $t \geq 0$, $\nu_i^*(j, r, t) \leq \nu_l^*(j, r, t)$ and $\xi_i^*(j, r, t) \leq \xi_l^*(j, r, t)$.*

Corollary 4.4.5. *For any user i and phase $j \in \{1, 2, \ldots, j_i\}$, and $t \geq 0$, $\nu_i^*(j, r_{i,2}, t) \leq \nu_i^*(j, r_{i,2}, t+1)$ and $\xi_i^*(j, r_{i,1}, t) \leq \xi_i^*(j, r_{i,1}, t+1)$.*

Corollary 4.4.4 implies that we will give priority to users with 'steeper' holding cost functions. Corollary 4.4.5 implies that the priority of a user increases with the time spent in the system. This is because of the non-decreasing property of $c_i(t)$, i.e., convex cumulative holding costs. Corollary 4.4.5 will be useful for studying the structural properties of the primary and secondary indices. The main result for the primary index is given below and it is proved in Appendix 4.14.1.

Theorem 4.4.6. *Under Assumption 4.2.1, phase-type file size distributions and i.i.d. two-state channel model, for any* $(j', r_{i,2}, t')$ *and* $(j, r_{i,2}, t)$*, if* $j' \geq j$ *and* $j'+t' \geq j+t$*, then* $\nu_i^*(j, r_{i,2}, t) \leq \nu_i^*(j', r_{i,2}, t')$ *.*

The j' and t' which satisfy the condition in Thm. 4.4.6 for a given j and t are shown in Fig. 4.2. An important corollary to this theorem is given next

Corollary 4.4.7. $\nu_i^*(j, r_{i,2}, t)$ *is a non-decreasing function of both* j *and* t*.*

The above result implies that for any two identical users with the same i.i.d. service rate statistics, holding cost function if they both are in their lowest possible service rates, then the user with the *largest* number of phases remaining to be completed will have priority. This is similar to the Longest Remaining Time First (LRTF) scheduling policy. Intuitively, this is because a user with a large residual job size will have to transmit when service rates are low to reduce the overall holding cost, whereas, a user with a small residual job size can be served opportunistically, i.e., wait for a slot with higher service rate. Since $\nu_i^*(j, r_{i,2}, t)$ is a non-decreasing function of time, the priority for that user in the next slot is higher if we make a transition to $\nu_i^*(j, r_{i,2}, t+1)$, i.e., either if we do not transmit in state $(j, r_{i,2}, t)$ or we transmit and fail to complete a phase. However, if we transmit in $(j, r_{i,2}, t)$, complete a phase, and make a transition to $(j-1, r_{i,2}, t+1)$, then the priority may not necessarily increase.

Next we will consider the secondary index. We define the set of 'reachable' states from any state (j, r, t) as follows:

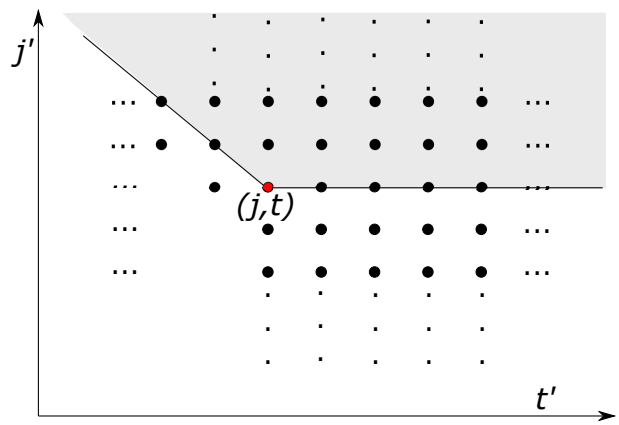


Figure 4.2: The shaded region shows j' and t' which satisfy the conditions in Thm. 4.4.6.

Definition If there exists a Markov policy with non-zero transition probability from (j, r, t) to (j', r', t') (in one or more time slots), then (j', r', t') is said to be *reachable* from (j, r, t). The set of all reachable states from (j, r, t) is denoted by $\mathcal{R}(j, r, t)$.

Note that our system model permits only transitions from (j, r, t) to (j', r', t') such that $j' \leq j$ and $t' > t$. Also, we can complete at most one phase in a slot. Therefore, (j', r', t') is reachable from (j, r, t), if and only if 1) $j' \leq j$ and 2) $j' + t' \geq j + t$. The value of r' can be either $r_{i,1}$ or $r_{i,2}$, irrespective of the values of j, r, and t. This can be visualized with the help of Fig. 4.3. The states are exhibited as a two dimensional grid, with time represented in the x-axis and the residual number of phases on the y-axis. We do not explicitly show the channel rate in the representation but it can be understood from the context of the discussion. For $r \in \{r_{i,1}, r_{i,2}\}$, the shaded region represents $\mathcal{R}(j, r, t)$, i.e., if j' and t' are in the shaded region, then both $(j', r_{i,1}, t')$ and $(j', r_{i,2}, t')$ are in $\mathcal{R}(j, r, t)$. For the secondary index, we have the following result which is proved in Appendix 4.14.2.

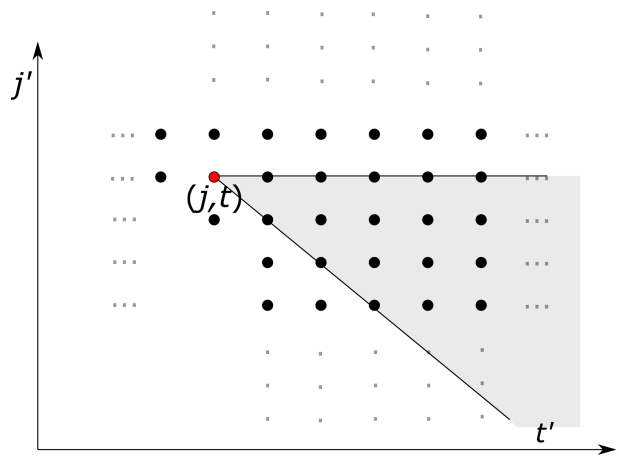


Figure 4.3: The shaded region shows j' and t' such that (j', r, t') is reachable from (j, r, t) for any $r \in \{r_{i,1}, r_{i,2}\}$.

Theorem 4.4.8. *Under Assumption 4.2.1, phase-type file size distributions and i.i.d. two-state channel model, if* $(j', r_{i,1}, t')$ *is reachable from* $(j, r_{i,1}, t)$*, then* $\xi_i^*(j', r_{i,1}, t') \geq \xi_i^*(j, r_{i,1}, t)$.

The above theorem implies that for a given user i, the secondary index in slot $t+1$ is higher than $\xi_i^*(j, r_{i,1}, t)$, whatever is the action taken in the state $(j, r_{i,1}, t)$. Therefore, similar to the primary index, the secondary index for the user in slot $t+1$ is higher if we do not transmit in $(j, r_{i,1}, t)$ or if we transmit and fail to complete a phase. However, unlike the primary index, the secondary index also increases in slot $t+1$ if we complete a phase in slot t.

The previous results do not help us characterize the ODIP when users have heterogeneous channels and/or cases where one cost function does not dominate the other. For this we have to find exact expressions for primary and secondary indices. These qualitative results, however, give basic insights and help us in further deriving

exact expressions. We derive expressions for indices in the next section.

4.5 Quantitative Results

We consider the three different cases mentioned in Table 4.2. Starting with the simplest case in which we schedule users with fixed service rates and where job sizes are known to the scheduler.

4.5.1 Fixed Service Rate, Known Deterministic File Sizes

As explained in Sec. 4.2, we model the job sizes using phase-type distributions. The fixed service rate is a special case of the two-level model described in the previous section where $q_{i,1} = 1$. Suppose user i is served at a fixed rate r_i bits/slot. In this case we shall assume that for all i, $\mu_i r_i = 1$. This would imply that if user i is scheduled in a given slot, then it will complete the phase with probability one. One can also visualize this as splitting the job into j_i equal parts where each part has a size of r_i bits and if user i is selected for transmission, then one part is serviced in that slot. Our main result for this setting is the following. A proof of this result is given in Appendix 4.15.1.

Theorem 4.5.1. *Under Assumption 4.2.1, fixed service rate and phase type service requirement with $\mu_i r_i = 1$, ODIP reduces to scheduling a user with the highest secondary index. For a user i in state (j, r_i, t), the secondary index $\xi_i^*(j, r_i, t)$ is given by*

$$\xi_i^*(j, r_i, t) = \frac{1}{j} c_i(t + j). \tag{4.17}$$

The priority rule described above considers two factors– the residual service time and the cost function of the user. Recall that j, corresponds to the number of phases left to complete, i.e., the number of slots that will be required for that particular user to complete service. Therefore, on the R.H.S. of (4.17), the term $1/j$ gives more weight to a user with a smaller residual service time and the term $c_i(t+j)$ gives more weight to users with a steeper cost function. Note that $c_i(t+j)$ is the holding cost when the user i leaves the system if it is served without preemption till completion.

This policy can be viewed as a generalization of SRPT, which is known to be the mean delay optimal policy when the job sizes are known and service rate is fixed. If the holding cost function is constant and is same for all users, then (4.17) reduces to SRPT. With more general cost functions, the priority rule in (4.17) achieves a trade-off between accelerating short flows and giving priority to users with higher holding cost functions.

4.5.2 Two-state I.I.D. Service Rates, Geometric File Sizes

In this sub-section we consider the two-state channel model described in Sec. 4.2. We shall assume that file sizes are geometric. This is a special case of the phase-type distribution where each user has one phase. Since there is only one-phase for each user, we do not have to track the phase of active users. However, we shall explicitly represent this by $j = 1$ to maintain consistent notation as in other cases. We state the main result for this setting next which is proved in Appendix 4.15.2.

Theorem 4.5.2. *Under Assumption 4.2.1 on $c_i(t)$, geometric file sizes and two-state*

i.i.d. service rate variations, the primary index for user i is given by

$$\nu_i^*\left(1, r_{i,2}, t\right) = \frac{\mu_i \overline{r}_i r_{i,2}}{\overline{r}_i - r_{i,2}} \sum_{k=1}^{\infty} c_i(t+k)\left(1 - \mu_i \overline{r}_i\right)^{k-1} \tag{4.18}$$

where $\overline{r}_i := q_{i,1} r_{i,1} + (1 - q_{i,1}) r_{i,2}$. *The secondary index in turn is given by*

$$\xi_i^*\left(1, r_{i,1}, t\right) = q_i\left(\mu_i r_{i,1}\right)^2 \sum_{k=1}^{\infty} c_i(t+k)\left(1 - q_i \mu_i r_{i,1}\right)^{k-1}. \tag{4.19}$$

Let us now consider how the indices depend on the residual job size, cost functions and the service rates. Since the file sizes are geometric, and thus memoryless, the residual file size at any slot is given by $1/\mu_i$ bits. The larger the value of μ_i the smaller the residual file size. For a given $c_i(t)$, $r_{i,1}$, $r_{i,2}$, and $q_{i,1}$, it can be shown that $\nu_i^*\left(1, r_{i,2}, t\right)$ is a non-increasing function of μ_i. This means that among the users who have the same cost function and who are not in their best possible rates, the users with larger residual file sizes are given priority over users with smaller residual file sizes. The intuition behind this is similar to that underlying Corollary 4.4.7. However, unlike $\nu_i^*\left(1, r_{i,2}, t\right)$, the properties associated with the changes in $\xi_i^*\left(1, r_{i,1}, t\right)$ as function of μ_i depend on $c_i(t)$.

For a given $\overline{r}_i$ and $c_i(t)$, $\nu_i^*\left(1, r_{i,2}, t\right)$ and $\xi_i^*\left(1, r_{i,1}, t\right)$ are increasing functions of $r_{i,2}$ and $r_{i,1}$, respectively. This means that we give priority to users with better service rates when the other parameters are the same. It can be easily seen that a higher holding cost function results in a higher value for $\nu_i^*\left(1, r_{i,2}, t\right)$ and $\xi_i^*\left(1, r_{i,1}, t\right)$. Therefore, the primary and the secondary indices together achieve a trade-off between minimizing cost and opportunistically scheduling users. Note that if for all t we have $c_i(t) = c_i$, then ODIP reduces to the *Size-Aware Whittle's Index* SWA policy derived in [61]. Our results are thus the generalization of SWA.

4.5.3 Multi-state I.I.D. Service Rates, Known Deterministic File Sizes

The exact expressions for the primary indices are analytically intractable. Therefore, we will derive a lower bound. We state the main result for this setting which is proved in Appendix 4.15.3.

Theorem 4.5.3. *Under Assumption 4.2.1, phase-type file size distributions, and i.i.d. multi-state channels for any $j \in \{1, 2, \ldots, j_i\}$, $t \geq 0$, and $l \in \{2, 3, \ldots, L\}$, the primary index for user i is lower bounded by:*

$$\nu_i^*(j, r_{i,l}, t) \geq \frac{\mu_i \left(\sum_{n=1}^{l} q_{i,n} r_{i,n}\right) r_{i,l}}{\sum_{n=1}^{l} q_{i,n} r_{i,n} - r_{i,l}\left(\sum_{n=1}^{l} q_{i,n}\right)} \sum_{m=0}^{\infty} c_i(t+j-1+m) \left(1 - \mu_i \sum_{n=1}^{L} q_{i,n} r_{i,n}\right)^m. \tag{4.20}$$

The secondary index for user i is given by the following equation.

$$\xi_i^*(j, r_{i,1}, t) = \frac{q_{i,1}(\mu_i r_{i,1})^2}{j} \left[H_{i,1}^{\dagger}(j, t+1) - H_{i,1}^{\dagger}(j-1, t+1)\right], \tag{4.21}$$

where $H_{i,1}^{\dagger}(j, t)$ is the average total holding cost (transmission cost not included) incurred by the policy in which transmissions are done only when channel state $r = r_{i,1}$, when there are j remaining phases at time t. Its value is obtained by solving the following set of equations for all t:

$$H_{i,1}^{\dagger}(j, t) = c_i(t) + (1 - \mu_i q_{i,1} r_{i,1}) H_{i,1}^{\dagger}(j, t) \tag{4.22}$$

$$+ \mu_i q_{i,1} r_{i,1} H_{i,1}^{\dagger}(j-1, t), \; j = 2, 3, \ldots, j_i, \tag{4.23}$$

$$H_{i,1}^{\dagger}(1, t) = \sum_{k=0}^{\infty} c_i(t) (1 - \mu_i q_{i,1} r_{i,1})^k, \tag{4.24}$$

$$H_{i,1}^{\dagger}(0, t) = 0. \tag{4.25}$$

The lower bound (4.20) retains the properties mentioned in Thm. 4.4.3 and 4.4.6. Therefore, it retains the priority ordering of various states for a given user as well as the priority ordering among states for two users when the cost function of one user dominates the other. However, it may affect the priority ordering between two users when cost functions do not dominate each other. This will not adversely affect the performance of our ODIP because at moderate to high system loads there would be a sufficient number of users in the system such that at least one user is in its best possible rate and therefore, the scheduling is primarily done based on secondary indices for which we can derive exact expressions.

Let us consider an example for the computation of $\xi_i^*(j, r_{i,1}, t)$. If $c_i(t) = t$, then we will get the following expression for $H_{i,1}^{\dagger}(j, t)$.

$$H_{i,1}^{\dagger}(j, t) = \frac{j}{\mu_i q_{i,1} r_{i,1}} t + \frac{j(j+1)}{2} \left[\frac{1 - \mu_i q_{i,1} r_{i,1}}{(\mu_i q_{i,1} r_{i,1})^2} \right]. \tag{4.26}$$

Substituting (4.26) in (4.21), we get the following equation for secondary index.

$$\xi_i^*(j, r_{i,1}, t) = \frac{\mu_i r_{i,1}(t+1)}{j} + \frac{1 - \mu_i q_{i,1} r_{i,1}}{q_{i,1}}. \tag{4.27}$$

In the above example, the secondary index is a non-decreasing function of the remaining service requirement j for a given t. However, in general, for a given t, the manner in which $\xi_i^*(j, r_{i,1}, t)$ varies as a function of j depends on $c_i(t)$. Also for a given j, $\xi_i^*(j, r_{i,1}, t)$ is a non-decreasing function of time. From Corollary 4.4.5 this holds for any $c_i(t)$ which is a non-decreasing function of t. Another interesting property is that $\xi_i^*(j, r_{i,1}, t)$ is a non-increasing function of $q_{i,1}$, if all the other parameters are fixed. This can be proved using (4.21). A smaller $q_{i,1}$ implies that there is less

chance of user i being in its best possible rate. Since it is a rare 'good' event, it is good to opportunistically use it to serve user i. Therefore, if all parameters except $q_{i,1}$ are the same for a set of users, then the user with the smallest $q_{i,1}$ gets the highest priority in this set. This is reminiscent of quantile based scheduling [69] and references therein.

4.6 Dynamic System

In this section we discuss properties and performance of ODIP when applied to a dynamic setting. As we have stated previously, we propose to use ODIP as a heuristic for the dynamic setting. Instead of starting with a finite number of jobs at time $t = 0$, here we shall consider a system in which jobs arrive according to a Poisson process. Jobs are classified into K different classes based on their holding cost functions. All jobs in a class have the same cost function. Let λ_k be the arrival rate of jobs of class k. We shall assume the same channel model for jobs as in Sec. 4.2. We shall also assume that all jobs associated with a class have i.i.d. service rate distributions, both across time and between users. Therefore, with a slight abuse of notation, instead of specifying holding cost functions and the service rates of the individual jobs, we will specify them for an entire class. For example, $c_k(\cdot)$ is the cost function of class k and $r_{k,1}$ is the maximum service rate for a job of class k. Finally to specify the holding cost of job in given slot, it will be based on the sojourn time since its arrival to the system.

In a dynamic system, the first concern is whether the system is stable for a given set of arrival rates $\lambda_k, k = 1, 2, \ldots, K$. Let S_k be a r.v. denoting the job size

(in bits) of a typical class k job. If the system stability is not maintained, then the delays experienced by jobs may grow unboundedly. From Theorem 5.2 in [66], we have the following result on the stability of the system under ODIP.

Corollary 4.6.1. *In a dynamic multi-class system with Poisson arrivals and multi-state i.i.d. service rates for jobs, ODIP is maximally stable and the arrival rates must satisfy:*

$$\sum_{k=1}^{K} \frac{\lambda_k \mathbb{E}\left[S_k\right]}{r_{k,1}} < 1. \tag{4.28}$$

Proof. A policy is said to be maximally stable if it can stabilize the system for any arrival rate for which a stabilizing policy exists. It has been shown in [66] that a class of policies called Best Rate (BR) policies are maximally stable. A BR policy serves a user whose current rare is best possible whenever such a user is present in the system. Our ODIP is also a BR policy and hence, maximally stable. □

We will evaluate the delay cost performance of ODIP for dynamic systems via simulation. In our simulations, we will classify the arriving jobs into two classes based on their QoE requirements. Let λ_1 and λ_2 be the average arrival rates of jobs of Class 1 and 2, respectively. We assume that we can make a scheduling decision every 0.01 sec, i.e., slot duration is 0.01 sec. A job of Class 1 has cost $C_1(d) = d^2$ for a delay of d seconds. We use the gradient of $C_1(\cdot)$ to obtain $c_1(\cdot)$, i.e., $c_i(t) = C_i(t) - C_i(t-1)$. Similarly, a job of Class 2 has cost $C_2(d) = \left(\frac{d}{1.5}\right)^2$. Therefore, Class 1 users are more sensitive to delays than Class 2 users. For Class 1 traffic the cost increases steeply after a delay of one second, whereas the Class 2 traffic can tolerate delays upto 1.5 seconds. We shall compare our scheme with the following three policies:

1. *Size-Aware Whittle's Index Policy (SW):* This is a BR policy which considers the optimization of weighted mean delay in dynamic systems. It is a special case of ODIP which minimizes a weighted function of mean delays. The weight could be different for each user. This approach does not consider the non-linearity of user experience with respect to delay. In the sequel we will show that even if we optimize the weights for SW scheduling such that it has the least cost among all SW policies for a given set of arrival rates, the costs due to this policy are still higher than the costs under ODIP.

2. *Proportional Fair (PF):* This is a commonly used rate-based policy in wireless networks in which at any time we schedule a user with the highest ratio of its current rate to the average rate allocated to the user previously. When the service rate is constant for each user, then this policy reduces to Processor Sharing. In a dynamic system with time-varying service rates, it has been shown in [43] that PF is maximally stable. We will compare our scheme with a weighted version of the PF algorithm where we assign a higher weight to the more delay sensitive class. We shall optimize the weight for each arrival rate vector so that the cost is least among all weighted Proportional Fair schedulers. We will show that even with optimized weights this policy cannot achieve good QoE.

3. *Priority Based Policy:* We consider a simple priority based policy where we give absolute preemptive priority to the more delay sensitive Class 1 jobs over Class 2 jobs and within each class we will schedule users according to SW discipline

with unit weights for all jobs. However, this policy is not a maximally stable and hence, we can compare with this policy only for smaller range of arrival rates.

In all the simulation scenarios considered, we shall generate jobs having Pareto file size distribution with c.c.d.f. $\overline{G}(x) = \left(\frac{4}{x+4}\right)^5$, where the size is measured in Mbits. This distribution has a mean of 1 Mbit. For practical systems, these parameters can be scaled appropriately. We now discuss the simulation results for two different settings based on the service rate model: fixed and time-variant service rates.

4.6.1 Fixed Service Rate

In this section we shall assume that all jobs can be processed at a constant rate of 1 Mbps. If we fix the arrival rate of a class and sweep the arrival rate of the other class, we will get two sets of simulation results. In Fig. 4.4, we compare the average cost of all the policies when λ_1 is fixed at 0.5 arrivals/sec. and λ_2 is swept. Similarly in Fig. 4.5, we have fixed λ_2 at 0.5 arrivals/sec. and have swept λ_1. In both the scenarios, ODIP performs better than the other policies. Note that we have optimized the weights of SWA and weighted PF for each data point. ODIP performs better than other policies because it takes into the non-linearity of cost functions. To understand this better, we have plotted the average cost per class when we sweep λ_1 and λ_2 in Figures 4.6 and 4.7, respectively. We have only plotted the comparisons with SW as it is the second best policy in terms of the average cost. In both the scenarios, as the overall system load increases, ODIP protects the delay sensitive Class 1 at the expense of other class. SW which considers the minimization

of weighted linear functions of delays does not have the required flexibility to make trade-offs as it can only give a higher weight to the more delay sensitive Class 1 jobs without considering the time spent by the jobs in the system. The priority scheme fully prioritizes Class 1 traffic and hence, jobs of Class 2 traffic have poor delay responses, which has resulted in higher overall cost.

4.6.2 Time-varying Service Rate

Next we compare ODIP with other policies in a system where users have time-varying service rates. We consider a two-state service rate for all jobs which is i.i.d. across time and users. The maximum rate is 1 Mbps and the minimum rate is 0.5 Mbps, and probability of being in the best possible rate is 0.5 for both the classes.

As in the fixed service case, we compare the average cost under different policies. Note that the priority based scheme is not maximally stable and hence, we cannot simulate it for the full range of arrival rates in the stability region. Figures 4.8and 4.9 exhibit the average cost versus λ_2 and λ_1 sweeps, respectively. In both scenarios, ODIP performs better than other policies. The priority scheme performs poorly because it does not fully exploit the opportunism in the system and becomes unstable. The weighted PF does not take into account the delay of jobs while scheduling. Therefore, it has a poor cost performance. SW and ODIP have similar costs at low loads, however, as load increases, ODIP performs better than SW. We have also compared the average cost per class in Fig 4.10 and 4.11. As the load increases, ODIP is able to balance the delays experienced by both the classes,

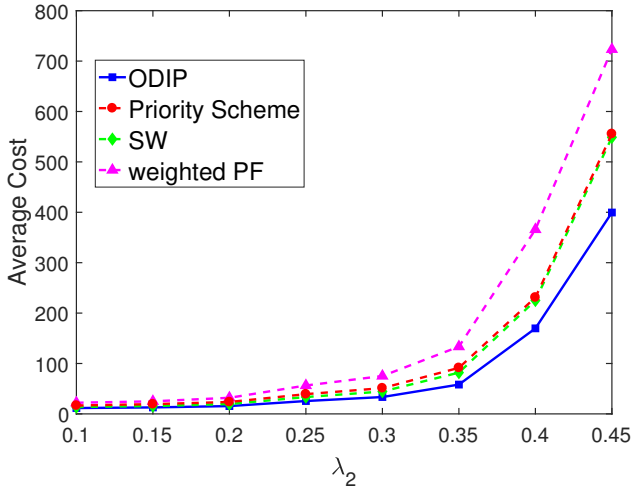


Figure 4.4: Average cost as a function of λ_2 ($\lambda_1 = 0.5$ arrivals/sec.) in the system with fixed service rates for jobs (1 Mbps).

wheres, SW can only give a higher weight to the more delay sensitive Class 1 at the expense of Class 2. This results in a better performance for Class 1, but the delays experienced by Class 2 traffic easily exceeds 1.5 seconds and hence, results in a larger cost.

4.7 Conclusions

In this chapter we have explored the three inter-related problems in scheduling for wireless systems: 1) non-linear relationships between a user's QoE and flow delays; 2) managing load dependent QoE trade-offs among heterogeneous application classes; and 3) striking a good balance between opportunistic scheduling and greedy QoE optimization. We have used Whittle's relaxation to develop our proposed scheme ODIP and to study its structural properties. Simulations confirm the effectiveness

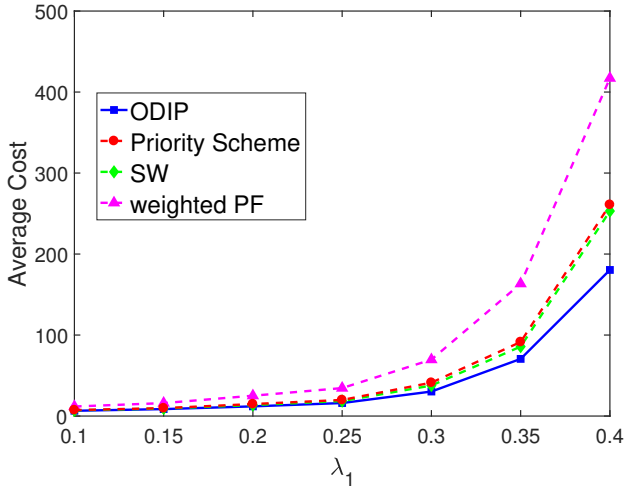


Figure 4.5: Average cost as a function of λ_1 ($\lambda_2 = 0.5$ arrivals/sec.) in the system with fixed service rates for jobs (1 Mbps).

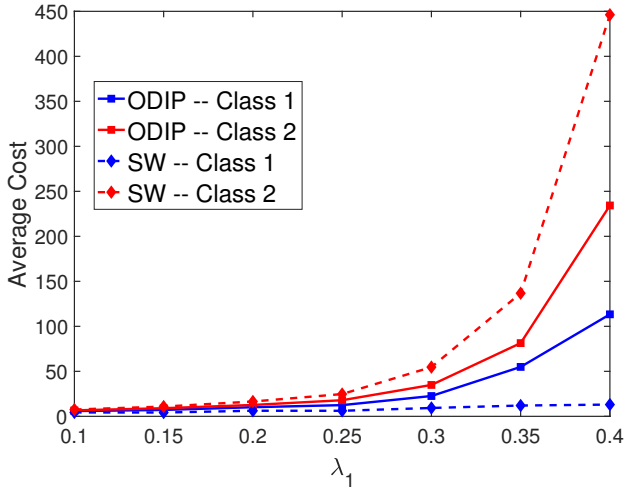


Figure 4.6: Average cost as a function of λ_1 ($\lambda_2 = 0.5$ arrivals/sec.) in the system with fixed service rates for jobs (1 Mbps).

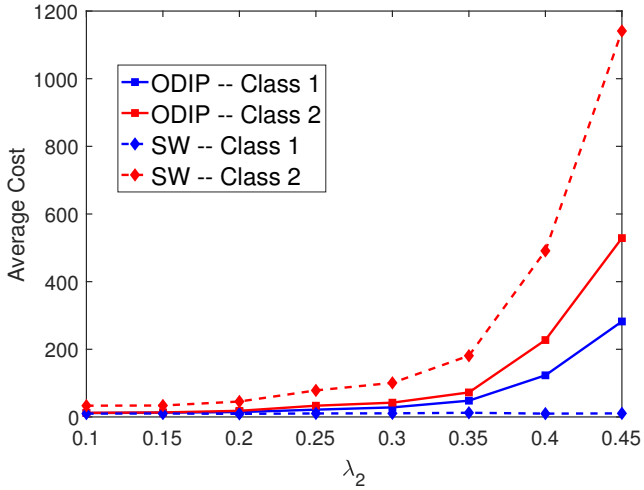


Figure 4.7: Average cost as a function of λ_2 ($\lambda_1 = 0.5$ arrivals/sec.) in the system with fixed service rates for jobs (1 Mbps).

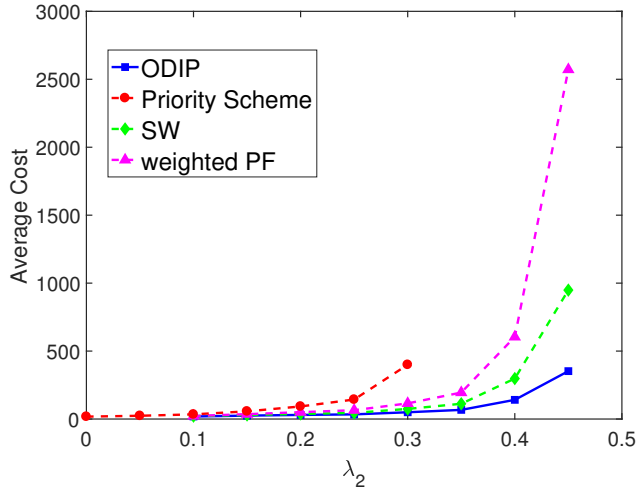


Figure 4.8: Average cost as a function of λ_2 ($\lambda_1 = 0.5$ arrivals/sec.) in the system with time-varying service rates for jobs (peak rate 1 Mbps).

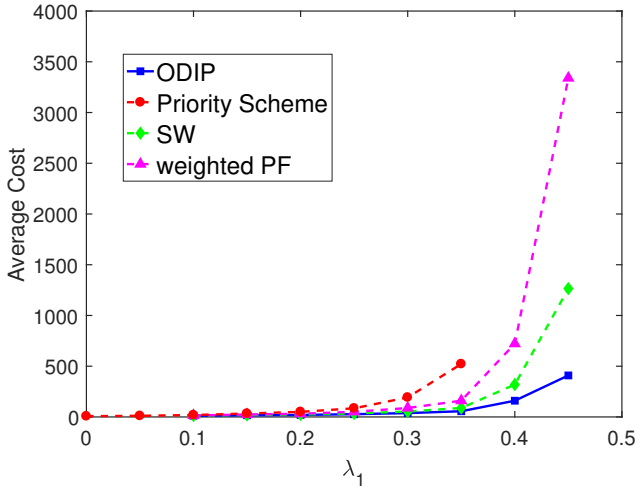


Figure 4.9: Average cost as a function of λ_1 ($\lambda_2 = 0.5$ arrivals/sec.) in the system with time-varying service rates for jobs (peak rate 1 Mbps).

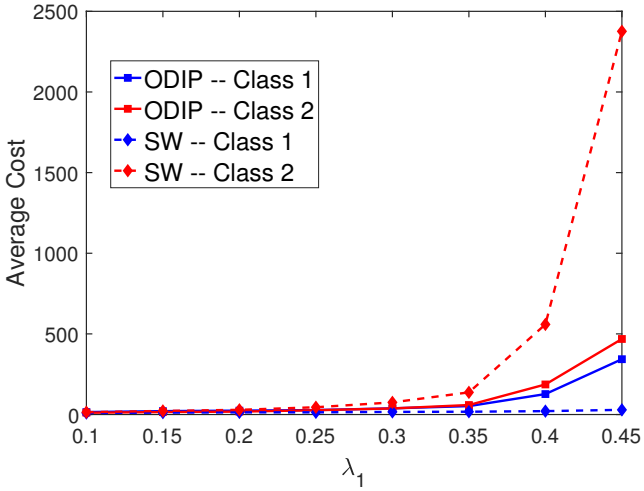


Figure 4.10: Average cost as a function of λ_1 ($\lambda_2 = 0.5$ arrivals/sec.) in the system with time-varying service rates for jobs (peak rate 1 Mbps).

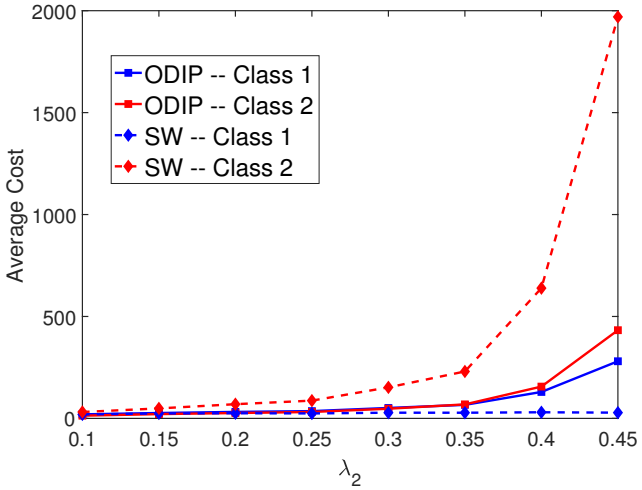


Figure 4.11: Average cost as a function of λ_2 ($\lambda_1 = 0.5$ arrivals/sec.) in the system with time-varying service rates for jobs (peak rate 1 Mbps).

of ODIP in achieving the complex QoE trade-offs among different traffic classes for a range of system loads.

Appendix

4.8 Proof of Theorem 4.3.1

We will use the following definitions to explain the proofs:

$$\Delta_i^*(j,t,\nu) := \begin{cases} \overline{V}_i^*(j,t;\nu) - \overline{V}_i^*(j-1,t;\nu), & \text{if } j > 1, \\ \overline{V}_i^*(j,t;\nu), & \text{if } j = 1, \end{cases} \tag{4.29}$$

$$\Delta_{i,\beta}^*(j,t,\nu) := \begin{cases} \overline{V}_{i,\beta}^*(j,t;\nu) - \overline{V}_{i,\beta}^*(j-1,t;\nu), & \text{if } j > 1, \\ \overline{V}_{i,\beta}^*(j,t;\nu), & \text{if } j = 1. \end{cases} \tag{4.30}$$

We use the following two important lemmas which are proved in Sec. 4.9 to prove Thm. 4.4.1.

Lemma 4.8.1. *For any user i, $j \in \{1,2,\ldots,j_i\}$, $t \geq 0$, and $\nu > 0$, we have that $\Delta_i^*(j,t,\nu) > \frac{\nu}{\mu_i r_{i,1}}$.*

Lemma 4.8.2. *For any user i, $j \in \{1,2,\ldots,j_i\}$, and $t \geq 0$, we have that*

1. *$\Delta_i^*(j,t,\nu)$ is an non-decreasing concave function of ν and the following equation has a fixed point:*

$$\mu_i r_{i,l} \Delta_i^*(j,t,\nu) = \nu \quad l \in \{2,3,\ldots,L\} \tag{4.31}$$

2. *$\Delta_i^*(j,t,0) > 0$.*

For $r \in \{r_{i,1}, r_{i,2}, \ldots, r_{i,L}\}$, and a fixed j and t, let us look at the fixed point of $\mu_i r \Delta_i^*(j,t,\nu)$, i.e., the solution to the following equation:

$$\nu = \mu_i r \Delta_i^*(j,t,\nu). \tag{4.32}$$

By Lemma 4.8.1 and the fact that $\Delta_i^*(j,t,\nu)$ is continuous in ν, there does not exist a fixed point, i.e., solution to $\mu_i r \Delta_i^*(j,t,\nu) = \nu$ when $r = r_{i,1}$ and for any finite ν we have that $\nu < \mu_i r_{i,1} \Delta_i^*(j,t,\nu)$. From Bellman equation (4.7), this is implies that it is always optimal to transmit when $r = r_{i,1}$ for any $\nu < \infty$. Hence, $\nu_i^*(j, r_{i,1}, t) = \infty$.

Property 1 in Lemma 4.8.2 shows that there exists a fixed point for $\mu_i r \Delta_i^*(j,t,\nu)$ when $r = r_{i,l}$, $l = 2, 3, \ldots, L$. Let us choose any such fixed point as the Whittle's index denoted by $\nu_i^*(j, r_{i,l}, t)$. For $\nu < \nu_i^*(j, r_{i,l}, t)$, from properties 1 and 2 in Lemma 4.8.2, $\mu_i r_{i,l} \Delta_i^*(j,t,\nu) \geq \nu$. Therefore, from the Bellman equations (4.7) it is optimal to transmit in $(j, r_{i,l}, t)$. Similarly, for $\nu > \nu_i^*(j, r_{i,l}, t)$, it is optimal not to transmit in $(j, r_{i,l}, t)$. Thus we conclude that the problem is indexable for the multi-level i.i.d. service rate model.

4.9 Proof of Lemmas

4.9.1 Proof of Lemma 4.8.1

We will prove this inequality by contradiction. Suppose that the inequality is not true. From the Bellman equations (4.7), this would imply that it is not optimal to transmit in states $(j, r_{i,1}, t)$ $l \in \{1, 2, \ldots, L\}$. From this we get the following:

$$\overline{V}_i^*(j,t;\nu) = c_i(t) + \overline{V}_i^*(j,t+1;\nu). \tag{4.33}$$

By Assumption 4.2.1 that for any t, there exists a t' such that $t' > t$ and $c_i(t') > 0$, it can be easily shown that $\overline{V}_i^*(j,t;\nu) < \overline{V}_i^*(j,t+1;\nu)$. However, (4.33) implies a contradiction. Hence, the inequality
$\Delta_i^*(j,t,\nu) > \frac{\nu}{\mu_i r_{i,1}}$ must be true.

4.9.2 Proof of Lemma 4.8.2

We will use the following two intermediate lemmas proved in Sec. 4.10 to prove Lemma 4.8.2.

Lemma 4.9.1. *Let the truncated holding cost function for user i is defined as follows:*

$$c_i^{(k)}(t) := \begin{cases} c_i(t), & \text{if} \quad t \leq k, \\ c_i(k), & \text{if} \quad t > k. \end{cases} \tag{4.34}$$

Let $\overline{V}_i^{*,(k)}(j,t;\nu)$ be the corresponding averaged optimal value function under the cost function $c_i^{(k)}(\cdot)$, then for all j, t, and ν we have that

$$\lim_{k\to\infty} \overline{V}_i^{*,(k)}(j,t;\nu) = \overline{V}_i^{*}(j,t;\nu). \tag{4.35}$$

Lemma 4.9.2. *If the cost function of user i is constant in time, i.e., $c_i(t) = c$, then under the multi-state channel model we have that $\Delta_i^*(j,t,\nu)$ is independent of j and t and is a concave, non-decreasing, piecewise linear function of ν.*

The proof of Lemma 4.8.2 is as follows. First we shall prove the non-decreasing property of $\Delta_i^*(j,t,\nu)$ with respect to ν.

1) ***Non-decreasing:*** First we shall prove the non-decreasing property of $\Delta_i^*(j,t,\nu)$. To that end we will approximate $c(t)$ with a sequence of truncated holding cost functions $\{c_i^{(k)}(t), k = 1,2,3,\ldots\}$ as defined in (4.34). Let us define $\Delta_i^{*,(k)}(j,t,\nu) := \overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^{*,(k)}(j-1,t;\nu)$. We will show that $\Delta_i^{*,(k)}(j,t,\nu)$ is a non-decreasing function of ν and use Lemma 4.9.1 to conclude that $\Delta_i^*(j,t,\nu)$ is also a non-decreasing function of ν.

$c_i^{(k)}(\cdot)$ is a 'truncated' approximation of the holding cost function, in which the holding cost has a constant value of $c_i(k)$ after time k. Since the holding cost function is fixed after time k, the policy in the state (j, r, t') for any $t' > k$ is the same. Also, $\overline{V}_i^{*,(k)}(j, t; \nu)$ depends only on the actions in other states $\left(j', r', t'\right)$ such that $j' \leq j$ and $t' \geq t$. Because of this we have to consider a finite number of feasible policies and the decisions that have to be made over time interval $[0, k]$.

Let $\pi^* \left(c_i^{(k)}(\cdot), \nu\right)$ be the optimal policy when the price is ν and the holding cost function is $c_i^{(k)}(\cdot)$. If we fix a policy π, then the overall average cumulative holding cost from the state (j, r, t), denoted by $\overline{V}_i^{\pi,k}(j, t; \nu)$ is a linear function of ν. Therefore, to find $\overline{V}_i^{*,(k)}(j, t; \nu)$, we are taking a minimum over a finite number of linear functions in ν when the cost functions is $c_i^{(k)}(\cdot)$. This implies that $\overline{V}_i^{*,(k)}(j, t; \nu)$ is a piece-wise linear function in ν and is concave. Therefore, for any ν, there exists a *neighborhood* $N_\delta(\nu)$ where the policy $\pi^* \left(c_i^{(k)}(\cdot), \nu\right)$ is optimal. When we say neighborhood, we mean any of the three sets: $(\nu - \delta, \nu]$, $(\nu - \delta, \nu + \delta)$, or $[\nu, \nu + \delta)$, where $\delta > 0$. Next we state an important lemma which is proved in Sec. 4.10.

Lemma 4.9.3. $\Delta_i^{*,(k)}(j, t, \nu)$ *is non-decreasing function of* ν *in* $N_\delta(\nu)$.

Since $\Delta_i^{*,(k)}(j, t, \nu)$ is continuous in ν and piece-wise linear function, the above lemma implies that $\Delta_i^{*,(k)}(j, t, \nu)$ is a non-decreasing function of ν. Therefore, $\lim_{k\to\infty} \Delta_i^{*,(k)}(j, t, \nu) = \Delta_i^*(j, t, \nu)$ is also a non-decreasing function of ν.

Concavity: Next we shall prove the concavity of $\Delta_i^*(j, t, \nu)$. We shall use truncated holding cost functions to prove this property. We shall prove that $\Delta_i^{*,(k)}(j, t, \nu)$ is concave in ν. Using the fact that concavity is preserved on taking

the limit $\lim_{k\to\infty} \Delta_i^{*,(k)}(j,t,\nu)$ we will conclude that $\Delta_i^*(j,t,\nu)$ is concave in ν. We shall use prove the concavity of $\Delta_i^{*,(k)}(j,t,\nu)$ by induction. Let us assume that $t \leq k$.

Base Case: For $t' \geq k$, we have that $\Delta_i^{*,(k)}(j,t',\nu)$ is a concave function of $\nu\ \forall\ i$ and j. This is proved in Lemma 4.9.2.

Induction Hypothesis: Let us assume that for any user i, $\Delta_i^{*,(k)}(j,t',\nu)$ is a concave function ν for $t+1 \leq t' < k$.

We have to prove that $\Delta_i^{*,(k)}(j,t,\nu)$ is a concave function of $\nu\ \forall\ j$ and k. We can re-write $\Delta_i^{*,(k)}(j,t,\nu)$ as follows:

$$\Delta_i^{*,(k)}(j,t,\nu) = \Delta_i^{*,(k)}(j,t+1,\nu) + \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^{*,(k)}(j,t+1,\nu)\right\}\right] \\ - \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^{*,(k)}(j-1,t+1,\nu)\right\}\right], \quad (4.36)$$

where the expectation is computed with respect to R_i which is a r.v. with the same distribution as $R_i(t)$. Define

$$\tilde{l} := \max\left\{l :\ \nu \leq \mu_i r_{i,l} \Delta_i^{*,(k)}(j,t+1,\nu)\right\}.$$

From Lemma 4.8.1, $\tilde{l} \geq 1$. Therefore, the first two terms in the R.H.S. of (4.36) sum upto $\nu + \left(1 - \mu_i \sum_{l=1}^{\tilde{l}} q_{i,l} r_{i,l}\right) \Delta_i^{*,(k)}(j,t+1,\nu)$, which is a concave function of ν from the induction hypothesis. Similarly one can argue that the third term in the R.H.S. of (4.36) is also a concave function of ν. Since sum of concave functions is a concave function, $\Delta_i^{*,(k)}(j,t,\nu)$ is also a concave function. Therefore, from Lemma 4.9.1, $\Delta_i^*(j,t,\nu)$ is also concave in ν.

To prove that (4.32) has a fixed point, we will have to show that curves $\mu_i r_{i,l} \Delta_i^*(j,t,\nu)$ as a function of ν and the linear function ν intersect when $l \neq 1$.

For this we derive an upper bound on $\Delta_i^*(j,t,\nu)$. If we use the optimal policy when starting with $j-1$ stages at time t for the first $j-1$ phases when starting with j phases at time t, we will get an upper bound for $\overline{V}_i^*(j,t;\nu)$ which is given below:

$$\overline{V}_i^*(j,t;\nu) \leq \mathbb{E}\left[\overline{V}_i^*(1,T(j-1,t,j-1;\nu);\nu)\right] + \overline{V}_i^*(j-1,t;\nu), \tag{4.37}$$

where $\mathbb{E}\left[\overline{V}_i^*(1,T(j-1,t,j-1;\nu);\nu)\right]$ is the average cumulative cost to finish one remaining phase if the time taken to finish the first $j-1$ phases is $T(j-1,t,j-1;\nu)$. Using this we can re-write $\Delta_i^*(j,t,\nu)$ as follows:

$$\Delta_i^*(j,t,\nu) = \overline{V}_i^*(j,t;\nu) - \overline{V}_i^*(j-1,t;\nu) \tag{4.38}$$

$$\leq \mathbb{E}\left[\overline{V}_i^*(1,T(j-1,t,j-1;\nu);\nu)\right]. \tag{4.39}$$

We can bound the term the R.H.S. of the above equation with the average cumulative cost under the policy in which we transmit only when $R_i(t) = r_{i,1}$. We get the following:

$$\mathbb{E}\left[\overline{V}_i^*(1,T(j-1,t,j-1;\nu);\nu)\right] \leq \mathbb{E}\left[H_i^\dagger(j,T(j-1,t,j-1;\nu))\right] + \frac{\nu}{\mu_i r_{i,1}}, \tag{4.40}$$

where $H_i^\dagger(j,t)$ is the cumulative average holding cost under the policy which transmits only when $R_i(t) = r_{i,1}$. Under this policy, the probability of success of completing a phase given that the user i transmits is given by $\mu_i r_{i,1}$. Hence, the average transmission cost is given by $\frac{\nu}{\mu_i r_{i,1}}$. The expectations in the above expression are all with the respect to the r.v. $T(j-1,t,j-1;\nu)$. So we have that

$$\Delta_i^*(j,t,\nu) \leq \mathbb{E}\left[H_i^\dagger(j,T(j-1,t,j-1;\nu))\right] + \frac{\nu}{\mu_i r_{i,1}}. \tag{4.41}$$

Let $T_i^{\dagger}(j-1)$ be a r.v. denoting the time taken to finish $j-1$ stages under the policy in which transmits only when $R_i(t) = r_{i,1}$. Since it is optimal to transmit $R_i(t) = r_{i,1}$, we have that $T_i^{\dagger}(j-1) \overset{s.t.}{>} T(j-1, t, j-1; \nu)$. Since $\mathbb{E}\left[H_i^{\dagger}(j,t)\right]$ is a non-decreasing function of t, we have a further bound on $\Delta_i^*(j,t,\nu)$ and is given below:

$$\Delta_i^*(j,t,\nu) \leq \mathbb{E}\left[H_i^{\dagger}\left(j, T_i^{\dagger}(j-1)\right)\right] + \frac{\nu}{\mu_i r_{i,1}}. \tag{4.42}$$

Therefore, $\Delta_i^*(j,t,\nu)$ is a concave, non-decreasing function of ν which is upper bounded by an affine function of ν with slope $1/\mu_i r_{i,1}$. This implies that for $l \neq 1$, $\mu_i r_{i,l} \Delta_i^*(j,t,\nu)$ is upper bounded by a function of ν with slope strictly less than one since $\frac{\mu_i r_{i,l}}{\mu_i r_{i,1}} < 1$. Hence, $\mu_i r_{i,1} \Delta_i^*(j,t,\nu)$ should intersect with ν and therefore, there exists a fixed point. Hence, this part of the lemma is proved.

2) When $\nu = 0$, it is optimal to transmit in all states. Therefore, the average cumulative cost includes only the holding cost component. $\Delta_i^*(j,t,0) = H_i^*(j,t,0) - H_i^*(j-1,t,0)$. The average cumulative cost to finish j phases is more than the cost to finish $j-1$ phases if we transmit in all states, and hence, $\Delta_i^*(j,t,0) > 0$.

4.10 Proof of Auxiliary Lemmas: Indexibilty

4.10.1 Proof of Lemma 4.9.1

Let us consider $|\overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^*(j,t;\nu)|$. Let us also consider $t \leq k$. This is not a restrictive assumption as we would be taking the limit $k \to \infty$ for a fixed t in the sequel. First we will find an upper bound on the term $|\overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^*(j,t;\nu)|$. Let $\pi^*(c_i(\cdot),\nu)$ be the optimal policy when the cost function is $c_i(\cdot)$. Similarly, $\pi^*\left(c_i^{(k)}(\cdot),\nu\right)$ be the optimal policy when the cost function is $c_i^{(k)}(\cdot)$. To get an upper

bound we shall use the following hybrid policy which combines both $\pi^*\left(c_i(\cdot),\nu\right)$ and $\pi^*\left(c_i^{(k)}(\cdot),\nu\right)$

- For $t \leq k$, use $\pi^*\left(c_i(\cdot),\nu\right)$.
- For $t > k$, use $\pi^*\left(c_i^{(k)}(\cdot),\nu\right)$.

This policy is clearly sub-optimal for $c_i^{(k)}(\cdot)$ and hence, the average cumulative holding cost under this hybrid policy will be an upper bound on $\overline{V}_i^{*,(k)}\left(j,t;\nu\right)$. Let the total cost under this policy be denoted by $V_i^{h,(k)}\left(j,t;\nu\right)$.

We shall use a coupling argument next. Let us consider two systems, one which uses the hybrid policy with holding cost function $c_i^{(k)}(\cdot)$ and the other with $\pi^*\left(c_i(\cdot),\nu\right)$ and holding cost function $c_i(\cdot)$. Let us couple the job size random variables and the channel state process. Let us consider two mutually exclusive and exhaustive events 1) user i is served to completion before slot k 2) user i is served to completion after slot k. Conditioned on event 1, for any sample path, the difference between the cumulative cost of both the systems is zero. This is because, the policies are same and the holding are also the same for $t \leq k$. Let us look at event 2. From lemma 4.8.1 and Bellman equations (4.7), it is always to optimal to transmit when $R_i(t) = r_{i,1}\ \forall\ t$. Event 2 happens only if there less than j phases are successfully completed in $k-t$ slots. Therefore, probability of event 2 is upper bounded by $\sum_{j'=0}^{j}\binom{k-t}{j'}\left(q_{i,1}\mu_i r_{i,1}\right)^{j'}\left(1-q_{i,1}\mu_i r_{i,1}\right)^{k-t-j'}$. If event 2 occurs, then there will be non-zero residual phases that has to be served after slot k. We can bound this cost by taking $\max_{j''\leq j} V_i^{h,(k)}\left(j'',k;\nu\right) - \overline{V}_i^*\left(j'',k;\nu\right)$. From the above discussion we have

the following inequalities:

$$\overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^{*}(j,t;\nu) \leq V_i^{h,(k)}(j,t;\nu) - \overline{V}_i^{*}(j,t;\nu) \tag{4.43}$$

$$\leq \sum_{j'=0}^{j} \binom{k-t}{j'} (\tilde{p}_i)^{j'} (1-\tilde{p}_i)^{k-t-j'} \tag{4.44}$$

$$\times \max_{j'' \leq j} \left[V_i^{h,(k)}\left(j'',k;\nu\right) - \overline{V}_i^{*}\left(j'',k;\nu\right) \right], \tag{4.45}$$

where $\tilde{p}_i := q_{i,1}\mu_i r_{i,1}$. Since we have assumed that the holding cost functions are upper bounded by polynomials, the term $V_i^{h,(k)}\left(j'',k;\nu\right) - \overline{V}_i^{*}\left(j'',k;\nu\right)$ is a polynomial function of k. This is because the under $c_i^{(k)}(\cdot)$, holding cost is a constant $c_i(k)$ for $t \geq k$, and the average holding cost to complete any phase is just scaling an appropriate geometric random variable with $c_i(k)$. Note that this term is multiplied by an exponentially decaying function of k in (4.44). Therefore, on taking the limit $k \to \infty$, the R. H. S. goes to zero. Hence, we have shown that the upper bound goes to zero. We can derive a lower bound for $\overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^{*}(j,t;\nu)$ in a similar manner by interchanging the roles of $\pi^*(c_i(\cdot),\nu)$ and $\pi^*\left(c_i^{(k)}(\cdot),\nu\right)$ in the construction of hybrid policy and then using that to upper bound $\overline{V}_i^{*}(j,t;\nu)$. We shall skip the details in the interest of space. Therefore, we have that $\lim_{k\to\infty}|\overline{V}_i^{*,(k)}(j,t;\nu) - \overline{V}_i^{*}(j,t;\nu)| = 0$.

4.10.2 Proof of Lemma 4.9.2

Suppose if we have that $\forall t\ c_i(t) = c$, then it should be clear that $\Delta_i^*(j,t,\nu)$ is independent of t. To study the effect of j, from the definition of $\Delta_i^*(j,t,\nu)$ we can

write the following equation:

$$\Delta_i^*(j,t,\nu) = \Delta_i^*(j,t+1,\nu) + \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^*(j,t+1,\nu)\right\}\right] \\ - \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^*(j-1,t+1,\nu)\right\}\right], \quad (4.46)$$

where R_i is a r.v. denoting the random service rate in a typical slot. Since $\Delta_i^*(j,t,\nu)$ is independent of t under constant holding cost assumption, we shall suppress the argument t in the sequel. Then the above equation simplifies to the following:

$$\mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^*(j,\nu)\right\}\right] = \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \Delta_i^*(j-1,\nu)\right\}\right]. \quad (4.47)$$

Since the above equation holds for any service rate distribution, we have that $\Delta_i^*(j;\nu)$ must be independent of j. Therefore, we can re-write $\Delta_i^*(j;\nu)$ in the following manner:

$$\Delta_i^*(j,\nu) = \Delta_i^*(1,\nu) = \overline{V}_i^*(1;\nu). \quad (4.48)$$

From Bellman equations (4.7), if it is optimal to transmit in $R_i(t) = r_{i,l}$, then it is also optimal to transmit when $R_i(t) = r_{i,l'}$ for $l' < l$. We shall restrict ourselves to such policies. Let π be a policy where we transmit when $R_i(t) = r_{i,l'}$ for $l' = 1, 2, \ldots, l$. The average cumulative cost under such a policy is given by:

$$\overline{V}_i^\pi(1;\nu) = \frac{c}{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}} + \frac{\nu \sum_{l'=1}^{l} q_{i,l'}}{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}}. \quad (4.49)$$

This is because of probability of transmitting in a slot is $\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}$, and therefore the number of slots required to complete a phase on an average is $\frac{1}{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}}$. Given that one transmits, probability of succeeding in completing a phase in a given slot is given by $\frac{\sum_{l'=1}^{l} q_{i,l'} r_{i,l'}}{\sum_{l'=1}^{l} q_{i,l'}}$. This is because the average rate conditioned on the fact that

user i transmits is $\frac{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}}{\sum_{l'=1}^{l} q_{i,l'}}$. Therefore, for any ν, to determine the optimal cost to go, we need only to take a minimum over a finite number of policies parametrized by $l = 1, 2, \ldots, L$. For each policy, the average cumulative cost is a non-decreasing linear function of ν. Therefore, from (4.48) $\Delta_i^* (j, \nu)$ is a non-decreasing, piecewise linear, concave function of ν.

4.10.3 Proof of Lemma 4.9.3

Let $Y_i^{*,(k)}(t)$ be an r.v. denoting the residual number of phases of user i at time t. We can write $\overline{V}_i^* (j, t; \nu)$ as follows:

$$\overline{V}_i^{*,(k)} (j, t; \nu) = H_i^{*,(k)} (j, t, \nu) + \nu \mathbb{E}^{\pi^* \left(c_i^{(k)}(\cdot), \nu\right)} \left[\sum_{t'=t}^{\infty} A_i(t') | Y_i^{*,(k)}(t) = j \right], \tag{4.50}$$

where $H_i^{*,(k)} (j, t, \nu)$ is the average cumulative holding cost starting with j phases at time t and the second term is the average cumulative transmission cost incurred due to transmissions under the policy $\pi^* \left(c_i^{(k)}(\cdot), \nu\right)$. Therefore, we can re-write $\Delta_i^{*,(k)} (j, t, \nu)$ as follows:

$$\begin{aligned} &\Delta_i^{*,(k)} (j, t, \nu) = H_i^{*,(k)} (j, t, \nu) - H_i^{*,(k)} (j-1, t, \nu) \\ &+\nu \left(\mathbb{E}^{\pi^* \left(c_i^{(k)}(\cdot), \nu\right)} \left[\sum_{t'=t}^{\infty} A_i(t') | Y_i^{*,(k)}(t) = j \right] - \mathbb{E}^{\pi^* \left(c_i^{(k)}(\cdot), \nu\right)} \left[\sum_{t'=t}^{\infty} A_i(t') | Y_i^{*,(k)}(t) = j-1 \right] \right). \end{aligned} \tag{4.51}$$

Since the optimal policy is same for all $\nu \in N_\delta (\nu)$, the term $H_i^{*,(k)} (j, t, \nu) - H_i^{*,(k)} (j-1, t, \nu)$ is independent of ν for $\nu \in N_\delta (\nu)$. If we can show that the slope of second term with respect to ν is greater than zero, then we can prove this lemma. To that end let us define $T (j, t, k; \nu)$ to be the random variable denoting the time to complete

first k phases starting with j phases at time t, when the price is ν, under the optimal policy.

First we show that $T(j,t,j-1;\nu) \overset{s.t.}{\leq} T(j-1,t,j-1;\nu)$, i.e., the time to complete the first $j-1$ phases when starting with j phases at time t is stochastically less than the time to complete $j-1$ phases when starting with $j-1$ phases at time t. To see this, we can re-write $\overline{V}_i^{*,(k)}(j,t;\nu)$ as

$$\overline{V}_i^{*,(k)}(j,t;\nu) = \text{Average cumulative cost to finish first } j-1 \text{ phases} + \text{Average cumulative cost to finish the last phase.} \quad (4.52)$$

Individually each of the two terms on the R.H.S. above consists of a part due to the holding cost and a part due to the transmission cost ν. Also, note the two terms in the R.H.S. are not independent of each other. If the time to complete to first $j-1$ phases is longer, then the average cumulative holding cost in completing the last phase is also higher because the transmission of the last phase starts at a later time and the holding cost function is non-deceasing function of time. If $T(j,t,j-1;\nu) \overset{s.t.}{>} T(j-1,t,j-1;\nu)$, then we can replace the policy for the first $j-1$ phases when starting with j phases with the optimal policy for $j-1$ stages when starting with $j-1$ stages and therefore, we can obtain a better policy. Hence, $T(j,t,j-1;\nu) \overset{s.t.}{\leq} T(j-1,t,j-1;\nu)$.

Next observe that the average cumulative cost in completing $j-1$ phases starting with $j-1$ phases initially has to be less than the average cumulative cost in completing $j-1$ phases when starting with j phases. $T(j,t,j-1;\nu) \overset{s.t.}{\leq} T(j-1,t,j-1;\nu)$ would imply that the average cumulative holding cost in com-

pleting the first $j-1$ phases when starting with j phases is less than the average cumulative holding cost in completing $j-1$ phases when starting with $j-1$ phases. The only way that the average cumulative cost to complete the $j-1$ phases when starting with j phases is more than the average cumulative cost in completing $j-1$ phases when starting with $j-1$ phases is by having a larger average cumulative transmission cost. This would imply that the slope of the R.H.S. of (4.51) is positive with respect to ν. Hence, the Lemma 4.9.3 is proven.

4.11 Proof of Theorem 4.4.1

In order to find the Whittle's index for any state (j, r, t), we have to find the fixed point of the following equation:

$$\nu = \mu_i r \Delta_i^* (j, t, \nu) . \tag{4.53}$$

We have already shown in the Appendix 4.8 that when $r = r_{i,1}$ there does not exist a finite fixed point for the above equation and the $\nu_i^* (j, r_{i,1}, t) = \infty$. We have also shown that there exists a finite fixed point when $r \neq r_{i,1}$, and therefore, for $l \neq 1$, $\nu_i^* (j, r_{i,l}, t) < \infty$. Hence, proved.

4.12 Secondary Index

4.12.1 Proof of Theorem 4.4.2

Consider the discounted sub-problem $\mathcal{SP}_i^\beta$. From the definition of Whittle's index for the discounted case, to find $\nu_{i,\beta}^* (j, r_{i,1}, t)$, we have to find the supremum of

the fixed points of the following equation

$$\nu = \mu_i r_{i,1} \beta \Delta^*_{i,\beta}(j,t,\nu). \tag{4.54}$$

The supremum of the fixed points of the above equation is finite because of the following reasons

1. When $\nu = 0$, it is optimal to transmit in all states and $\Delta^*_{i,\beta}(j,t,0) > 0$.

2. When $\nu \to \infty$, it is optimal not to transmit in any of the states, and $\lim_{\nu\to\infty} \Delta^*_{i,\beta}(j,t,\nu) = 0$. This is because if it is not optimal to transmit in any of the states, then only average cumulative discounted holding cost is incurred. Therefore, $\overline{V}^*_{i,\beta}(j,t;\nu) = \sum_{k=t}^{\infty} c_i(k)\beta^k$, for any $j \in \{1,2,\ldots,j_i\}$. By our assumption that $c_i(t) < \delta t^{\zeta}$, we get that $\sum_{k=t}^{\infty} c_i(k)\beta^k < \infty$.

3. We also know that $\overline{V}^*_{i,\beta}(j,t;\nu)$ is a continuous function of ν.

From the above observations and Intermediate Value Theorem, we can conclude that there exists at least a fixed point for (4.54), and we can find a supremum of the fixed points.

We know that $\lim_{\beta\to 1} \nu^*_{i,\beta}(j,r_{i,1},t) = \nu^*_i(j,r_{i,1},t) = \infty$. To the find the asymptote of $\nu^*_{i,\beta}(j,r_{i,1},t)$ as $\beta \to 1$, we can use (4.54), since $\nu^*_{i,\beta}(j,r_{i,1},t)$ is a fixed point of (4.54). To that end we will first study the characteristics of $\overline{V}^*_{i,\beta}(j,t;\nu)$ evaluated at $\nu = \nu^*_{i,\beta}(j,r_{i,1},t)$ as $\beta \to 1$, which we denote by $\overline{V}^*_{i,\beta}\left(j,t;\nu^*_{i,\beta}(j,r_{i,1},t)\right)$. We will show that the asymptote of $\overline{V}^*_{i,\beta}\left(j,t;\nu^*_{i,\beta}(j,r_{i,1},t)\right)$ is same as that of a policy in which transmissions are always performed when $r = r_{i,1}$ and never performed

otherwise. For any ν we can split the average cumulative cost into two, the average cumulative holding and transmission costs. Let $H^*_{i,\beta}(j,t,\nu)$ be the average cumulative holding cost under optimal policy starting from the phase j at time. Similarly, let the $N^*_{i,\beta}(j,t,\nu)$ be the cumulative discounted average number of transmissions under the optimal policy, i.e., $\mathbb{E}\left[\sum_{k=t}^{\infty}\beta^{k-t}A^*_{i,\beta}(k)\right]$, where $A^*_{i,\beta}(k)=1$ if the optimal decision is to transmit in slot k and 0 otherwise. Therefore, the average cumulative cost is given by:

$$\overline{V}^*_{i,\beta}(j,t;\nu) = H^*_{i,\beta}(j,t,\nu) + \nu N^*_{i,\beta}(j,t,\nu). \tag{4.55}$$

Similarly we can define $N^\dagger_{i,\beta}(j,t)$ and $H^\dagger_{i,\beta}(j,t)$ for the policy in which transmission are done only if $r = r_{i,1}$ for all j and t. Note that $N^\dagger_{i,\beta}(j,t)$ and $H^\dagger_{i,\beta}(j,t)$ are independent of ν as the policy is fixed and does not change with ν. The average cumulative cost associated with this policy is thus given by:

$$\overline{V}^\dagger_{i,\beta}(j,t;\nu) = H^\dagger_{i,\beta}(j,t) + \nu N^\dagger_{i,\beta}(j,t). \tag{4.56}$$

The main result connecting the optimal policy for $\mathcal{SP}_i(\nu)$ and the policy with transmissions only in $r_{i,1}$ is given next. Proof of this lemma is given in Sec. 4.13.1.

Lemma 4.12.1. *Let $\overline{V}^\dagger_{i,\beta}(j,t;\nu)$ be the average cumulative cost starting from j and t for the policy in which transmissions are performed only when the channel is in the best possible state. We have that*

$$\frac{\lim_{\beta\to 1} H^*_{i,\beta}\left(j,t,\nu^*_{i,\beta}(j,r_{i,1},t)\right)}{\lim_{\beta\to 1} H^\dagger_{i,\beta}(j,t)} = 1, \tag{4.57}$$

$$\frac{\lim_{\beta\to 1} N^*_{i,\beta}\left(j,t,\nu^*_{i,\beta}(j,r_{i,1},t)\right)}{\lim_{\beta\to 1} N^\dagger_{i,\beta}(j,t)} = 1. \tag{4.58}$$

The above lemma proves that $\lim_{\beta\to 1} \overline{V}^{*}_{i,\beta}\left(j,t;\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)$ and $\lim_{\beta\to 1} \overline{V}^{\dagger}_{i,\beta}\left(j,t;\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)$ have same asymptotes, and hence, we can use the latter to find the asymptote of $\overline{V}^{*}_{i,\beta}\left(j,t;\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)$. For $\overline{V}^{\dagger}_{i,\beta}\left(j,t;\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)$ we can find closed form expressions as we know the structure of the policy.

First we will find an expression for $\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)$. Substituting (4.55) in (4.54) and noting that $\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)$ is a fixed point for (4.54), we get the following expression for $\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)$:

$$\begin{aligned}&\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\\&\quad=\frac{\mu_i r_{i,1}\beta\left[H^{*}_{i,\beta}\left(j,t+1,\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)-H^{*}_{i,\beta}\left(j-1,t+1,\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)\right]}{1-\mu_i r_{i,1}\beta\left[N^{*}_{i,\beta}\left(j,t+1,\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)-N^{*}_{i,\beta}\left(j-1,t+1,\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)\right)\right]}.\end{aligned} \tag{4.59}$$

Next we multiply both sides of (4.59) with $1-\beta$ and take the limit $\beta\to 1$ on both the sides. Using Lemma 4.12.1, we can the replace the average cumulative costs related to the optimal policy with that of the policy in which transmissions are done only in $r=r_{i,1}$. Note that $N^{\dagger}_{i,\beta}\left(j,t\right)$ depends only on j and not on t. We have used this notation to maintain consistency. Further it can be shown that

$$\frac{\left(1-\beta\right)N^{\dagger}_{i,\beta}\left(j,t\right)}{q_{i,1}}=1-\mu_i r_{i,1}\beta\left(N^{\dagger}_{i,\beta}\left(j,t\right)-N^{\dagger}_{i,\beta}\left(j-1,t\right)\right). \tag{4.60}$$

Substituting (4.60) in (4.59), re-arranging the terms, and using the fact that $\lim_{\beta\to 1} N^{\dagger}_{i,\beta}\left(j,t\right)=\frac{j}{\mu_i r_{i,1}}$, we get that

$$\begin{aligned}&\xi^{*}_{i}\left(j,r_{i,1},t\right)=\\&\lim_{\beta\to 1}\left(1-\beta\right)\nu^{*}_{i,\beta}\left(j,r_{i,1},t\right)=\frac{q_{i,1}\left(\mu_i r_{i,1}\right)^2}{j}\left[H^{\dagger}_{i,1}\left(j,t+1\right)-H^{\dagger}_{i,1}\left(j-1,t+1\right)\right].\end{aligned} \tag{4.61}$$

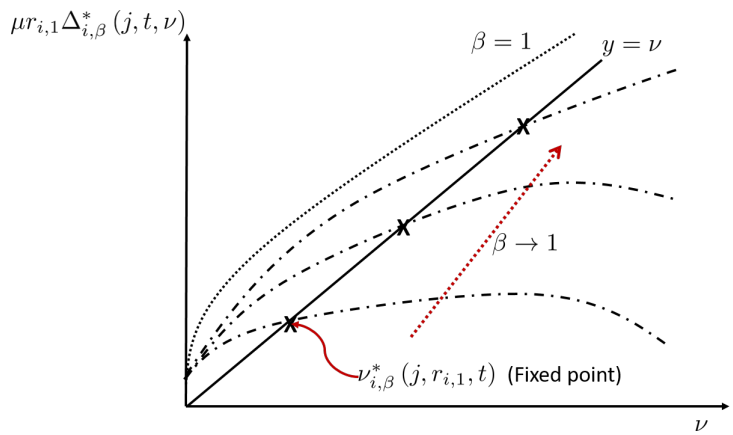


Figure 4.12: Increasing β while setting $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$ is illustrated here.

Due to Assumption 4.2.1 on $c_i(\cdot)$, it is bounded by a polynomial function of t. Therefore, the above expression is finite.

4.13 Proof of Auxiliary Lemmas: Secondary Index

4.13.1 Proof of Lemma 4.12.1

We have to the find the optimal policy when $\beta \to 1$ while we set $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$. This procedure is shown in the Fig. 4.12. In this proof, we shall show the following two properties of the optimal policy as $\beta \to 1$, while $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$:

1. It is not optimal to transmit in $r = r_{i,l}$ when $l \neq 1$ for any j and t.

2. It is always optimal to transmit in $r = r_{i,1}$ for j' and t' such that $(j', r_{i,1}, t')$ is reachable from $(j, r_{i,1}, t)$

First we have the following result. Proof of the following lemma is given in Appendix 4.13.2.

Lemma 4.13.1. *For a given ν, i, j, and t, $\Delta^*_{i,\beta}(j,t,\nu)$ is a non-decreasing function β.*

This would imply that $\nu^*_{i,\beta}(j, r_{i,l}, t)$ is a non-decreasing function of β. Hence, for any $\beta \in [0,1]$ and $l \neq 1$, we have

$$\nu^*_{i,\beta}(j, r_{i,l}, t) \leq \nu^*_i(j, r_{i,l}, t) < \infty. \tag{4.62}$$

From the indexability property, if the price $\nu > \nu^*_{i,\beta}(j, r_{i,l}, t)$, it is not optimal to transmit in $(j, r_{i,l}, t)$. Let us take the limit $\beta \to 1$ while $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$. We know as $\beta \to 1$, $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t) \to \infty$. We also know that as $\beta \to 1$, $\nu^*_{i,\beta}(j, r_{i,l}, t) < \infty$. This implies that for any j and t there exists some $\beta'(j, r_{i,l}, t)$ such that for $\beta > \beta'(j, r_{i,l}, t)$, it is optimal not to transmit in $(j, r_{i,l}, t)$.

Now we have to show that it is optimal to transmit in when $r = r_{i,1}$ in all states *reachable* from $(j, r_{i,1}, t)$. We say that a state is reachable from $(j, r_{i,1}, t)$ if there exists a policy π such that there is a strictly positive probability of making a transition into that state in the future. The reachable states from $(j, r_{i,1}, t)$ is shown in the Fig. 4.3. Note that the transition probabilities permit only transition into states where $t > t'$, $j' \leq j$, and if it is in the region shown in the figure. This is because we can get only at most one successful transmission in a slot. The following lemma will help us characterize the optimal policy when β is increased to 1, such that $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$. Proof of this lemma is given in the Appendix 4.13.3.

Lemma 4.13.2. *For large enough β, if it is optimal to transmit in $(j, r_{i,1}, t)$, then it is optimal to transmit in all states $(j', r_{i,1}, t')$ such that $(j', r_{i,1}, t')$ is reachable from $(j, r_{i,1}, t)$.*

The above lemma tells that if it is optimal to transmit when $r = r_{i,1}$ in any given time, then it is optimal to transmit in $r = r_{i,1}$ in all future times. If we choose $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$, we know that it is optimal to transmit in $(j, r_{i,1}, t)$. Hence, it is optimal to transmit in all states in the future where $r = r_{i,1}$. Therefore, as $\beta \to 1$ while $\nu = \nu^*_{i,\beta}(j, r_{i,1}, t)$, it is optimal to transmit when $r = r_{i,1}$ and not optimal to transmit when $r \neq r_{i,1}$. This completes the proof of this lemma.

4.13.2 Proof of Lemma 4.13.1

We will show that this property holds for any $c_i^{(k)}(\cdot)$ and hence, in the limiting case too due to lemma 4.9.1. We will first prove that $\Delta^{*,(k)}_{i,\beta}(j, t, \nu)$ is a non-decreasing function of β.

To prove the result for $c_i^{(k)}(\cdot)$, we will use induction over time which proceeds backwards from time k to t.

Base Case : We will first prove that $\Delta^{*,(k)}_{i,\beta}(j, t, \nu)$ is non-decreasing function of β for $t \geq k$. From the Bellman equations 4.7, we can re-write the value function as follows:

$$\overline{V}^{*,(k)}_{i,\beta}(j, t; \nu) = c_i^{(k)}(t) + \beta \overline{V}^{*,(k)}_{i,\beta}(j, t+1; \nu) + \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \beta \left[\Delta^{*,(k)}_{i,\beta}(j, t+1, \nu)\right]\right\}\right], \quad (4.63)$$

where R_i has the same distribution as $R_i(t)$. Using the above form of $\overline{V}^*_{i,\beta}(j, t; \nu)$,

we can re-write $\Delta_{i,\beta}^{*,(k)}(j,t,\nu)$ as follows:

$$\Delta_{i,\beta}^{*,(k)}(j,t,\nu) = \beta\Delta_{i,\beta}^{*,(k)}(j,t+1,\nu) + \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \beta \Delta_{i,\beta}^{*,(k)}(j,t+1,\nu)\right\}\right] - \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \beta \Delta_{i,\beta}^{*,(k)}(j-1,t+1,\nu)\right\}\right]. \quad (4.64)$$

We know that when the holding cost function $c_i^{(k)}(\cdot)$ has a constant value of $c_i(k)$ for $t \geq k$. Therefore, $\Delta_{i,\beta}^{*,(k)}(j,t,\nu) = \Delta_{i,\beta}^{*,(k)}(j,k,\nu)$ once $t \geq k$. Hence, substituting this in (4.64), we get that

$$(1-\beta)\,\Delta_{i,\beta}^{*,(k)}(j,k,\nu) - \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \beta \Delta_{i,\beta}^{*,(k)}(j,k,\nu)\right\}\right] = - \mathbb{E}\left[\min\left\{0, \nu - \mu_i R_i \beta \Delta_{i,\beta}^{*,(k)}(j-1,k,\nu)\right\}\right]. \quad (4.65)$$

Using the above equation, we can argue that $\Delta_{i,\beta}^{*,(k)}(j,k,\nu)$ is an non-decreasing function of β. This is done via induction over j. If $j = 1$, then $\Delta_{i,\beta}^{*,(k)}(j,k,\nu) = \overline{V}_{i,\beta}^{*,(k)}(1,k;\nu)$. $\overline{V}_{i,\beta}^{*,(k)}(1,k;\nu)$ is an non-decreasing function of β because for any policy π, the average cumulative cost to complete (average cumulative holding cost + transmission cost) is a non-decreasing function of β and therefore, $\overline{V}_{i,\beta}^{*,(k)}(1,k;\nu)$, which is obtained by computing infemum of the cost under all policies, is also a non-decreasing function of β. If we assume the induction hypothesis that $\Delta_{i,\beta}^{*,(k)}(j,k,\nu)$ is a non-decreasing function of β till $j-1$, then from (4.65), it can be easily shown that $\Delta_{i,\beta}^{*,(k)}(j,k,\nu)$ is a non-decreasing function of β. Hence we have proved that $\Delta_{i,\beta}^{*,(k)}(j,k,\nu)$ is a non-decreasing function of β.

Induction Hypothesis: Assume that $\Delta_{i,\beta}^{*,(k)}(j,t',\nu)$ is a non-decreasing of β for any j and $t' \geq t+1$.

We have to show that $\Delta_{i,\beta}^{*,(k)}(j,t,\nu)$ is a non-decreasing function of β. Consider (4.64). Its R.H.S. is a non-decreasing function of β because of our induction assumption that $\Delta_{i,\beta}^{*,(k)}(j-1,t+1,\nu)$ and $\Delta_{i,\beta}^{*,(k)}(j-1,t+1,\nu)$ are non-decreasing functions of β. Therefore, $\Delta_{i,\beta}^{*,(k)}(j,t,\nu)$ is also a non-decreasing function of β. Hence, we have proved that $\Delta_{i,\beta}^{*,(k)}(j,t,\nu)$ is a non-decreasing function of β when the holding cost function is $c_i^{(k)}(\cdot)$. Therefore, on taking the limit as $k \to \infty$, we get the result for $c_i(\cdot)$.

4.13.3 Proof of Lemma 4.13.2

We will show that for large enough β, if it is optimal to transmit in $(j, r_{i,1}, t)$, then it is optimal to transmit in the states $(j, r_{i,1}, t+1)$ and $(j-1, r_{i,1}, t+1)$. This is enough to show that it is optimal to transmit in all states reachable from $(j, r_{i,1}, t)$ because we can iteratively use this result on the states $(j, r_{i,1}, t+1)$ and $(j-1, r_{i,1}, t+1)$ and its neighboring states and so on. We will prove this result for any $c_i^{(k)}(\cdot)$.

We have already argued that for large enough β (say $\beta > \beta'$), it is optimal not to transmit in $r_{i,l}$ $l \neq 1$ in all states reachable from $(j, r_{i,1}, t)$ if the price is scaled such that $\nu = \nu_{i,\beta}^{*}(j, r_{i,1}, t)$. Let us assume that β is large enough that it is optimal not to transmit in $r_{i,1}$ for all states reachable from $(j, r_{i,1}, t)$. Note that if we transmit in $(j, r_{i,1}, t)$, then it must be optimal to transmit in either $(j, r_{i,1}, t+1)$ or $(j-1, r_{i,1}, t+1)$. Else, it is optimal not to transmit in $(j, r_{i,1}, t)$, and instead transmit in the state $(j, r_{i,1}, t+1)$ incurring only the discounted cost $\beta\nu$. Next we have to show that it is optimal to transmit in both $(j, r_{i,1}, t+1)$ and $(j-1, r_{i,1}, t+1)$.

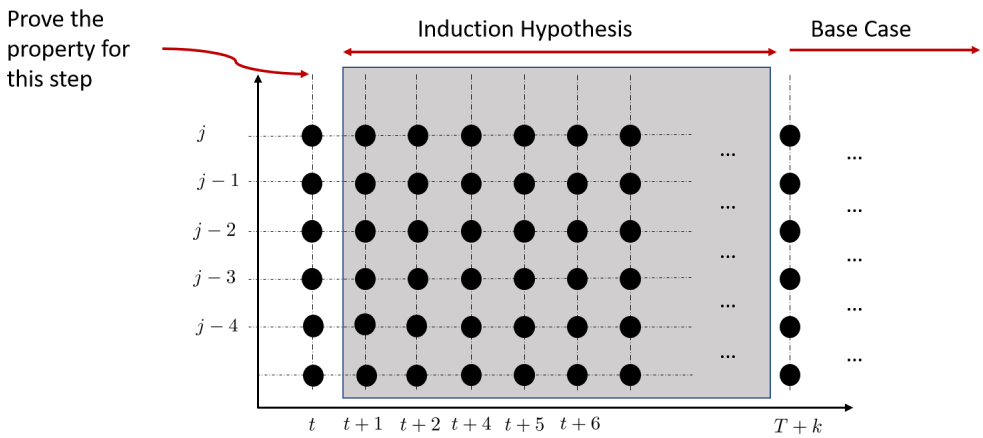


Figure 4.13: Illustration of the induction procedure

We will prove this as two separate cases. The induction process is illustrated in the Fig. 4.13.

Base Case: We have to prove that if it is optimal to transmit in the state $(j, r_{i,1}, k)$, then it is optimal to transmit in the states $(j, r_{i,1}, k+1)$ and $(j-1, r_{i,1}, t+1)$. If it is optimal to transmit in $(j, r_{i,1}, k)$, then from Bellman equations, we know the following:

$$\nu \leq \mu_i r_{i,1} \beta \Delta_{i,\beta}^{*,(k)} (j, k+1, \nu) . \tag{4.66}$$

We know that $\Delta_{i,\beta}^{*,(k)} (j, k+1, \nu) = \Delta_{i,\beta}^{*,(k)} (j, k+2, \nu)$ as the holding cost function has a constant value for $t \geq k$. Therefore, $\nu \leq \mu_i r_{i,1} \beta \Delta_{i,\beta}^{*,(k)} (j, k+2, \nu)$. From Bellman equations, this would imply that it is optimal to transmit in $(j, r_{i,1}, k+1)$. Hence, base case is proved.

Induction Hypothesis: We shall assume that if $t+1 \leq t' \leq k$ and if it is optimal to transmit in $(j, r_{i,1}, t')$, then it is optimal to transmit in $(j, r_{i,1}, t'+1)$ and $(j-1, r_{i,1}, t'+1)$.

Using the induction hypothesis we will have to show that if it is optimal to transmit in $(j, r_{i,1}, t)$, then it is optimal to transmit in $(j, r_{i,1}, t+1)$ and $(j-1, r_{i,1}, t+1)$. We will consider two separate cases:

1. If it is optimal to transmit in $(j, r_{i,1}, t)$ and $(j, r_{i,1}, t+1)$, then it is optimal to transmit in $(j-1, r_{i,1}, t+1)$.

2. If it is optimal to transmit in $(j, r_{i,1}, t)$ and $(j-1, r_{i,1}, t+1)$, then it is optimal to transmit in $(j, r_{i,1}, t+1)$.

We will prove the above two cases separately via proof by contradiction.

Case 1

Suppose it is optimal to transmit in $(j, r_{i,1}, t)$ and $(j, r_{i,1}, t+1)$, and it is not optimal to transmit in $(j-1, r_{i,1}, t+1)$. Let us also assume that $j \geq 2$. From our induction hypothesis and Bellman equations the following is true for $t+1 \leq t' \leq k$:

$$\overline{V}_{i,\beta}^{*,(k)}(j, t'; \nu) - \overline{V}_{i,\beta}^{*,(k)}(j-1, t'; \nu) \leq \overline{V}_{i,\beta}^{*,(k)}(j-1, t'+1; \nu) - \overline{V}_{i,\beta}^{*}(j-2, t'+1; \nu). \tag{4.67}$$

The above equation is true because of the induction hypothesis that if it is optimal to transmit in $(j, r_{i,1}, t')$, then it is optimal to transmit in $(j-1, r_{i,1}, t'+1)$. First observe that if it is optimal to transmit in $(j, r_{i,1}, t)$, then from Bellman equations we get the following:

$$\nu \leq \mu_i r_{i,1} \beta \left(\overline{V}_{i,\beta}^{*,(k)}(j, t+1; \nu) - \overline{V}_{i,\beta}^{*,(k)}(j-1, t+1; \nu) \right). \tag{4.68}$$

Since we have assumed that it is optimal to transmit in $(j, r_{i,1}, t+1)$, we have the following:

$$\overline{V}^{*,(k)}_{i,\beta}(j, t+1; \nu) = c_i^{(k)}(t+1) + q_{i,1}\nu + (1 - \mu_i q_{i,1} r_{i,1})\, \beta \overline{V}^{*,(k)}_{i,\beta}(j, t+2; \nu) + \mu_i q_{i,1} r_{i,1} \beta \overline{V}^{*,(k)}_{i,\beta}(j-1, t+2; \nu). \quad (4.69)$$

Similary, since it is not optimal to transmit in $(j-1, r_{i,1}, t+1)$, then we have that

$$\overline{V}^{*,(k)}_{i,\beta}(j-1, t+1; \nu) = c_i^{(k)}(t+1) + \beta \overline{V}^{*,(k)}_{i,\beta}(j-1, t+2; \nu). \quad (4.70)$$

Substituting (4.69) and (4.70) in (4.68), we get the following inequality:

$$\nu \leq \frac{\beta(1 - \mu_i q_{i,1} r_{i,1})}{1 - \mu_i q_{i,1} r_{i,1}} \left[\mu_i r_{i,1} \beta \left(\overline{V}^{*,(k)}_{i,\beta}(j, t+2; \nu) - \overline{V}^{*,(k)}_{i,\beta}(j-1, t+2; \nu)\right)\right]. \quad (4.71)$$

Now let us look at the state $(j-1, r_{i,1}, t+1)$. Since it is not optimal to transmit in this state, from Bellman equations, we will get the following inequality:

$$\nu > \mu_i r_{i,1} \beta \left(\overline{V}^{*,(k)}_{i,\beta}(j-1, t+2; \nu) - \overline{V}^{*,(k)}_{i,\beta}(j-2, t+2; \nu)\right). \quad (4.72)$$

We will expand the terms in the R.H.S. of the above inequality. From our induction hypothesis, the states in which it is optimal transmit is shown in the Fig. 4.14. This includes all states reachable from $(j, r_{i,1}, t+1)$. This would imply that it is optimal to transmit in $(j-1, r_{i,1}, t+2)$. This will give us the following equation:

$$\overline{V}^{*,(k)}_{i,\beta}(j-1, t+2; \nu) = c_i^{(k)}(t+1) + q_{i,1}\nu + (1 - \mu_i q_{i,1} r_{i,1})\, \beta \overline{V}^{*,(k)}_{i,\beta}(j-1, t+3; \nu) + \mu_i q_{i,1} r_{i,1} \beta \overline{V}^{*,(k)}_{i,\beta}(j-2, t+3; \nu) \quad (4.73)$$

Also, it has to be true that it is optimal not to transmit in $(j-2, r_{i,1}, t+2)$. This is because if it is optimal to transmit in both $(j, r_{i,1}, t+1)$ and $(j-2, r_{i,1}, t+2)$,

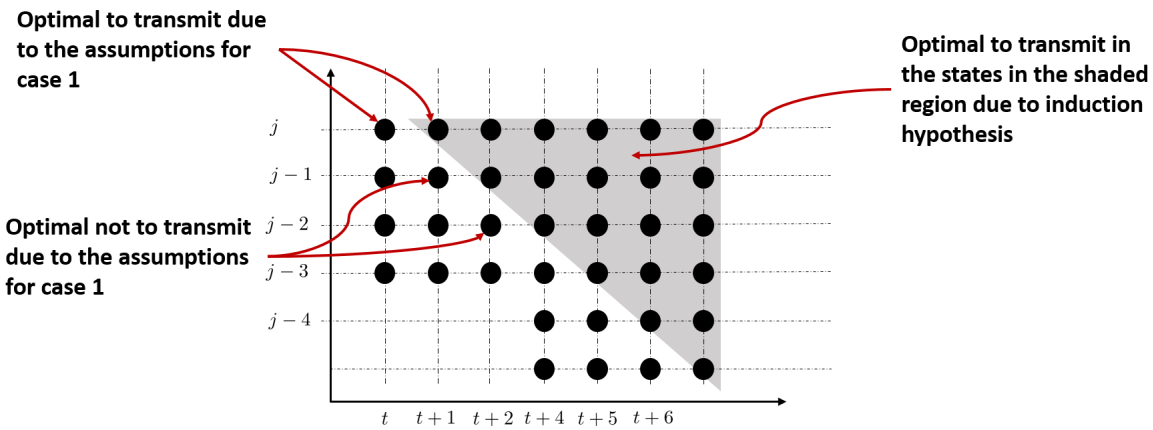


Figure 4.14: Illustration of the induction steps

then it must be optimal to transmit in $(j-1, r_{i,1}, t+1)$. This is obtained directly from the Bellman equations and our induction hypothesis. Therefore, we have the following equation:

$$\overline{V}_{i,\beta}^{*,(k)}(j-2, t+2; \nu) = c_i^{(k)}(t+1) + \beta \overline{V}_{i,\beta}^{*,(k)}(j-2, t+3; \nu) \tag{4.74}$$

Substituting (4.73) and (4.74) in (4.72), we will get the following:

$$\nu > \frac{\beta\left(1-\mu_i q_{i,1} r_{i,1}\right)}{1-\mu_i q_{i,1} r_{i,1}} \left[\mu_i r_{i,1} \beta \left(\overline{V}_{i,\beta}^{*,(k)}(j-1, t+3; \nu) - \overline{V}_{i,\beta}^{*,(k)}(j-2, t+3; \nu)\right)\right]. \tag{4.75}$$

Using (4.71) and (4.75), we will get the following inequality:

$$\overline{V}_{i,\beta}^{*,(k)}(j-1, t+3; \nu) - \overline{V}_{i,\beta}^{*,(k)}(j-2, t+3; \nu) < \overline{V}_{i,\beta}^{*,(k)}(j, t+2; \nu) - \overline{V}_{i,\beta}^{*,(k)}(j-1, t+2; \nu). \tag{4.76}$$

However, this cannot be true due to (4.67). Therefore, we have proved the result via contradiction.

Case 2

Let us assume that it is optimal to transmit in both $(j, r_{i,1}, t)$ and $(j-1, r_{i,1}, t+1)$ and not optimal to transmit in $(j, r_{i,1}, t+1)$. We will prove that this is not possible by contradiction.

From our induction hypothesis if it is optimal to transmit in $(j-1, r_{i,1}, t+1)$, then it is optimal to transmit in the states shown in the Fig. 4.14. This would imply that if it is optimal not to transmit in $(j, r_{i,1}, t+1)$, then it is optimal not to transmit in any $(j, r_{i,1}, t')\,, \forall t' \geq t+2$. This is because if it was true for some t'', then using the fact that it is also optimal to transmit in $(j-1, r_{i,1}, t'')$, we can iteratively show that it is optimal to transmit in $(j, r_{i,1}, t')\,,\ \forall t' \geq t+1$. Therefore, if the transmission does not succeed in $(j, r_{i,1}, t)$, then there are no future transmissions. To derive analytic expressions for this property, let us first define the following term:

$$\widehat{H}_\beta(t) := \sum_{m=0}^{\infty} c_i^{(k)}(t+m)\beta^m. \tag{4.77}$$

$\widehat{H}_\beta(t)$ is the average cumulative cost if no transmission is performed after time t. This summation is guaranteed to be finite because of our assumption that $c_i(t) < \delta t^\zeta$ for large t. Therefore, in this setting, from our previous discussion $\overline{V}_{i,\beta}^{*,(k)}\left(j, t+1; \nu\right) = \widehat{H}_\beta(t)$. Since we have assumed that it is optimal to transmit in $(j, r_{i,1}, t)$, from Bellman equations, we have the following inequality:

$$\nu \leq \mu_i r_{i,1} \beta \left(\widehat{H}_\beta(t+1) - \overline{V}_{i,\beta}^{*,(k)}\left(j-1, t+1; \nu\right) \right). \tag{4.78}$$

Since it is not optimal to transmit in $(j, r_{i,1}, t+1)$, we can similarly write the follow-

ing inequality:

$$\nu > \mu_i r_{i,1} \beta \left(\widehat{H}_\beta(t+2) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+2; \nu\right) \right). \tag{4.79}$$

Let us look at the term $\widehat{H}_\beta(t+1) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+1; \nu\right)$. We can re-write this term as follows:

$$\widehat{H}_\beta(t+1) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+1; \nu\right) = \mathbb{E}\left[\sum_{t'=t+2+T(j-1,t+1,j-1;\nu)}^{\infty} c_i^{(k)}(t')\beta^{t'} \right] - N^*_{i,\beta}\left(j-1, t+1, \nu\right) \tag{4.80}$$

The above equation is obtained by re-writing $\overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+1; \nu\right)$ as follows:

$$\overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+1; \nu\right) = \mathbb{E}\left[\sum_{t'=t+1}^{t+1+T(j-1,t+1,j-1;\nu)} c_i^{(k)}(t')\beta^{t'} \right] + N^*_{i,\beta}\left(j-1, t+1, \nu\right). \tag{4.81}$$

Similarly, we can re-write $\widehat{H}_\beta(t+2) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+2; \nu\right)$ as follows:

$$\widehat{H}_\beta(t+2) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+2; \nu\right) = \mathbb{E}\left[\sum_{t'=t+3+T(j-1,t+2,j-1;\nu)}^{\infty} c_i^{(k)}(t')\beta^{t'} \right] - N^*_{i,\beta}\left(j-1, t+2, \nu\right). \tag{4.82}$$

By our induction hypothesis and the fact that we are only transmitting when $r = r_{i,1}$, $T\left(j-1, t+1, j-1; \nu\right)$ and $T\left(j-1, t+2, j-1; \nu\right)$ are statistically identical. We also have that

$$N^*_{i,\beta}\left(j-1, t+1, \nu\right) = N^*_{i,\beta}\left(j-1, t+2, \nu\right). \tag{4.83}$$

Therefore, using the non-decreasing property of $c_i^{(k)}(t)$, we get the following inequality:

$$\widehat{H}_\beta(t+2) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+2; \nu\right) > \widehat{H}_\beta(t+1) - \overline{V}^{*,(k)}_{i,\beta}\left(j-1, t+1; \nu\right). \tag{4.84}$$

Therefore, a lower bound for ν is greater than its upper bound, which is a contradiction. Hence, proved.

4.14 Qualitative Results

4.14.1 Proof of Theorem 4.4.6

First we will prove the following lemma which is useful to prove this theorem.

Lemma 4.14.1. *If it is optimal to transmit in the state $(j, r_{i,2}, t)$, then it is optimal to transmit in any state $(j', r_{i,2}, t')$ such that $j' \geq j$ and $t' \geq t$.*

Proof. We will show that this holds for the cost function $c_i^{(k)}(\cdot)$. For this we will use induction starting from time k and proceeding backwards to t as shown in Fig. 4.13.

Base Case: Note that for $t \geq k$ the holding cost function is a constant. For constant holding cost functions, $\Delta_i^{*,(k)}(j, t, \nu)$ is independent of j, see Proposition 1 in [61]. Therefore if it is optimal to transmit in $(j, r_{i,2}, t)$, then it is optimal to transmit in $(j', r_{i,2}, t)$ such that $j' \geq j$.

Induction Hypothesis: If it is optimal to transmit in $(j, r_{i,2}, t')$ for any j and $t' \geq t+1$, then it is optimal to transmit in $(j', r_{i,2}, t'')$ for any $j' \geq j$ and $t'' \geq t'$.

Using the induction hypothesis, we will prove the result for any j at time t. First note that if it is optimal to transmit in $(j, r_{i,2}, t)$, then it is optimal to transmit in either $(j, r_{i,2}, t+1)$ or $(j-1, r_{i,2}, t+1)$. This can be proved using contradiction, i.e., we shall assume that it is optimal to transmit in $(j, r_{i,2}, t)$ and it is not optimal to transmit in $(j, r_{i,2}, t+1)$ and $(j-1, r_{i,2}, t+1)$. Now consider another policy in which we do not transmit in $(j, r_{i,2}, t)$ and we transmit in both $(j, r_{i,2}, t+1)$ and

$(j-1, r_{i,2}, t+1)$, while leaving the remaining actions unchanged with respect to an optimal policy. Starting with phase j at time t, the average cumulative cost with this policy is same as the average cumulative cost with the optimal policy, which is a contradiction as we had assumed that it is not optimal to transmit in $(j, r_{i,2}, t+1)$ and $(j-1, r_{i,2}, t+1)$. Therefore, it is optimal to transmit in either $(j, r_{i,2}, t+1)$ or $(j-1, r_{i,2}, t+1)$.

From induction hypothesis, if it is optimal to transmit in $(j, r_{i,2}, t+1)$ or $(j-1, r_{i,2}, t+1)$, it is also optimal to transmit in $(j, r_{i,2}, t+1)$ and $(j+1, r_{i,2}, t+1)$. If it is optimal to transmit in both $(j, r_{i,2}, t+1)$ and $(j+1, r_{i,2}, t+1)$, then it is optimal to transmit in $(j+1, r_{i,2}, t)$. To see this, let us re-write $\Delta_i^*(j+1, t, \nu)$ as follows:

$$\Delta_i^{*,(k)}(j+1, t, \nu) = \overline{V}_i^{*,(k)}(j+1, t+1; \nu) - \overline{V}_i^{*,(k)}(j, t+1; \nu), \tag{4.85}$$

$$= (1-\mu_i \overline{r}_i)\left(\Delta_i^{*,(k)}(j+1, t+1, \nu)\right) + \mu_i \overline{r}_i \left(\Delta_i^{*,(k)}(j, t+1, \nu)\right) \tag{4.86}$$

Note that in writing (4.86), we have used the fact that is optimal to transmit in $(j, r_{i,2}, t+1)$, $(j+1, r_{i,2}, t+1)$, $(j, r_{i,1}, t+1)$, and $(j+1, r_{i,1}, t+1)$. Since it is optimal to transmit in $(j, r_{i,2}, t+1)$ and $(j+1, r_{i,2}, t+1)$, from Bellman equations, we have that

$$\frac{\nu}{\mu_i r_{i,2}} \leq \Delta_i^{*,(k)}(j+1, t+1, \nu), \tag{4.87}$$

$$\frac{\nu}{\mu_i r_{i,2}} \leq \Delta_i^{*,(k)}(j, t+1, \nu). \tag{4.88}$$

From (4.86), this would imply that $\Delta_i^{*,(k)}(j+1, t, \nu) \geq \nu/\mu_i r_{i,2}$. This would mean that it is optimal to transmit in $(j+1, r_{i,2}, t)$. Since this was proved for any $c_i^{(k)}(\cdot)$,

from lemma 4.9.1 it holds for $c_i(t)$ too. $\square$

The above lemma would imply that $\nu_i^*(j, r_{i,2}, t) \leq \nu_i^*(j', r_{i,2}, t')$. Since j, j', t, and t' are arbitrarily chosen, this would imply that $\nu_i^*(j, r_{i,2}, t)$ is a non-decreasing function of j and t. To extend this result to the entire shaded region as shown in Fig. 4.2, from (4.64), one could show that if it is optimal to transmit in $(j, r_{i,2}, t+1)$ and $(j-1, r_{i,2}, t+1)$, then it is also optimal to transmit in $(j, r_{i,2}, t)$. If we use this property and the above lemma iteratively, then it can be shown that if it is optimal to transmit in $(j, r_{i,2}, t)$, then it is optimal to transmit in any state $(j', r_{i,1}, t')$ such that $j' \geq j$ and $j' + t' \geq j + t$.

4.14.2 Proof of Theorem 4.4.8

We have already proved in Lemma 4.13.2 that for large enough β if it is optimal to transmit in $(j, r_{i,1}, t)$, then it is optimal to transmit in all states reachable from $(j, r_{i,1}, t)$. This would also imply that it is optimal to transmit in all states $(j, r_{i,1}, t')$ such that $t' \geq t$. Hence, $\nu_{i,\beta}^*(j, r_{i,1}, t) \leq \nu_{i,\beta}^*(j, r_{i,1}, t')$. This would imply that

$$\lim_{\beta \to 1} (1-\beta)\, \nu_{i,\beta}^*(j, r_{i,1}, t) \leq \lim_{\beta \to 1} (1-\beta)\, \nu_{i,\beta}^*(j, r_{i,1}, t')\,. \tag{4.89}$$

Hence, the result is proved.

4.14.3 Proof of Theorem 4.4.3

We will show that this property holds for truncated holding cost functions $c_i^{(k)}(\cdot)$ and $c_l^{(k)}(\cdot)$. To prove this result for any k, we will use induction.

Base Case: By the definition of $c_i(t)$ and $c_l(t)$, we have that $c_i^{(k)}(t) \leq c_l^{(k)}(t)$. This would also imply that $c_i(k) \leq c_l(k)$. Using the result from [61] for constant holding costs, when the cost functions are $c_i^{(k)}(\cdot)$ and $c_l^{(k)}(\cdot)$, we get that $\Delta_i^{*,(k)}(j,k,\nu) \leq \Delta_l^{*,(k)}(j,k,\nu)$. Hence, base case is true.

Induction Hypothesis: Assume that $\Delta_i^{*,(k)}(j,t',\nu) \leq \Delta_l^{*,(k)}(j,t',\nu)$ for all $t+1 \leq t' \leq k$.

We will show that $\Delta_i^{*,(k)}(j,t,\nu) \leq \Delta_l^{*,(k)}(j,t,\nu)$. Note that from (4.64) (with $\beta = 1$), $\Delta_i^{*,(k)}(j,t,\nu)$ is an increasing function of $\Delta_i^{*,(k)}(j,t+1,\nu)$ and $\Delta_i^{*,(k)}(j-1,t+1,\nu)$. Then from our induction hypothesis it follows that $\Delta_i^{*,(k)}(j,t,\nu) \leq \Delta_l^{*,(k)}(j,t,\nu)$. Since we have proved it for truncated holding cost functions, from Lemma 4.9.1 it follows that the result holds for $c_i(\cdot)$ and $c_l(\cdot)$.

4.15 Quantitative Results

4.15.1 Proof of Theorem 4.5.1

This is a special case with $q_{i,1} = 1$ and $\mu_i = 1$. We have already proved that $\nu_i^*(j,r_i,t) = \infty, \forall t$ and j. In the proof of Theorem 4.4.2, we have given a constructive proof to study the asymptote of $\nu_{i,\beta}^*(j,r_i,t)$ (with respect to β) in which we have shown that the optimal policy and the policy in which transmission is done only in $r_{i,1}$ have the same asymptote when we set $\nu = \nu_{i,\beta}^*(j,r_i,t)$. In this setting, we have $\mu_i r_i = 1$, i.e., all transmissions are successful in completing a phase with probability one. Substituting this in (4.61), we get

$$\xi_i^*(j,r_{i,1},t) = \frac{1}{j} c_i(t+j). \tag{4.90}$$

Note that in writing the above equation, we have used the following expression for $H^{\dagger}_{i,1}(j,t)$, which was obtained because $\mu_i r_i = 1$:

$$H^{\dagger}_{i,1}(j,t) = \sum_{t'=t}^{t+j} c_i(t'). \tag{4.91}$$

4.15.2 Proof of Theorem 4.5.2

From Thm. 4.4.6, if it is optimal to transmit in $(1, r_{i,2}, t)$, then it is optimal to transmit in $(1, r_{i,2}, t')$ $\forall t' \geq t$. From Bellman equations, if it is optimal to transmit in $(1, r_{i,2}, t')$, then it is also optimal to transmit in $(1, r_{i,1}, t')$. Therefore, if it is optimal to transmit in $(1, r_{i,2}, t)$, then it is optimal to transmit in all states in future. To find $\nu^*_i(1, r_{i,1}, t)$, we have to solve the following equation in ν:

$$\nu = \mu_i r_{i,2} \overline{V}^*_i(1, t+1; \nu). \tag{4.92}$$

Since it is optimal to transmit in all future states we can re-write $\overline{V}^*_i(1, t+1; \nu)$ as follows:

$$\overline{V}^*_i(1, t+1; \nu) = \sum_{j=1}^{\infty} c_i(t+j)(1-\mu_i \overline{r}_i)^{j-1} + \frac{\nu}{\mu_i \overline{r}_i}. \tag{4.93}$$

Substituting in (4.92), we get the expression for $\nu^*_i(1, r_{i,1}, t)$.

4.15.3 Proof of Theorem 4.5.3

Primary indices: It is difficult to find an exact expression for the primary index when $R(t) \neq r_{i,1}$ in a multi-state i.i.d. service rate setting with phase-type distribution for jobs sizes. In any state (j, r, t) we know from the Bellman equations (4.7) that if it is optimal to transmit in $r = r_{i,l}$, then it is optimal to transmit

when $r = r_{i,l'}$, $l' = 1, 2, \ldots, l-1$. However, we do not know if it is optimal to transmit it when $R_i(t) = r_{i,l}$ for the future states.

We shall approximate $\nu_i^*(j, r_{i,l}, t)$ with a lower bound. Observe that if it is optimal to transmit in state $(1, r_{i,l}, t+j-1)$, then it is also optimal to transmit in state $(j, r_{i,l}, t)$. This directly follows from Thm. 4.4.6[2]. Therefore, $\nu_i^*(1, r_{i,l}, t+j-1)$ is a lower bound for $\nu_i^*(j, r_{i,l}, t)$.

Next we shall discuss computation of $\nu_i^*(1, r_{i,l}, t+j-1)$. For $j = 1$, we have that

$$\Delta_i^*(1, t, \nu) = \overline{V}_i^*(1, t; \nu) = H_i^*(1, t, \nu) + \nu N_{i,1}^*(1, t, \nu). \tag{4.94}$$

If it is optimal to transmit in $(1, r_{i,l}, t)$, then it is optimal to transmit when the rate is greater than or equal to $r_{i,l}$ in all future states from Lemma 4.14.1. However, we cannot say if it is optimal to transmit in future states with service rates strictly lower that $r_{i,l}$. Therefore, we shall find a lower bound for $\overline{V}_i^*(1, t+j-1; \nu)$ and use to find the fixed point of $\mu_i r_{i,l} \overline{V}_i^*(1, t+j-1; \nu)$. This fixed point using the lower bound of $\overline{V}_i^*(1, t+j-1; \nu)$ will be a lower bound for $\nu_i^*(1, r_{i,l}, t+j-1)$. First we shall derive a lower bound for $H_i^*(1, t, \nu)$. For any policy, the average holding cost is lower bounded by the cost under the policy in which transmission is always performed irrespective of the channel state. Therefore, we have that

$$H_i^*(1, t+j-1, \nu) \geq \sum_{m=0}^{\infty} c_i(t+j-1+m) \left(1 - \mu_i \sum_{n=1}^{L} q_{i,n} r_{i,n}\right)^m. \tag{4.95}$$

[2]One can easily extend the derivation for the two state i.i.d. channel to a multi-state i.i.d. channel setting and we state the result without proof.

Since we know that it is optimal to transmit when the rate is greater than or equal to $r_{i,l}$, we can lower bound the term $N^*_{i,1}(1,t,\nu)$ as follows:

$$N^*_{i,1}(1,t,\nu) \geq \frac{\sum_{l'=1}^{l} q_{i,l'}}{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}}. \tag{4.96}$$

Solving the following equation in ν gives the required expression:

$$\nu = \mu_i r_{i,l} \left(\sum_{m=0}^{\infty} c_i(t+j-1+m) \left(1 - \mu_i \sum_{n=1}^{L} q_{i,n} r_{i,n}\right)^m + \frac{\nu \sum_{l'=1}^{l} q_{i,l'}}{\mu_i \sum_{l'=1}^{l} q_{i,l'} r_{i,l'}} \right). \tag{4.97}$$

Secondary indices: We have computed the expression for secondary indices for i.i.d. multi-state channel in (4.61). This gives (4.21). If we transmit only when $R_i(t) = r_{i,1}$, then probability of completing a phase in any given slot is $q_{i,1}\mu_i r_{i,1}$. Using this and the definition of $H^{\dagger}_{i,1}(j,t)$ one could derive equations (4.22)–(4.25).

Chapter 5

Resource Allocation Strategies and HARQ Optimization for URLLC Traffic

5.1 Introduction

5G wireless networks are expected to support a new class of traffic called Ultra Reliable Low Latency Communication (URLLC) for appplications like industrial automation, mission critical traffic, virtual reality, etc., see e.g., [11–16]. URLLC traffic have stringent packet latency requirement of less than 1 msec along with very high reliability of 99.999 %. The design of wireless systems subject to such stringent requirements is a challenging task which is the focus of this chapter[1]. Specifically we consider downlink transmission of URLLC traffic in an Frequency Division Duplex (FDD) based system with separate frequency bands for uplink and downlink.

The Quality of Service (QoS) requirements URLLC traffic places on the Radio Resource Management (RRM) layer of the protocol stack are specified as follows: A packet of size L bits must be successfuly delivered to the receiver by the Base Station (BS) within a end-to-end delay of no more than d seconds with a probability of at least $1 - \delta$. The delay experienced by a packet includes queuing delay at

[1]Publications based on this chapter: Arjun Anand and Gustavo de Veciana, "Resource Allocation and HARQ Optimization for URLLC Traffic in 5G Networks", submitted to IEEE JSAC, arxiv version [70]

the BS, transmission duration, receiver processing delay, packet decoding feedback transmission duration, and time to make further re-transmissions as needed. Typical values of QoS parameters mentioned in the literature are $L = 50$ bytes, $d = 1$ msec, and $\delta = 10^{-6}$, see [16] for more details. This chapter investigates how design choices impact the the URLLC capacity, i.e., the maximum URLLC load the system can support and how this is affected by the stringency of the QoS requirements. In particular, the chapter addresses the following three inter-related questions:

1. How does resource allocation in the time-frequency plane of an Orthogonal Frequency Division Multiple Access (OFDMA) based system affect URLLC 'capacity'?

2. How does URLLC 'capacity' scale with L, d, δ and system bandwidth W?

3. What is the effect of Forward Error Correction (FEC) and Hybrid Automatic Repeat Request (HARQ) schemes on the URLLC 'capacity'?

The answers to the above questions are inter-related, for example, the scaling of URLLC capacity with system bandwidth W depends on the FEC scheme and HARQ schemes used.

A characterization of the impact of FEC and HARQ on URLLC capacity is important because one can then optimize the FEC and HARQ schemes to maximize the URLLC capacity. For example, one can optimize the appropriate number of re-transmissions and the target decoding failure probability after each stage. The maximum number of re-transmissions is constrained by the deadline d. Once the

target decoding failure probability is known for each stage one can choose the coding rate appropriately which in turn affects the capacity of the system. Since URLLC packets are generally small one cannot use the large blocklength Shannon capacity results to analyze the system. Thus we shall use the channel capacity results for finite blocklength regimes given in [1] for our analysis.

Another important aspect which needs careful consideration is how resources are allocated to URLLC transmissions. 5G standards are OFDMA based and hence, users' packets are allocated different parts of a time-frequency plane for data transmission. To send a URLLC packet, we can use a 'tall' transmission which uses a large bandwidth for a short duration or a 'wide' transmission, i.e., small bandwidth over a longer duration. If we use a 'tall' transmission, the number of concurrent transmissions possible will decrease which may affect the capacity for concurrent transmissions. However, a 'wide' transmission will take longer to complete and reduce the number of re-transmissions possible before the delay deadline expires. Hence, it may be desirable to implement a robust coding (with more redundancy bits) for 'wider' transmissions. This chapter proposes an analytical framework to capture and optimize trade-offs between 'tall' and 'wide' transmissions.

In addition, users may have different wireless channel conditions due to varying distances from the Base Station and fading. Therefore, different users may require different amounts of resources to reliably send L bits and hence, the system capacity depends on the geographical distribution of users in the cell. To summarize, wireless system design for URLLC traffic has to tackle the complex dependencies between L, d, δ, W, FEC and HARQ mechanisms, and the resource allocation to URLLC

transmissions.

5.1.1 Related Work

URLLC traffic has recently received a lot of attention. The 3GPP standards committee has recognized the need for a new OFDMA based frame structure to support URLLC traffic, which is different from that used for enhanced Mobile Broadband (eMBB) traffic, see [16] for a discussion of various proposals. In particular to meet the stringent latency constraints of URLLC traffic, they have proposed a *mini-slot* level access to radio resources for URLLC traffic with mini-slot durations of $0.125-2$ msec. This is different from the standard *slot* level access to radio resources for eMBB traffic where a slot has a duration of 1 msec or higher. The use of flexible traffic dependent slot durations and resource allocation has also been proposed in [14].

System level designs for URLLC networks have been explored in [17, 71–74]. In [74], the authors discuss information theoretic results on sending short packets. They also discuss protocols to transmit small length packets between two nodes, in a downlink broadcast setting and for random access based uplink. Their protocol for sending small packets reliably is related to our work, however, they do not focus on optimizing the resources required in an OFDMA based system supporting stochastic loads. In [71], the authors have covered various aspects of URLLC traffic like the overhead due to packet headers, decoding failure probability of URLLC transmissions, Channel State Information (CSI) acquisition at the transmitter. They have also proposed using interface diversity, grant-free access for uplink and device-to-device communication (D2D) as possible solutions to achieve the stringent latency

requirements of URLLC traffic. In [72], the authors discuss QoS requirements for URLLC traffic. They also specify various methods to share resources among URLLC and other types of traffic. In [73], the authors study the effect of physical layer waveforms, OFDMA numerology, and FEC schemes on the URLLC capacity via simulation. They have proposed the use of Tail Biting Convolution Codes (TBCC) to achieve a reliability of 10^{-9}.

The work in [17] is most closely related to ours. The authors have have used a queue based model and simulations to study the design of wireless systems for supporting URLLC traffic. In particular they introduce simple $M/M/m/k$ and $M/D/m/m$ queuing models to study trade-offs among system capacity, latency requirements and reliability for the worst case scenario where all users are at the cell edge. They have observed that decreasing the Round Trip Time (RTT) and Transmit Time Interval (TTI) increases the URLLC capacity. They have also considered trade-offs among system capacity, reliability, and latency requirements. However, in the analysis of system trade-offs, they have only considered packet loss due to blocking at the BS and have not explicitly considered the effect of decoding failures and re-transmissions on system capacity. Also, they do not consider the design of FEC and HARQ. Our work is inspired by this initial work.

The above mentioned work [17] also focussed on multiplexing of enhanced Mobile Broadband (eMBB) and URLLC traffic. They showed that allocating dedicated frequency bands to URLLC and eMBB traffic is inefficient, and have advocated a wide-band resource allocation for both URLLC and eMBB traffic. In addition to [17], there are few other works [18,38] which address multiplexing URLLC and eMBB traf-

fic. In [38], we have also considered the multiplexing of eMBB and URLLC traffic via puncturing/superposition of eMBB traffic and developed joint scheduling policies for eMBB and URLLC. We will explain this in detail in Chapter 6

Many works focus on the industrial applications of URLLC traffic and exhibit simulation based studies for such systems, see [11–13]. Some works, e.g., [75, 76] focus exclusively on physical layer aspects like modulation and coding, fading and link budget analysis. However, the above mentioned works do not holistically address the design of wireless systems supporting URLLC traffic.

5.1.2 Our Contributions

In this chapter we shall consider a simple Poisson model for URLLC packet arrivals. In line with the previous works, we shall also assume a wide-band allocation of resources to URLLC traffic by considering systems where such traffic can preemptively puncture/superpose URLLC packets upon previously scheduled eMBB traffic when necessary. We thus assume URLLC packets are scheduled immediately upon arrival. Such a model is reasonable due the stringent latency and reliability requirements of URLLC traffic. Based on this model this chapter makes the following key contributions.

1. *'Tall' vs 'Wide' transmission:* We first consider a *one-shot transmission* setting where URLLC packets are transmitted once and there are no further re-transmissions. We model the Base Station (BS) as a multi-class queuing system where each class of users corresponds to users with same quantizeed SINR. We show that extending URLLC transmissions in time (while reducing

the corresponding bandwidth usage) subject to deadline constraints increases the URLLC load that can be supported. Hence, 'wide' transmissions in time which take least amount of bandwidth to meet the delay deadline and reliability requirements are optimal.

2. *URLLC Capacity scaling:* Using an extension of the classical square-root staffing rule, we characterize the minimum overall system bandwidth to support a given URLLC load. Leveraging the capacity results in [1] under finite blocklength regime to study the scaling of URLLC capacity as a function of SINR, W, L, d and δ.

3. *Modeling and Performance Analysis:* We extend the one-shot transmission model to incorporate FEC and HARQ schemes which allow re-transmissions if needed. The entire downlink system, the BS and associated users are modeled as Jackson queuing network. We derive closed form expressions for various important parameters of the system like average packet delay, distribution of the number of packets in the system, average bandwidth utilization etc. Our framework can also quantify the effect of a given FEC and HARQ scheme on the URLLC capacity.

4. *Optimization of FEC and HARQ:* Finally we consider the optimization of HARQ and FEC schemes to maximize URLLC capacity. Instead of maximizing URLLC capacity to ease analysis we focus on the dual problem of minimizing the bandwidth required to support a given URLLC load. We consider two HARQ and FEC schemes, namely, *repetition coding with homogeneous trans-*

missions and *heterogeneous transmissions*. In both these schemes, a packet is re-transmitted and decoded independently of previous transmissions until it is successfully decoded or the maximum number of re-transmission attempts have been completed. In repetition coding with homogeneous transmissions, the same bandwidth and codeword are used for all re-transmissions. The performance under this scheme can be viewed as a lower bound on the *Chase combining*. In repetition coding with heterogeneous transmissions, the bandwidth and the codeword length are allowed to change across re-transmissions. These schemes provide a lower bound on the *Incremental Redundancy* (IR) HARQ schemes where joint decoding is performed based on all attempted transmissions. In the case of repetition coding with homogeneous transmissions, we find the following two observations:

a) At low loads, the required system bandwidth W is minimized when we use only one transmission with appropriate coding to meet the reliability requirement while spreading out the transmission in time as much as possible without violating the deadline d. This holds for a range of SINRs and packet sizes.

b) At high loads, the problem of the required system bandwidth W reduces to minimizing the mean bandwidth utilization. The maximum number of allowed re-transmissions under the optimal scheme is more than one and the block length required depends on L, SINR and d. In general at low SINRs the maximum number of re-transmissions required under the optimal scheme is more than the corresponding number at high SINR.

In the case of repetition coding with heterogeneous transmissions, we have the

following two findings:

a) Increasing the number of re-transmissions beyond two does not provide any additional benefit in terms of minimizing the bandwidth required to support URLLC traffic.

b) The optimal scheme has a first transmission with probability of failure of 10^{-2} and a second transmission with very high reliability to meet the required reliability constraint.

5.1.3 Organization

The chapter is organized as follows. In Sec. 5.2 we explain our one-short transmission model and the important results under this model. In Sec. 5.3 we extend the one-shot transmission model to incorporate of FEC and HARQ schemes. In Sec. 5.4 we discuss the optimization of FEC and HARQ schemes to maximize URLLC capacity followed by our conclusions in Sec. 5.5.

5.2 Performance Analysis: One-Shot Transmission

In this paper we focus on downlink transmissions in a wireless system with a single Base Station serving a dynamic population of URLLC users and their associated packets. The wireless system is OFDMA based where different parts of the time-frequency plane are allocated to URLLC users' packets based on transmission requests. A URLLC packet may suffer from queuing delays at the BS, transmission and propagation delays, and receiver processing delays. The system should be engineered such that the QoS requirements of URLLC traffic are satisfied, i.e., a URLLC

packet of size L bits must be delivered successfully to the receiver within a total delay of d seconds with a success probability of at least $1-\delta$. We start by introducing our system model.

5.2.1 System Model– One Shot Transmission

We consider a system operating in a large aggregate bandwidth of say W Hz[2]. For simplicity we ignore the slotted nature of the system. To model the 'near far' effects in wireless systems, we shall consider a multi-class system with C classes of users where each class represents users with same SINR[3]. The aggregate traffic generated at the BS by class c users is modeled as a Poisson process with rate λ_c packets/sec. Define the vector of arrival rates $\boldsymbol{\lambda} := (\lambda_1, \lambda_2, \ldots, \lambda_C)$. Let $SINR_c$ denote the SINR of a class c user's packets.

We initially assume that each URLLC packet is transmitted once. We will call this the *one-shot* transmission model. We will extend this to include re-transmissions in Sec. 5.3. A packet destined to a class c user requires r_c channel uses in the time-frequency plane to transmit its codeword. The codeword for a transmission is chosen such that the decoding is successful with probability of at least $1-\delta$. A URLLC packet of class c is allocated a bandwidth of h_c for a period of time s_c. These values are fixed and related to r_c by $\kappa s_c h_c = r_c$, where κ is a constant which denotes the number of channel uses per unit time per unit bandwidth of the OFDMA time-frequency

[2]This need not be a contiguous bandwidth, but result from the use of carrier aggregation across disjoint segments

[3]Ideally SINR is a continuous random variable, however, in practical systems the channel quality feedback from users are quantized to several discrete levels.

plane. The value of κ depends on the OFDMA frame structure and numerology. Since URLLC packets have a deadline of d seconds, we shall always choose $s_c \leq d$. For ease of analysis we shall also assume that for any class c, d is an integer multiple of s_c. Thus following vectors which characterize the system: $\mathbf{r} := (r_1, r_2, \ldots, r_C)$, $\mathbf{s} := (s_1, s_2, \ldots, s_C)$, $\mathbf{h} := (h_1, h_2, \ldots, h_C)$ and $\boldsymbol{\rho} := (\rho_1, \rho_2, \ldots, \rho_C)$, where $\rho_c := \lambda_c s_c$.

We shall make the following key assumption on the system operation.

Assumption 1. *(* **Immediate scheduling** *) A URLLC packet transmission request is scheduled immediately upon arrival if there is spare bandwidth is available. Otherwise the packet is lost. New packets do not preempt ongoing URLLC packet transmissions.*

Given the stringent latency requirements, the immediate scheduling assumption is a reasonable design choice.

5.2.2 Infinite System Bandwidth

Initially let us consider a system with infinite bandwidth, i.e., $W = \infty$. In such a system the base station can be modeled as a multi-class $M/GI/\infty$, see [77] for more details. Let $\mathbf{N} := (N_1, N_2, \ldots, N_C)$ be a random vector denoting the number of active transmissions when the system is in steady state. For any $\mathbf{n} \in \mathbb{Z}_+^C$, let $\pi(\mathbf{n}) := P(\mathbf{N} = \mathbf{n})$ be the stationary distribution. Using standard results for $M/GI/\infty$ queues (see [33]) one immediately gets the following results:

$$\pi(\mathbf{n}) = \Pi_{c=1}^{C} \left(\frac{\rho_c^{n_c}}{n_c!} \right) \exp(-\rho_c), \tag{5.1}$$

and the average bandwidth utilization is given by

$$\mathbb{E}\left[\mathbf{h}\mathbf{N}^T\right] = \mathbf{h}\boldsymbol{\rho}^T.$$

Observe that the number of active transmissions of any class c is Poisson distributed with mean ρ_c. Thus ρ_c as the average load of class c traffic.

5.2.3 Effect of Finite System Bandwidth

Although in practice the available system bandwidth W is not infinite but possibly large. We will consider a case where a wide-band allocation W is available to transmit URLLC traffic. This might be made available through a puncturing/superposition scheme between URLLC and eMBB traffic. see e.g., [38]. Even large bandwidth systems can occasionally suffer from congestion due to the stochastic variations in the arrival process and occasionally there may not be enough spare bandwidth to transmit a new URLLC packet. In such cases we shall assume that packets are *blocked* and dropped from the system. Let $\mathbf{N}(t) := (N_1(t), N_2(t), \ldots, N_C(t))$ be a random vector denoting the number of packets of each class in the system at time t. A class c packet arriving at time t is blocked if the following condition holds:

$$h_c + \sum_{c'=1}^{C} h_{c'} N_{c'}(t) > W. \tag{5.2}$$

We address the following two questions in this section:

1. How do the choices of $\mathbf{h}$ and $\mathbf{s}$ affect the blocking of URLLC packets?

2. What is the required system bandwidth W given a desired packet reliability δ?

To study the effect of $\mathbf{h}$ and $\mathbf{s}$ on the blocking of URLLC traffic, we shall first consider the blocking probability of a typical class c packet. Observe that the blocking probability experienced by packets of a class depends on $\mathbf{h}, \mathbf{s}$ (of all classes), $\boldsymbol{\lambda}$ and W. Let $p_{b,c}(\mathbf{h}, \mathbf{s}, \boldsymbol{\lambda}, W)$ be the blocking probability experienced by a typical class c packet arrival. The fraction of class c traffic admitted is then given by $\lambda_c(1 - p_{b,c}(\mathbf{h}, \mathbf{s}, \boldsymbol{\lambda}, W))$. Hence, lowering the blocking probability increases the admitted URLLC traffic. The following result which is proved in Appendix5.5.1 gives us the key insight on optimal choices of $\mathbf{h}$ and $\mathbf{s}$ for URLLC packet transmissions.

Theorem 5.2.1. *For a given* $\mathbf{h}$ *and* $\mathbf{s}$*, positive integer* q*, and* $i \in \{1, 2, \ldots, C\}$ *define* $\mathbf{h}' := (h_1, h_2, \ldots, h_i/q, \ldots, h_C)$ *and* $\mathbf{s}' := (s_1, s_2 \ldots, qs_i, \ldots, s_C)$*. Under the one-shot transmission model and Assumption 1, if* $\rho_i < 1$*, then for any* $c \in \{1, 2, \ldots, C\}$*, there exists* $\tilde{W}_c$ *such that for* $W > \tilde{W}_c$ *we have that* $p_{b,c}(\mathbf{h}, \mathbf{s}, \boldsymbol{\lambda}, W) \geq p_{b,c}(\mathbf{h}', \mathbf{s}', \boldsymbol{\lambda}, W)$*.*

Remarks: Observe that in wide-band systems scaling h_i and s_i by an integer q as required in the above theorem increases the number of concurrent transmissions of class i and is also beneficial for all classes (including class i). To understand this let us look at the mean and variance of the bandwidth utilization of class i in a system with parameters $\mathbf{h}'$ and $\mathbf{s}'$ and infinite bandwidth. The average bandwidth utilization of class i, given by $h_i \lambda_i s_i$, does not change with scaling factor q, as the decrease in bandwidth of class i is compensated by corresponding increase in the average number of users of class i. However, the variance of the bandwidth utilization, given by $\frac{1}{q} h_i^2 \lambda_i s_i$ decreases with q. Therefore, the congestion events occur less frequently and the system admits more traffic. Note that this observation is in line with the

previous work on URLLC traffic (see [**?**]) where the emphasis is on such events corresponding to the 'tail' of URLLC traffic demand. Further, the assumption $\rho_i < 1$ is not restrictive as one can divide a class into various 'virtual' sub-classes such that the average load in each sub-class is less than unity.

Therefore, one should scale s_i with an integer q such that $qs_i = d$. Such an integer q exists because of our assumption that d is an integer multiple of s_i. Hence, this motivates the following optimal choices of s_i and h_i:

$$s_i = d \quad \text{and} \quad h_i = \frac{r_i}{\kappa d}. \tag{5.3}$$

To summarize, one might think that 'tall' transmissions are better as they take less time, however, according to the above result it is better to decrease the bandwidth per transmission and spread out the transmissions as 'wide' as possible in the time axis, i.e., increase s_i (and decrease h_i) as long as the deadline is not violated.

To meet the reliability requirements of URLLC traffic, the system bandwidth W must be chosen such that the probability of blocking of a typical URLLC packet arrival is of the order of δ. To that end we shall use a multi-class extension of the classical square-root staffing rule (see [33] for more details) to relate W, $\mathbf{r}$, $\boldsymbol{\lambda}$ and δ. Under this dimensioning rule, to support a URLLC load of $\boldsymbol{\lambda}$ with reliability δ for a given $\mathbf{r}$, the system bandwidth should satisfy the following condition:

$$W \geq \zeta^{\text{mean}}(\mathbf{r}) + c(\delta)\sqrt{\zeta^{\text{variance}}(\mathbf{r})}, \tag{5.4}$$

where $c(\delta) = Q^{-1}(\delta)$, $Q(\cdot)$ is the Q-function, $\zeta^{\text{mean}}(\mathbf{r}) := \sum_{c=1}^{C} \lambda_c \frac{r_c}{\kappa}$ is the mean bandwidth utilization, and $\zeta^{\text{variance}}(\mathbf{r}) := \sum_{c=1}^{C} \lambda_c \frac{r_c^2}{\kappa^2 d}$ is the variance of the bandwidth utilization.

Next we study the URLLC capacity scaling with respect to W, $SINR_c$, d, and δ. This requires a model relating r_c, $SINR_c$, and δ which is described in the next subsection.

5.2.4 Finite Block Length Model

Since the URLLC packet sizes are typically small, we shall use the capacity results for the finite blocklength regime developed in [1]. In an AWGN channel the number of information bits L that can be transmitted with a codeword decoding error probability of p in r channel uses is given by

$$L = rC(SINR_c) - Q^{-1}(p)\sqrt{rV(SINR_c)} + 0.5\log_2(r) + o(1), \tag{5.5}$$

where $C(SINR_c) = \log_2(1 + SINR_c)$ is the AWGN channel capacity under infinite blocklength assumption and $V(SINR_c) = (\log_2(e))^2\left(1 - \frac{1}{(1+SINR_c)^2}\right)$. Using the above model one can approximate r as a function of p as follows:

$$r \approx \frac{L}{C(SINR_c)} + \frac{(Q^{-1}(p))^2 V(SINR_c)}{2(C(SINR_c))^2} + \frac{(Q^{-1}(p))^2 V(SINR_c)}{2(C(SINR_c))^2}\sqrt{1 + \frac{4LC(SINR_c)}{V(SINR_c)(Q^{-1}(p))^2}}. \tag{5.6}$$

A derivation of this approximation is given in Appendix5.6. We can now write r_c as a function of δ, L and $SINR_c$ for various user/packet classes.

5.2.5 Capacity Scaling

We shall define the *single class URLLC capacity* as follows.

Definition For any class c, its single class URLLC capacity λ_c^* is the maximum URLLC arrival rate that can be supported by the system while satisfying the QoS requirements if only class c traffic is present in the system.

Note that λ_c^* is a function of W, d. δ, $SINR_c$, and L. We would like to study the scaling of λ_c^* with respect to various system parameters. Recall that for $f, g : \mathbb{R}_+ \to \mathbb{R}_+$, we say that $f(x) \sim \Theta\left(g(x)\right)$ if there exist x_o, a, and b such that $a \leq b$ and for $x \geq x_o$ we have that

$$ag(x) \leq f(x) \leq bg(x). \tag{5.7}$$

The following result summarizes the scaling of λ_c^* with various system parameters. The proof of the theorem below is given in Appendix5.7.

Theorem 5.2.2. *Under one-shot transmission model and Assumption 1 we have that*

1. $\lambda_c^* \sim \Theta\left(W - \sqrt{W}\right)$.
2. *For* $SINR_c \gg 1$, *we have that* $\lambda_c^* \sim \Theta\left(\log_2\left(SINR_c\right) - \sqrt{\log_2\left(SINR_c\right)}\right)$.
3. $\lambda_c^* \sim \Theta\left(1 - \frac{1}{\sqrt{d}}\right)$.
4. $\lambda_c^* \sim \Theta\left(\frac{1}{-\log_2(\delta)}\right)$.

Remarks: Observe that λ_c^* scales as a strictly concave function of $SINR_c$, d, and δ. Hence, while increasing $SINR_c$ and d or decreasing δ one suffers from diminishing returns. However, as expected the scaling of λ_c with respect to W does

not suffer from diminishing returns. For large W, λ_c^* increases linearly with W which is the best one could hope.

5.3 Performance Analysis with Multiple Transmissions

In this section we shall extend the system model to include re-transmissions of a packet so that the effects of HARQ and FEC schemes are captured. We shall first explain the extension of our system model.

5.3.1 System Model– Multiple Transmissions

Similar to the one-shot transmission model in Sec. 5.2, we shall consider a multi-class system with Poisson arrivals for URLLC traffic, where a class represents users with same SINR. However, as opposed to a one-shot transmission model, in this section we shall permit packet re-transmissions. Suppose a class c packet can have up to m_c transmission attempts after which it is dropped. We index transmission attempts by $m = 1, 2, \ldots, m_c$, where $m = 1$ corresponds to the *initial* transmission and any $m > 1$ corresponds to a *re-transmission*. A class c packet in the mth transmission attempt is assumed to require $r_{c,m}$ resources in the time-frequency plane. The bandwidth used in m^{th} transmission $h_{c,m}$, and the time taken to transmit $s_{c,m}$, are related to $r_{c,m}$ by $h_{c,m}s_{c,m} = r_{c,m}$. For any $m \in \{1, 2, \ldots, m_c\}$, define $\mathbf{r}_c^{(m)} := (r_{c,1}, r_{c,2}, \ldots, r_{c,m})$. After every transmission the intended receiver sends a one bit feedback to the BS indicating success/failure of the packet decoding process. In general, the probability of decoding failure of a class c packet after m^{th} transmission attempt, denoted by $p_{c,m}\left(\mathbf{r}_c^{(m)}\right)$, is a function of $\mathbf{r}_c^{(m)}$. A decoding failure for a class c

packet occurs if the packet has not been successfully after m_c transmission attempts. It happens with probability $\Pi_{m=1}^{m_c} p_{c,m}\left(\mathbf{r}_c^{(m)}\right)$. Thus one would design system such that $\Pi_{m=1}^{m_c} p_{c,m}\left(\mathbf{r}_c^{(m)}\right) \leq \delta$. Therefore, the values of $r_{c,m}$, $p_{c,m}\left(\mathbf{r}_c^{(m)}\right)$ and m_c jointly characterize the FEC and HARQ scheme used for class c users.

The feedback on success/failure of a transmission will incur propagation delays, receiver processing delay, and the uplink channel access and scheduling delays. We shall assume that the uplink channel is well provisioned so that there are no scheduling and channel access delays. Therefore, the total feedback delay includes only the propagation delay and the receiver processing delay which we shall denote by a deterministic value f_c for a class c user. A class dependent feedback delay is consistent with our notion that classes denote users with similar channel characteristics, for example, users at the cell edge suffer from longer feedback delays.

For any class c, define the following vectors:

$$\mathbf{s}_c := (s_{c,1}, s_{c,2} \ldots, s_{c,m_c}), \ \mathbf{h}_c := (h_{c,1}, h_{c,2} \ldots, h_{c,m_c}), \text{ and } \boldsymbol{\rho}_c := (\rho_{c,1}, \rho_{c,2} \ldots, \rho_{c,m_c}), \tag{5.8}$$

where $\rho_{c,1} := \lambda_c s_{c,1}$ and for any $m > 1$, $\rho_{c,m} := \lambda_c \left(\Pi_{k=1}^{m-1} p_{c,k}\left(\mathbf{r}_c^{(k)}\right)\right) s_{c,m}$. Using the above definitions, we further define the following vectors capturing the overall system's designs and loads.

$$\mathbf{s} := (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_C), \ \mathbf{h} := (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_C), \text{ and } \boldsymbol{\rho} := (\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \ldots, \boldsymbol{\rho}_C). \tag{5.9}$$

We further let $\mathbf{m} := (m_1, m_2, \ldots, m_C)$ denote vector of maximum transmission attempts per class.

Next we shall also revise the *immediate scheduling* assumption for the setting with re-transmissions.

Assumption 2. *(***Immediate scheduling** *) An URLLC packet transmission request, initial or a re-transmission, is admitted and scheduled for transmission immediately if there is spare bandwidth available to transmit it without preempting ongoing URLLC transmissions. Otherwise the packet is lost.*

5.3.2 Infinite Bandwidth System

Once again let us initially assume that the system bandwidth W is infinite and there is no blocking of packets. In the multiple transmission model the BS has to wait for the feedback from the intended receiver before re-transmitting a packet. We model this system with feedback using a network of two multi-class $M/GI/\infty$ queues, one modeling BS transmissions and the other modeling the packets awaiting feedback, which we refer to as the *feedback* queue. This is discussed in detail below.

Base Station queue: Similar to Sec. 5.2, the BS is modeled as a multi-class $M/GI/\infty$ queue where each class corresponds to a set of users with the same SINR. However, unlike the one-shot transmission model, we further divide each class into various *sub-classes* to keep track of the re-transmissions. In particular each class c is further divided into m_c sub-classes with the sub-classes indexed by various stages of packet (re)transmission. A class c packet which is being transmitted for the mth time belongs to m^{th} sub-class and it will require $h_{c,m}$ bandwidth and take $s_{c,m}$ time to complete transmission. Further, because of our assumption of infinite bandwidth,

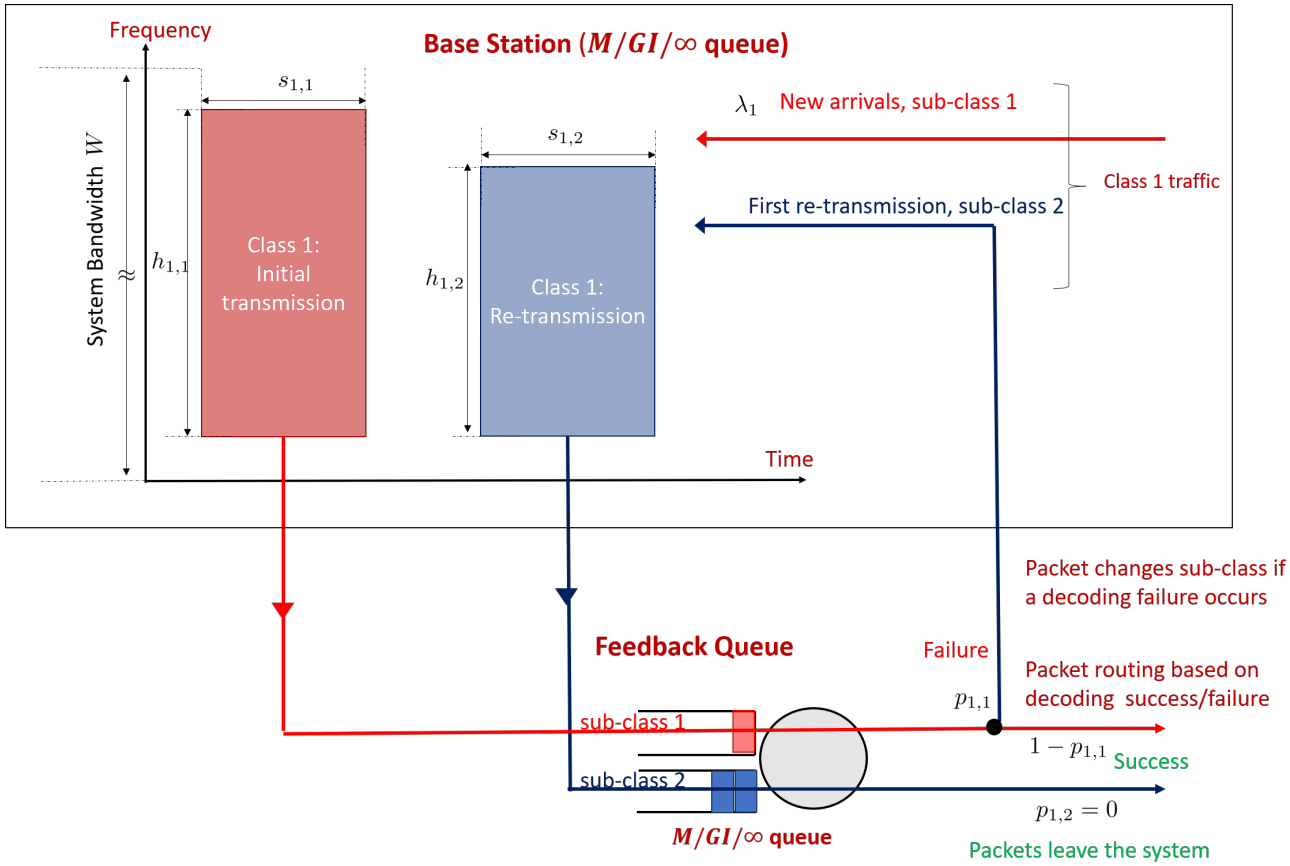


Figure 5.1: A wireless system with a single class of URLLC users modeled as a network of two $M/GI/\infty$ queues. Up to two transmissions attempts are allowed for all packets, i.e., $m_1 = 2$. Packets of sub-classes one and two are shown by red and blue colors, respectively. Observe that a packet will change its sub-class after a decoding failure.

the BS can transmit any number of packets from any of classes concurrently, i.e., the number of servers in the queuing model is ∞.

Feedback queue: We model the packet decoding and feedback sending processes via a multi-class $M/GI/\infty$ queue which uses the same notion of a class and sub-class in the feedback queue as in the BS queue. For a class c packet, the feedback associated with the decoding of a class c packet is received at the BS after f_c seconds. Based on the success/failure of the decoding process the BS then decides to re-transmit it or not. We abstract this process as follows. A class c packet after its mth transmission is routed from the BS queue to the feedback queue where it spends f_c seconds in the feedback queue. Note that the packet retains its class and sub-class indices in the feedback queue. After f_c seconds in the feedback queue it is then routed to the sub-class $m+1$ of class c with probability $p_{c,m}\left(\mathbf{r}_c^{(m)}\right)$ (decoding failure) or leaves the system with probability $1 - p_{c,m}\left(\mathbf{r}_c^{(m)}\right)$ (successful decoding). If a class c packet in m^{th} sub-class is routed to the BS, then it changes its sub-class index to $m+1$, i.e., it is being transmitted for $(m+1)^{\text{th}}$ time. This process repeats until the packet is successfully decoded, or m_c transmission attempts are made, whichever happens first. Thus a class c packet always leaves the system after m_c transmissions irrespective of the outcome of the decoding process of the m_c^{th} transmission. A queuing network consisting of a BS and a single class of URLLC traffic is illustrated in the Fig. 5.1.

Observe that it is assumed that any number of URLLC packets can be processed in parallel in the feedback queue, and hence it can also be modeled as an $M/GI/\infty$ queue. This is a reasonable assumption because the packet decoding pro-

cess across users are independent of each other and done in parallel and we assume sufficient uplink bandwidth is provisioned for feedback from various users.

The queuing model described previously can be used to study various important properties of the multi-class system which are given below. Let $\mathbf{N}$ be a random vector denoting the number of packets in different stages of re-transmissions of all classes in the steady state, i.e.,
$\mathbf{N} := (N_{1,1}, N_{1,2} \ldots, N_{1,m_1}, \ldots N_{c,1}, N_{c,2} \ldots, N_{c,m_c}, \ldots, N_{C,1}, N_{C,2} \ldots, N_{C,m_C})$. The steady state value probability distribution denoted by $\pi(\mathbf{n})$ is given by:

$$\pi\left(\mathbf{n}\right) = \Pi_{c=1}^{C} \Pi_{m=1}^{m_c} \left(\frac{\rho_{c,m}^{n_{c,m}}}{n_{c,m}!} \right) \exp\left(-\rho_{c,m}\right), \tag{5.10}$$

where $\rho_{c,m}$ is the average system load of class c packets in sub-class m. The average packet transmission delay for class c packets, denoted by τ_c, is given by the following expression

$$\tau_c = \frac{1}{\lambda_c} \sum_{m=1}^{m_c} \rho_{c,m} + f_c \left(1 + \sum_{m=1}^{m_c} \Pi_{j=1}^{m} p_{c,j} \left(\mathbf{r}_c^{(j)} \right) \right), \tag{5.11}$$

and the average bandwidth utilized is given by $\mathbb{E}\left[\mathbf{h}^T \mathbf{N}\right] = \mathbf{h}^T \boldsymbol{\rho}$.

5.3.3 Effect of Finite System Bandwidth

Similar to the case of one-shot transmission, a finite bandwidth system may suffer from congestion due to stochastic variations in loads and may have to block an immediate packet transmission request (a new packet or a re-transmission). Hence, we have to choose W appropriately to meet the reliability requirements. A natural extension to the result in (5.4) for the bandwidth requirement of the multi-class

system is as follows. Given a target blocking probability of δ, W is chosen such that

$$W \geq \eta^{\text{mean}} + c(\delta)\sqrt{\eta^{\text{variance}}}, \tag{5.12}$$

where

$$\eta^{\text{mean}} := \sum_{c=1}^{C} \lambda_c \left(r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k} \left(\mathbf{r}_c^{(k)} \right) \right) r_{c,m} \right),$$

$$\eta^{\text{variance}} := \sum_{c=1}^{C} \lambda_c \left(h_{c,1} r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k} \left(\mathbf{r}_c^{(k)} \right) \right) h_{c,m} r_{c,m} \right).$$

This directly follows by applying the square-root staffing rule to multi-class systems. The first term η^{mean} again represents the mean bandwidth utilization. The term η^{variance} represents the variance of the bandwidth utilization. Observe that while η^{mean} only depends on $\mathbf{r}$, each term in η^{variance} is multiplied with $h_{c,m}$ and thus is affected by the selected transmission modes across re-transmissions.

For $m_c = 1$, we have shown in Thm. 5.2.2 that it is advantageous in terms of blocking probability to decrease $h_{c,m}$ (or increase $s_{c,m}$) subject to the deadline constraint. Proof of the above result is not easily extendable to the case for $m_c > 1$. However, this result gives us a key insight on the choice of $h_{c,m}$. A natural extension of this insight to higher values of m_c is to increase the transmission times of all stages such that the cumulative transmission time of m_c stages and feedback delays add up to d, i.e.,

$$\sum_{m=1}^{m_c} s_{c,m} + m_c f_c = d. \tag{5.13}$$

Based on the previous discussion we shall list the various steps by which one can dimension a multi-class system appropriately to support URLLC traffic.

1. Choose $\mathbf{r}$ and $\mathbf{m}$ such that probability of decoding failure is less than or equal to δ.

2. Choose $\mathbf{s}$ such that the condition (5.13) is satisfied. This also determines $\mathbf{h}$ as $\mathbf{r}$ is chosen in the first step and $h_{c,m}s_{c,m} = r_{c,m}$.

3. To support any arrival rate vector $\boldsymbol{\lambda}$, determine the minimum necessary bandwidth using (5.12).

Even though (5.12) and (5.13) give us insights into the effect of re-transmissions on the URLLC capacity, however there are many possible solutions which satisfy (5.12) and (5.13). One has to find the optimal values for $\mathbf{r}$, $\mathbf{h}$, $\mathbf{s}$, and $\mathbf{m}$ to maximize the URLLC capacity. This is discussed in the next section.

5.4 URLLC Capacity Maximization/ Required Bandwidth Minimization

There are two ways to formulate the problem of optimizing FEC and HARQ schemes to maximize URLLC capacity. One can characterize the set of URLLC arrival rates which can be supported for a given bandwidth subject to the QoS constraints. This is will define a *multi-class URLLC capacity region*. Alternatively, one can formulate the problem in terms of minimizing the bandwidth required to support a given set of URLLC arrival rates subject to the QoS constraints. This second approach is somewhat simpler yet still allows one to study the most efficient system design choices given appropriate models for the FEC and HARQ schemes. One can then study the structural properties of the solution obtained. We shall

follow this second approach in the rest of this chapter. The associated optimization problem is as follows:

$$\mathcal{OP}_2: \quad \min_{\mathbf{m},\mathbf{r},\mathbf{h},\mathbf{s}}: \sum_{c=1}^{C} \lambda_c \big(r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k} \left(\mathbf{r}_c^{(k)} \right) r_{c,m} \right) \big) \tag{5.14}$$

$$+ c(\delta) \sqrt{\sum_{c=1}^{C} \lambda_c \left(h_{c,1} r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k} \left(\mathbf{r}_c^{(k)} \right) h_{c,m} r_{c,m} \right) \right)}, \tag{5.15}$$

$$\text{s.t.} \quad h_{c,m} s_{c,m} = r_{c,m}, \sum_{m=1}^{m_c} s_{c,m} + m_c f_c \leq d, \; h_c \leq W, \quad \forall c, \tag{5.16}$$

$$\Pi_{m=1}^{m_c} p_{c,m} \left(\mathbf{r}_c^{(m)} \right) \leq \delta. \tag{5.17}$$

The above problem is a non-convex, mixed integer programming problem, and in general is analytically intractable. To get some insights on this problem we will consider two specific schemes, namely, *repetition coding with homogeneous transmissions* and *repetition coding with heterogeneous transmissions*. The performance under these two schemes provide lower bounds on the performance under two commonly used schemes, namely, Chase combining and Incremental Redundancy schemes.

5.4.1 Repetition Coding– Homogeneous Transmissions

In repetition coding, the same codeword is transmitted repeatedly to the receiver until the packet is successfully decoded or the maximum number of retransmissions has been reached. We shall also further assume that the transmissions are homogeneous. This is stated formally below.

Assumption 3. *(***Homogeneous transmissions***) For all c and m, we have that $r_{c,m} = r_c$, $h_{c,m} = h_c$ and $s_{c,m} = s_c$.*

We also make the following assumption on the packet decoding process at the receiver.

Assumption 4. *(***Independent decoding***) The receiver decodes each transmission independently of the previous transmissions, and hence, the probability of failure in any transmission attempt depends only on the codeword used in that stage.*

Under the above assumptions, the decoding failure probability is independent across re-transmissions and driven by the resource required r_c, i.e., for any c and m we have that $p_{c,m}\left(\mathbf{r}_c^{(m)}\right) = p_c\left(r_c\right)$. Assuming independence between the decoding processes simplifies the analysis further. Also, due to the stringent latency requirements, complex HARQ schemes may not be practically feasible at the receiver. Using homogeneous transmissions reduces the overhead in control signals to indicate the allocation of bandwidth to users.

Unfortunately, under finite block length model and repetition coding, $\mathcal{OP}_2$ is still analytically intractable in a multi-class system. Therefore, we shall consider two regimes, *the variance dominated regime* and *the mean utilization dominated regimes* where the solutions simplify considerably. They are formally described next.

Definition

1. *Variance dominated regime:* In the variance dominated regime, the objective function includes only the variance of the bandwidth utilization (η^{variance}).

2. *Mean utilization dominated regime:* In mean utilization driven regime, the objective function includes only the mean of the bandwidth utilization (η^{mean}).

At low loads when λ_c's are small, in (5.15) the term corresponding to the overall variance is dominant, therefore, at low loads we shall minimize the variance of the total bandwidth usage. At high loads, the variance of bandwidth usage (second term) is smaller than the mean (first term). Hence, we shall focus on minimizing the mean utilization at high loads. We shall also use the finite blocklength model discussed in Sec. 5.2.4 to relate $p_c(r_c)$ and r_c. Under these simplifications, one can de-couple $\mathcal{OP}_2$ for each class and optimize the HARQ schemes separately for each class. The main result in the variance dominated regime is given below.

Proposition 5.4.1. *For the multiple transmissions model in Sec. 5.3, under Assumptions 2, 3, and 4, and in the variance dominated regime, the optimization problem $\mathcal{OP}_2$ decomposes across classes to per class optimization for class c as follows:*

$$\min_{m_c, r_c, h_c, s_c} : \sum_{m=0}^{m_c} h_c r_c (p_c(r_c))^m \tag{5.18}$$

$$s.t. \quad s_c h_c = r_c,\ h_c \leq W,\ m_c\,(s_c + f_c) = d, \tag{5.19}$$

$$(p_c(r_c))^{m_c} \leq \delta. \tag{5.20}$$

Furthermore, under the finite block length model (5.5) *relating $p_c(r_c)$ and r_c, for $L \leq 2000$ bits, $d \leq 2$ msec, $\delta \in [10^{-3}, 10^{-7}]$, $SINR_c \in [0, 20]\,dB$ the optimal solution has the following structure:*

1. *One shot transmission is optimal, i.e., the optimal value of m_c is one.*

2. *The optimal values of s_c and h_c satisfy*

$$s_c = d - f_c \text{ and } h_c = \frac{r_c}{d - f_c}, \tag{5.21}$$

where r_c is the smallest r such that $p_c(r) \leq \delta$.

An explanation with numerical results is given in Appendix 5.8.

The main result in the mean utilization dominated regime is given below.

Proposition 5.4.2. *For the multiple transmissions model in Sec. 5.3, under Assumptions 2, 3, and 4, and in the mean utilization dominated regime, the optimization problem $\mathcal{OP}_2$ can be decomposed across classes with the optimization for class c given by:*

$$\min_{m_c, r_c, h_c, s_c} : \sum_{m=0}^{m_c} r_c (p_c(r_c))^m \tag{5.22}$$

$$s.t. \quad s_c h_c = r_c,\ h_c \leq W,\ m_c\,(s_c + f_c) = d, \tag{5.23}$$

$$(p_c(r_c))^{m_c} \leq \delta. \tag{5.24}$$

Furthermore, under the finite block length model (5.5) relating r_c and $p_c(r_c)$, for $L \leq 2000$ bits, $d \leq 2$ msec, $\delta \in [10^{-3}, 10^{-7}]$, $SINR_c \in [0, 20]\,dB$ the optimal solution has the following structure:

1. *The optimal value of m_c is strictly more than one.*

2. *The optimal value of m_c is a non-increasing function of $SINR_c$.*

An explanation with numerical results is given in Appendix 5.9. Some observations regarding the two results are in order. If we compare the objective functions under mean and variance dominated regimes, the each term in the variance dominated regime is multiplied with an extra h_c. Since $h_c = \frac{r_c}{s_c} = \frac{r_c m_c}{d - m_c f_c}$, the objective function in the variance dominated regime increases sharply with increasing m_c. Therefore, the optimal value of m_c is lower in the variance dominated regime than in the mean dominated regime. In the mean dominated regime, as one decreases the SINR, the resources required per transmission (r_c) to meet a given reliability requirement increases sharply. Hence, it is advantageous at lower SINRs to increase m_c while choosing a lower reliability target per transmission. In the next section we shall further relax the assumption of homogeneous transmissions.

5.4.2 Repetition Coding– Heterogeneous Transmissions

In this section we shall again study $\mathcal{OP}_2$, however, we consider the possible benefits of heterogenous transmissions, i.e., $h_{c,m}$ and $r_{c,m}$ could possibly vary with m. We shall assume independent decoding across transmissions.

In the previous section, we have shown that in the variance dominated regime $m_c = 1$ is optimal, therefore, we do not optimize $h_{c,m}$ and $r_{c,m}$ in this setting. Instead we shall focus on the mean utilization dominated regime. URLLC users may access the channel every slot with a slot duration equal to that of a mini-slot ($0.125 - 0.25$ msec). Therefore, we cannot reduce $s_{c,m}$ to arbitrarily small values. Hence, we have shall place a lower bound on the minimum transmission duration for any stage which we denote by $s_{\min}$. The value of $s_{\min}$ is of the order of f_c as the feedback delay is at

least one slot. Therefore, we shall solve the following optimization problem:

$$\mathcal{OP}_5: \quad \min_{\mathbf{m},\mathbf{r},\mathbf{h},\mathbf{s}}: \sum_{c=1}^{C} \lambda_c \left(r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k}(r_{c,k}) \right) r_{c,m} \right), \tag{5.25}$$

$$\text{s.t.} \quad h_{c,m} s_{c,m} = r_{c,m} \quad \forall c,\, m, \tag{5.26}$$

$$\sum_{m=1}^{m_c} s_{c,m} + m_c f_c \leq d, \tag{5.27}$$

$$h_{c,m} \leq W \quad \forall c,\, m, \tag{5.28}$$

$$s_{c,m} \geq s_{\min} \quad \forall c,\, m, \tag{5.29}$$

$$\Pi_{m=1}^{m_c} p_{c,m}(r_{c,m}) \leq \delta \quad \forall c. \tag{5.30}$$

The main result in this setting is given below.

Proposition 5.4.3. *For the multiple transmissions model discussed in Sec. 5.3, under Assumptions 2 and 4, the optimization problem $\mathcal{OP}_5$ (mean dominated regime) can be decomposed across classes with the optimization for class c given by:*

$$\min_{m_c,\mathbf{r}_c,\mathbf{h}_c,\mathbf{s}_c}: \quad r_{c,1} + \sum_{m=2}^{m_c} \left(\Pi_{k=1}^{m-1} p_{c,k}(r_{c,k}) \right) r_{c,m}, \tag{5.31}$$

subject to constraints (5.26), (5.27), (5.28), (5.29), (5.30). *Furthermore, if we use the finite block length model* (5.5) *to relate $r_{c,m}$ and $p_{c,m}(r_{c,m})$ and restrict to $m_c = 2$, then we have the following conclusions:*

1. *For $L \leq 2000$ bits, $d \leq 2$ msec, $\delta \in [10^{-3}, 10^{-7}]$, $SINR_c \in [0, 20]\, dB$, the optimal value of $p_{c,1}(r_{c,1})$ is approximately 10^{-2}.*

2. *The marginal gains obtained by choosing an $m_c > 2$ is insignificant.*

An explanation with numerical results is given in Appendix 5.10. Consider the case where we restrict m_c to two. A low value of $p_{c,1}(r_{c,1})$, for example $p_{c,1}(r_{c,1}) \approx \delta$, ensures that there are very few re-transmissions. However, this is at the expense of a larger resource requirement $r_{c,1}$ for the initial transmission for all packets. At the other extreme, a higher value of $p_{c,1}(r_{c,1})$ can be achieved with a lower value of $r_{c,1}$ but at the expense of more re-transmissions. Hence, the term $p_{c,1}(r_{c,1})$ captures a trade-off between sending a robust initial transmission with fewer re-transmissions and sending a low reliability initial transmission with more re-transmissions. Our results suggest that most of the gains in reducing $p_{c,1}(r_{c,1})$ are obtained when $p_{c,1}(r_{c,1}) \approx 10^{-2}$ and further reduction is counter-productive. Choosing $p_{c,1}(r_{c,1}) \approx 10^{-2}$ also ensures that the resource utilization in the later transmissions has very little effect on the objective of $\mathcal{OP}_5$. Hence, there is very little advantage obtained by increasing m_c beyond two.

Since $\mathcal{OP}_5$ simplified to minimizing mean bandwidth utilization, the objective depends only on $r_{c,1}$ and $r_{c,2}$ for $m_c = 2$. One could choose any value of $h_{c,1}$ and $h_{c,2}$ such that constraints of $\mathcal{OP}_5$ are met. However, the variance of the system load is a non-decreasing functions of $h_{c,1}$ and $h_{c,2}$. Therefore, one would like to choose $h_{c,1}$ and $h_{c,2}$ such that the variance is as small as possible. Since the second transmission occurs only with probability 10^{-2}, one practical solution is to choose the lowest possible value for $h_{c,1}$ and the highest possible value for $h_{c,2}$, i.e., $h_{c,1} = \frac{r_{c,1}}{d-2f_c-s_{\min}}$ and $h_{c,2} = \frac{r_{c,2}}{s_{\min}}$. A representative figure contrasting the homogeneous and heterogeneous schemes is given in Fig. 5.2.

In the mean utilization dominated regime, the optimal value of m_c is often

SINR	Heterogeneous transmissions ($m_c = 2$)	Homogeneous transmissions with optimal m_c
0 dB	12.5	12.7
10 dB	3.4	3.4
20 dB	1.7	1.7

Table 5.1: Comparison of average bandwidth utilization (in MHz) under homogeneous and heterogeneous transmissions for different values of SINR. The other parameters are: $L = 100$ bits, $\lambda_c = 1000$ arrivals/sec., $d = 1$ msec., and $\delta = 10^{-6}$.

greater than 3, for example, for $L = 100$ bits, $SINR_c = 20$ dB, the optimal value of m_c is 3. A lower value of $SINR_c$ requires an even higher value of m_c. However, if we allow heterogeneous transmissions, choosing $m_c > 2$ does not give any significant benefit. Further numerical results show that one can achieve the same average bandwidth utilization with less number of stages under heterogeneous transmission. One such example is shown in Table 5.1 where we have the compared the average bandwidth utilization under homogeneous and heterogeneous transmissions. Note however that the control signal overhead for such a scheme would be higher as the BS has to signal the resource allocation for the second transmission and the coding rate used.

5.5 Conclusions

In this chapter we explored possible design of 5G wireless systems supporting URLLC traffic. We develop an appropriate model for URLLC packet transmissions which capture the essential properties of such a system pre-emptive/immediate URLLC scheduling and finite block-length transmissions. Based on this model we derive scaling results for URLLC capacity (admissible load subject to QoS constraints) with respect to various system parameters such as the link SINR, system bandwidth,

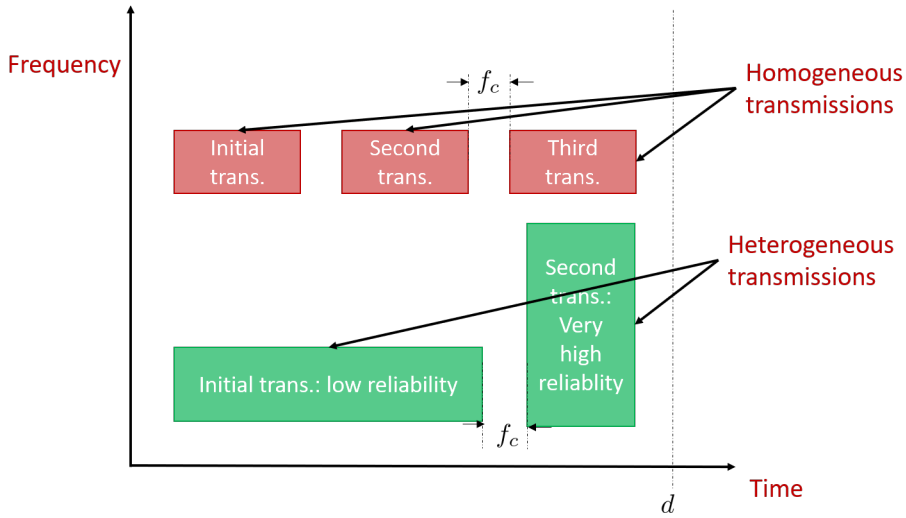


Figure 5.2: Comparison of repetition coding with homogeneous and heterogeneous transmissions. Observe that when we use heterogeneous transmissions, the initial transmission is spread out in time with a smaller bandwidth requirement, whereas the second transmission takes less time and uses a larger bandwidth.

and the packet latency and reliability requirements. Several key findings arise which are of practical interest. First, URLLC capacity is enhanced by extending URLLC transmissions in time as much as possible (subject to latency constraints) while using the least amount of bandwidth (to meet reliability requirements). Next we look at the results from the optimization of FEC and HARQ schemes. In the variance dominated regime (typically low loads), one-shot transmission satisfying the above mentioned requirements minimizes the necessary bandwidth required to support URLLC traffic. In the mean utilization dominated regime (high loads), optimal FEC/HARQ schemes minimizing the necessary bandwidth will leverage multiple transmissions and the maximum number of transmissions required depend on the type of FEC and HARQ schemes used and the SINR values.

Appendix

5.5.1 Proof of Theorem 5.2.1

Without loss of generality, let us consider $p_{b,1}(\mathbf{h}, \mathbf{s}, \boldsymbol{\lambda}, W)$. Using the standard results from queuing theory (see [33]), we have that:

$$\pi(\mathbf{n}) = G\,\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right), \tag{5.32}$$

where $G^{-1} = \sum_{\tilde{\mathbf{n}}\in\mathcal{S}} \Pi_{c=1}^{C}\left(\frac{\rho_c^{\tilde{n}_c}}{\tilde{n}_c!}\right)$ and $\mathcal{S} = \left\{\mathbf{n} \mid \mathbf{h}\mathbf{n}^T \leq W\right\}$. Here $\mathcal{S}$ is the set of all user configurations such that the total bandwidth constraint is not violated. Similarly define $\mathcal{S}' = \left\{\mathbf{n} \mid \mathbf{h}'\mathbf{n}^T \leq W\right\}$. From the definition of $\mathbf{h}$ and $\mathbf{h}'$, we have that

$$\mathbf{n} \in \mathcal{S} \Leftrightarrow [n_1, n_2, \ldots, qn_i, \ldots, n_C] \in \mathcal{S}'. \tag{5.33}$$

Define $\mathcal{S}_1 := \{\mathbf{n} \mid \mathbf{n} \in \mathcal{S} \text{ and } \mathbf{n} + \mathbf{e}_1 \notin \mathcal{S}\}$, where $\mathbf{e}_1$ is the unit vector with only the first coordinate as the non-zero element. $\mathcal{S}_1$ is the set of states in which class 1 users experience blocking. Similarly define $\mathcal{S}_1'$ for the case with bandwidths $\mathbf{h}'$ and $\mathbf{s}'$. Observe that due to (5.33) we have $\mathcal{S} \subseteq \mathcal{S}'$. Furthermore, if $\mathbf{n} \in \mathcal{S}_1$, then for $\tilde{n}_i \in \left\{qn_i - \lceil\frac{h_1 q}{h_i}\rceil + 1, qn_i - \lceil\frac{h_1 q}{h_i}\rceil + 2, \ldots, qn_i\right\}$ we have that $\mathbf{n}' := [n_1, n_2, \ldots, \tilde{n}_i, \ldots, n_C] \in \mathcal{S}_1'$. Using PASTA property (see [33]), the blocking probability experienced by a typical arrival to class 1 is given by

$$p_{b,1}(\mathbf{h}, \mathbf{s}, \boldsymbol{\lambda}, W) = \frac{\sum_{\mathbf{n}\in\mathcal{S}_1} \Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}} \Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}. \tag{5.34}$$

Similarly, the blocking probability experienced under $\mathbf{h}'$ and $\mathbf{s}'$ is given by

$$p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)=\frac{\sum_{\mathbf{n}\in\mathcal{S}_1'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}. \tag{5.35}$$

We will show that $p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)\leq p_{b,1}\left(\mathbf{h},\mathbf{s},\boldsymbol{\lambda},W\right)$. We can re-write (5.34) as follows:

$$p_{b,1}\left(\mathbf{h},\mathbf{s},\boldsymbol{\lambda},W\right)=\frac{1}{1+\frac{\sum_{\mathbf{n}\in\mathcal{S}\setminus\mathcal{S}_1}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}_1}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}}. \tag{5.36}$$

Next we will re-write (5.35) as follows:

$$p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)=\frac{1}{1+\frac{\sum_{\mathbf{n}\in\mathcal{S}\setminus\mathcal{S}_1}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}_1'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}+\frac{\sum_{\mathbf{n}\in\mathcal{S}'\setminus\mathcal{S}\setminus\mathcal{S}_1}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}_1'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}} \tag{5.37}$$

To compare $p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)$ and $p_{b,1}\left(\mathbf{h},\mathbf{s},\boldsymbol{\lambda},W\right)$, let us compare the denominators of (5.36) and (5.37). We will show the following:

$$\frac{\sum_{\mathbf{n}\in\mathcal{S}\setminus\mathcal{S}_1}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}_1'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}\geq\frac{\sum_{\mathbf{n}\in\mathcal{S}\setminus\mathcal{S}_1}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}{\sum_{\mathbf{n}\in\mathcal{S}_1}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)}. \tag{5.38}$$

If the above equation holds, then from (5.36) and (5.37), it can be easily shown that $p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)\leq p_{b,1}\left(\mathbf{h},\mathbf{s},\boldsymbol{\lambda},W\right)$. Note that in the above expression the numerator of the L.H.S. is greater than the numerator of the R.H.S. Next we have to compare the denominators. Due to (5.33), we can re-write the denominator of the L.H.S. as follows:

$$\sum_{\mathbf{n}\in\mathcal{S}_1'}q^{n_i}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right)=\sum_{\mathbf{n}\in\mathcal{S}_1}\sum_{i=1}^{\frac{qh_1}{h_i}}\frac{(q\rho_i)^{\left(qn_i-\lceil\frac{qh_1}{h_i}\rceil+i\right)}}{\left(qn_i-\lceil\frac{qh_1}{h_i}\rceil+i\right)!}\Pi_{c:c\neq i}\left(\frac{\rho_c^{n_c}}{n_c!}\right) \tag{5.39}$$

It can be shown that in wide-band systems with W large enough and $\rho_i < 1$ the following holds:

$$\sum_{\mathbf{n}\in\mathcal{S}_1}\sum_{i=1}^{\frac{qh_1}{h_i}}\frac{(q\rho_i)^{\left(qn_i-\lceil\frac{qh_1}{h_i}\rceil+i\right)}}{\left(qn_i-\lceil\frac{qh_1}{h_i}\rceil+i\right)!}\Pi_{c:c\neq i}\left(\frac{\rho_c^{n_c}}{n_c!}\right)\leq\sum_{\mathbf{n}\in\mathcal{S}_1}\Pi_{c=1}^{C}\left(\frac{\rho_c^{n_c}}{n_c!}\right). \tag{5.40}$$

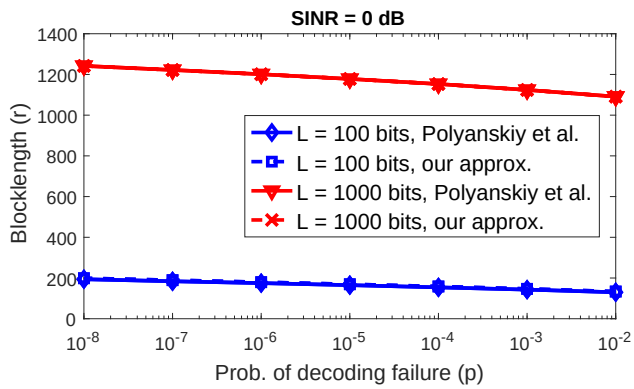
Therefore, the denominator of the L.H.S. of (5.38) is less than the denominator of its R.H.S. We have proved the inequality (5.38), and hence, $p_{b,1}\left(\mathbf{h}',\mathbf{s}',\boldsymbol{\lambda},W\right)\leq p_{b,1}\left(\mathbf{h},\mathbf{s},\boldsymbol{\lambda},W\right)$.

5.6 Approximate Expression for Blocklength

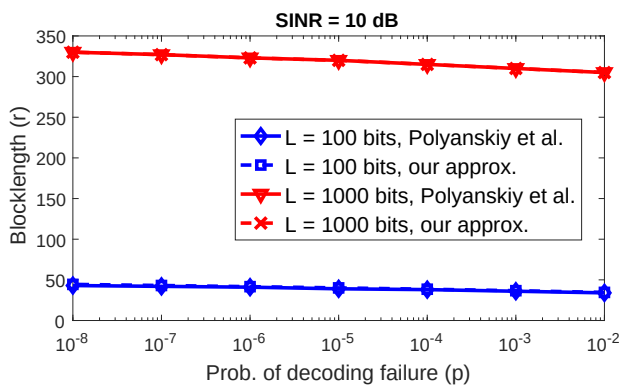
If we ignore the terms $0.5\log_2(r)$ and $o(1)$ in (5.5), we have the following approximate expression relating blocklength r, the number of information bits L and the probability of decoding failure p.

$$L\approx rC(SINR)-Q^{-1}\left(p\right)\sqrt{rV(SINR)}. \tag{5.41}$$

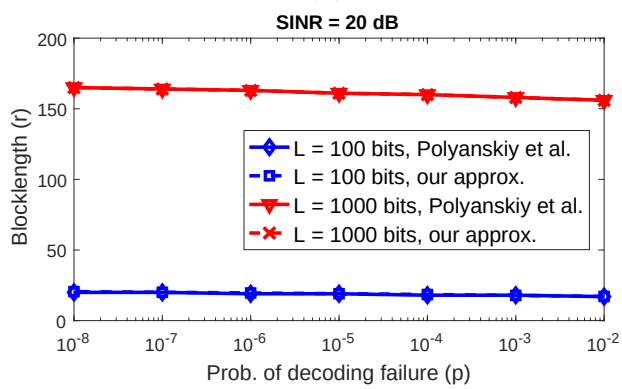
If we substitute $\sqrt{r}=x$, then the above equation is a quadratic equation in x. Solving it we get the approximate expression for r in (5.48). In Fig. 5.3 we have compared the values of r obtained from expression (5.5) and with the our approximation (5.48) for different packet sizes, SINRs and probability of decoding failure. Both the expressions give almost similar values of blocklengths.



(a)



(b)



(c)

Figure 5.3: Comparison r obtained using our approximation (5.48) with respect to the expression for blocklength derived in [1] and re-stated in (5.5).

5.7 Proof of Theorem 5.2.2

From (5.12) on single class system with one shot transmissions, we have the following relation between λ_c^* and W

$$W = \lambda_c^* r_c + c(\delta)\sqrt{h_c \lambda_c^* r_c}, \tag{5.42}$$

where r_c is chosen such that

$$r_c = \frac{L}{C(SINR_c)} + \frac{\left(Q^{-1}(\delta)\right)^2 V(SINR_c)}{2\left(C(SINR_c)\right)^2} + \frac{\left(Q^{-1}(\delta)\right)^2 V(SINR_c)}{2\left(C(SINR_c)\right)^2}\sqrt{1 + \frac{4LC(SINR_c)}{V(SINR_c)\left(Q^{-1}(\delta)\right)^2}}, \tag{5.43}$$

and $h_c = r_c/d$. Substituting h_c in (5.42), we get

$$W = \lambda_c^* r_c + c(\delta) r_c \sqrt{\frac{\lambda_c^*}{d}}. \tag{5.44}$$

Solving for λ_c^*, we have that

$$\lambda_c^* = \frac{W}{r_c} + \frac{c(\delta)^2}{d}\left(1 - \sqrt{1 + \frac{4Wd}{c(\delta)^2 r_c}}\right). \tag{5.45}$$

Scaling with respect to W directly follows from (5.45).

To understand the scaling with respect to $SINR_c$, we have to first study the scaling of r_c with respect to $SINR_c$. For large $SINR_c$, we have that

$$C(SINR_c) \sim \Theta\left(\log_2\left(SINR_c\right)\right), \tag{5.46}$$

$$V(SINR_c) \sim \Theta\left(1\right). \tag{5.47}$$

Therefore, $r_c \sim \Theta\left(\frac{1}{\log_2(SINR_c)}\right)$. Using (5.45), we get that $\lambda_c^* \sim \Theta\left(\log_2\left(SINR_c\right) - \sqrt{\log_2\left(SINR_c\right)}\right)$. Similarly, using (5.45), we get the scaling with respect to d as $\lambda_c^* \sim \Theta\left(1 - \frac{1}{\sqrt{d}}\right)$. If we use the square-root staffing rule with

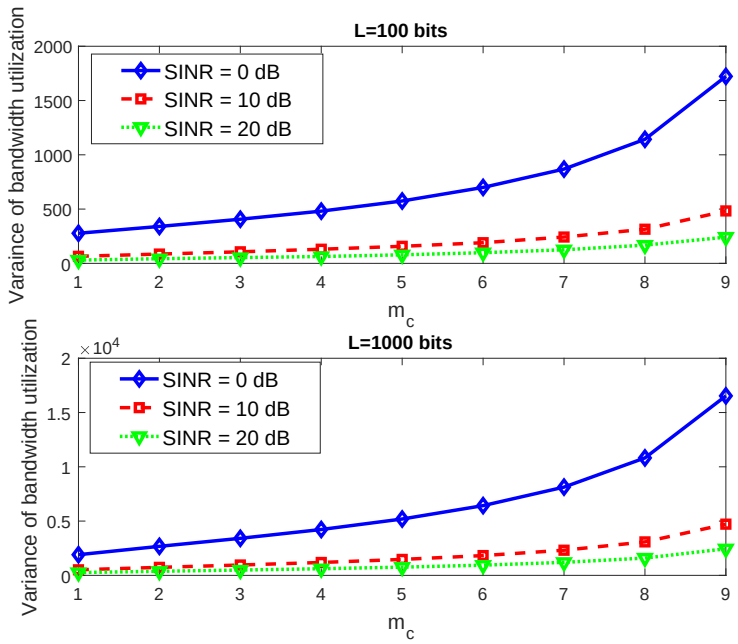


Figure 5.4: Variance of bandwidth utilization as a function of the number of stages for repetition coding with homogeneous transmissions and $\delta = 10^{-6}$.

the normal approximation (see [33]), we have that $c(\delta) = Q^{-1}(\delta) \sim \Theta\left(-\sqrt{\log(\delta)}\right)$. As we increase δ, we $c(\delta) \to 0$. Using $Q^{-1}(\delta) \sim \Theta\left(\sqrt{-\log(\delta)}\right)$ we have that $r_c \sim \Theta(-\log(\delta))$. Therefore, from (5.45) we get that $\lambda_c^* \sim \Theta\left(\frac{1}{-\log(\delta)}\right)$.

5.8 Numerical Results for Proposition 5.4.1

The optimal solutions to the decoupled problem for different values of SINR are plotted in Fig. 5.4. Note that $m_c = 1$ is optimal for all cases in the low load regime. To understand this, first observe that one can approximate the block length required r_c to transmit a packet of L bits with a probability of decoding failure target

of p_c under the finite block length model as follow:

$$r_c \approx \frac{L}{C(SINR)} + \frac{\left(Q^{-1}\left(p_c\right)\right)^2 V(SINR)}{2\left(C(SINR)\right)^2} + \frac{\left(Q^{-1}\left(p_c\right)\right)^2 V(SINR)}{2\left(C(SINR)\right)^2}\sqrt{1+\frac{4LC(SINR)}{V(SINR)\left(Q^{-1}\left(p_c\right)\right)^2}}. \quad (5.48)$$

This is derived in Appendix 5.6. Under repetition coding, the maximum probability of failure in each stage (p_c) is the same and equal to δ^{1/m_c} so that after m_c stages, the probability of failure would be exactly δ. Also, we have that

$$\sum_{m=0}^{m_c} h_c^2 \lambda_c s_c\left(p_c^m\right) = \frac{\lambda_c h_c r_c\left(1-\delta\right)}{1-\delta^{1/m_c}} \quad (5.49)$$

In Thm. 5.2.1, we have proved that extending the transmissions in time till the deadline is beneficial towards reducing the blocking probability. Using this property and from constraint that $h_c s_c = r_c$ and $s_c = d - m_c f_c$ we have that:

$$h_c = \frac{r_c}{\frac{d}{m_c} - f_c} \quad (5.50)$$

We also know that $Q^{-1}(\epsilon) \sim \Theta\left(\sqrt{-\log\epsilon}\right)$. Therefore, in (5.49), $r_c \sim \frac{L}{C(SINR_c)} + \Theta\left(\frac{1}{m_c}\right)$, $h_c \sim \Theta\left(m_c\right)$ and denominator is $1-\delta^{1/m_c}$. The bandwidth h_c, which is a non-decreasing function of m_c is the most sensitive to changes of m_c for the range of SINRs seen in a wireless system. Therefore, for range of SINRs in a wireless system $m_c = 1$ is the optimal solution.

5.9 Numerical Results for Proposition 5.4.2

In Fig. 5.5 we have plotted the average bandwidth utilization for various SINRs and packet sizes. The key observation is given below

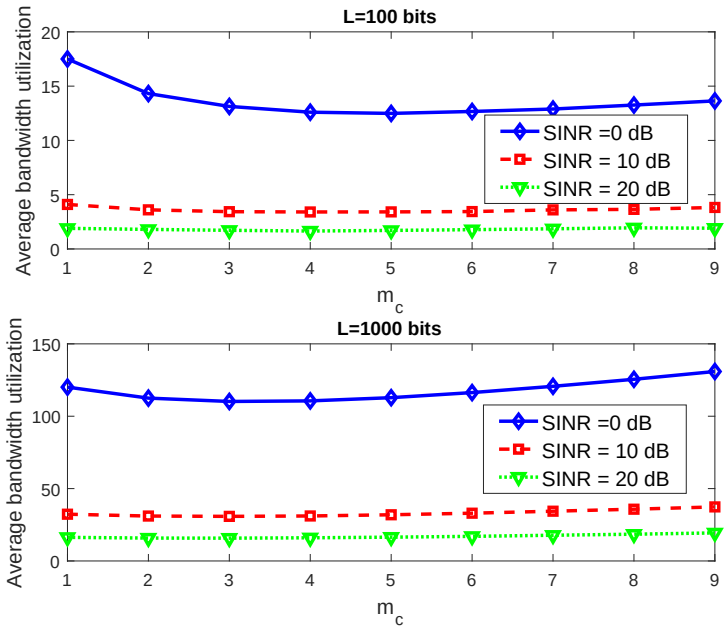


Figure 5.5: Average bandwidth utilization as a function of the number of stages for repetition coding with homogeneous transmissions for $\lambda = 100$ arrivals/sec. $d = 1$ msec., and $\delta = 10^{-6}$.

Observation 5.9.1. *For a given L, δ, and d, the optimal value of m_c decreases with increasing $SINR_c$.*

To understand the above observation, we use (5.48) with $p = \delta^{\frac{1}{m_c}}$. The expression for r_c as a function of m_c is given below

$$r_c \approx \frac{L}{C(SINR_c)} + \frac{\left(Q^{-1}\left(\delta^{1/m_c}\right)\right)^2 V(SINR_c)}{2\left(C(SINR_c)\right)^2} + \frac{\left(Q^{-1}\left(\delta^{1/m_c}\right)\right)^2 V(SINR_c)}{2\left(C(SINR_c)\right)^2}\sqrt{1+\frac{4LC(SINR_c)}{V(SINR_c)\left(Q^{-1}\left(\delta^{1/m_c}\right)\right)^2}} \quad (5.51)$$

We have the following lemma which is proved in Appendix 5.10.1.

Lemma 5.9.2. *There exists an $\epsilon > 0$ such that for $SINR_c \in [0, \epsilon]$, the term $\frac{V(SINR_c)}{(C(SINR_c))^2}$ is a non-increasing function of $SINR_c$ and $\lim_{SINR_c \to 0} \frac{V(SINR_c)}{(C(SINR_c))^2} = \infty$.*

Note that for a given δ, $\left(Q^{-1}\left(\delta^{1/m_c}\right)\right)^2 = \Theta\left(\frac{1}{m_c}\right)$. From the above lemma as $SINR_c$ decreases the constant that multiplies $\left(Q^{-1}\left(\delta^{1/m_c}\right)\right)^2$ increases, and hence the average bandwidth utilization increases sharply in reducing m_c. Therefore, a higher value of m_c is optimal for lower SINRs. To summarize, at high loads, multiple re-transmissions are preferred, and the number of re-transmissions required increases with decreasing SINR.

5.10 Numerical Results for Proposition 5.4.3

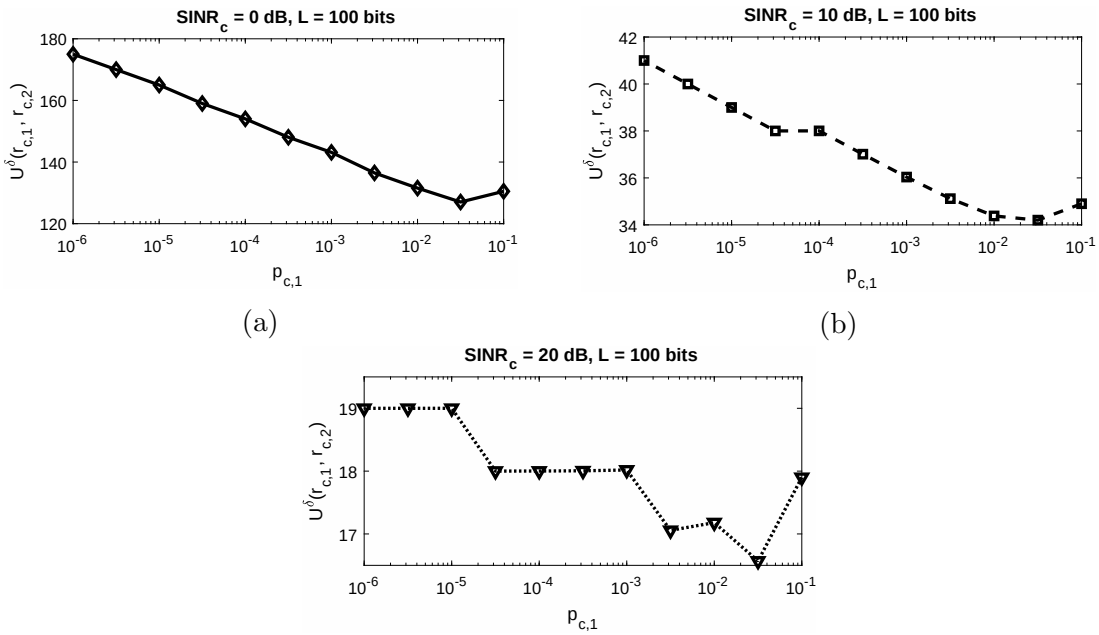
Let $U^\delta(r_{c,1}, r_{c,2}) := r_{c,1} + p_{c,1} r_{c,2}$. $U^\delta(r_{c,1}, r_{c,2})$ as a function of $p_{c,1}$ for different values of SINR is plotted in Figures 5.6 and 5.7. Note that once we reduce $p_{c,1}$ to 10^{-2} the term $p_{c,1} r_{c,2}$ is very small and hence, reducing $p_{c,1}$ any further does not

SINR	$U_c^\delta(r_{c,1}, r_{c,2})$ (MHz)	$U_c^\delta(r_{c,1}, r_{c,2}, r_{c,3})$ (MHz)
0 dB	12.7	12.7
10 dB	3.4	3.4
20 dB	1.7	1.7

Table 5.2: Comparison of $U_c^\delta(r_{c,1}, r_{c,2})$ and $U_c^\delta(r_{c,1}, r_{c,2}, r_{c,3})$ for different values of SINR for $L = 100$ bits, $\lambda_c = 1000$ arrivals/sec., $d = 1$ msec., and $\delta = 10^{-6}$.

help. Note that in Fig. 5.6c $U^\delta(r_{c,1}, r_{c,2})$ has a piece-wise linear structure. This is because of the quantization of $r_{c,1}$ to integer values. At high SINRs increasing the block length by one causes a sudden drop in $p_{c,1}$. For $L = 100$ bits, the optimal value of $p_{c,1} \approx 2 \times 10^{-2}$, whereas, for $L = 1000$ bits optimal $p_{c,1}$ is 10^{-2}. To understand this consider (5.51), the block length required is more sensitive to the second term when $L/C(SINR_c)$ is smaller relative to the second term. Therefore, for a large value of L we can have a slightly lower value of $p_{c,1}$. However, this effect is not so significant. To summarize, the optimal value of $p_{c,1}$ is close to 10^{-2} for all cases.

In Table 5.2 we have compared average bandwidth utilization under the optimal HARQ schemes obtained numerically for $m_c = 2$ and $m_c = 3$. Note that there is no difference between the values for $m_c = 2$ and $m_c = 3$. Most of the benefits of using heterogeneous transmissions are obtained from two stages of HARQ. This is because the value of $p_{c,1} \approx 10^{-2}$ and hence, the effect of any additional stages on the objective function is insignificant. Therefore, we restrict ourselves to $m_c = 2$.



(c)

Figure 5.6: $U^\delta(r_{c,1}, r_{c,2})$ as a function of $p_{c,1}$ for different SINRs and $L = 100$ bits.

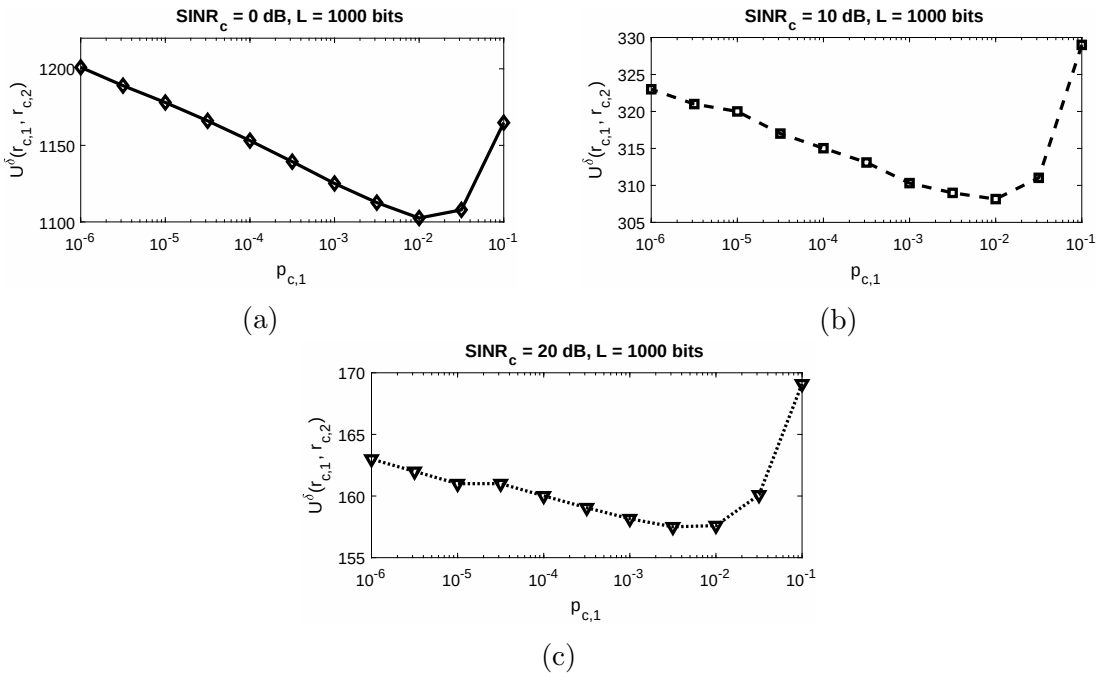


Figure 5.7: $U^{\delta}\left(r_{c,1}, r_{c,2}\right)$ as a function of $p_{c,1}$ for different SINRs and $L = 1000$ bits.

5.10.1 Proof of Lemma 5.9.2

One can re-write $V(SINR_c)/\left(C(SINR_c)\right)^2$ as follows:

$$\frac{V(SINR_c)}{\left(C(SINR_c)\right)^2} = \frac{2SINR_c + (SINR_c)^2}{(1+SINR_c)^2\left(\log_2\left(1+SINR_c\right)\right)^2}. \tag{5.52}$$

If we make the substitution $x = SINR_c$, we have that

$$\frac{V(SINR_c)}{\left(C(SINR_c)\right)^2} = \frac{x^2-1}{\left(x\log_2(x)\right)^2}. \tag{5.53}$$

Let us also define $f(x) := \frac{x^2-1}{(x\log_2(x))^2}$. If we take the derivative of $f(x)$, denoted by $f'(x)$, we get that

$$f'(x) = \frac{-2x\log_2(x)\left(x^2+1+\log_2(x)\right)}{\left(x\log_2(x)\right)^4}. \tag{5.54}$$

In the above expression, the numerator of R.H.S. is negative for small values of x. This is obtained by the fact that $x^2+1+\log_2(x)$ is negative for small x and $-2x\log_2(x)$ is positive for small x. Therefore, one can conclude that there exists ϵ such that $f(x)$ is non-increasing in the interval $[0,\epsilon]$. In the limit as $x \to 0$, the denominator $f(x)$ goes to 0 while the numerator tends to -1. Hence, $\lim_{x\to 0} f(x) = \infty$.

Chapter 6

Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks

6.1 Introduction

An [1] important requirement for 5G wireless systems is its ability to efficiently support both broadband and ultra-low-latency reliable communications. On one hand, broadband traffic – formally, enhanced Mobile Broadband (eMBB) – should support gigabit per second data rates (with a bandwidth of several 100 MHz) with moderate latency (a few milliseconds). On the other hand, Ultra Reliable Low Latency Communication (URLLC) traffic requires extremely low delays (0.25-0.3 msec/packet) with very high reliability (99.999%) [15]. To satisfy these heterogenous requirements, the 3GPP standards body has proposed an innovative *superposition/puncturing* framework for multiplexing URLLC and eMBB traffic in 5G cellular systems.

The proposed scheduling framework has the following structure [15]. As with current cellular systems, time is divided into slots, with proposed one millisecond (msec) slot duration. Within each slot, eMBB traffic can share the bandwidth over

[1] This chapter is a joint work with Prof. Sanjay Shakkottai. Publications based on this chapter: A. Anand, G. de Veciana, and S. Shakkottai, "Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks", in Proceedings of INFOCOM, 2018 (accepted). Arxiv version [78].

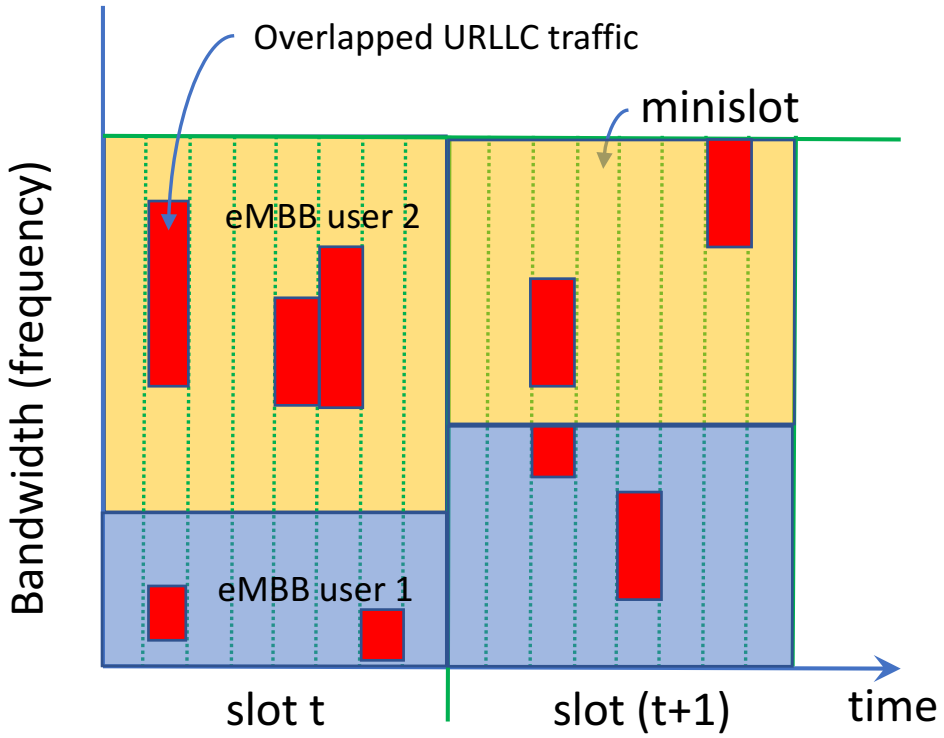


Figure 6.1: Illustration of superposition/puncturing approach for multiplexing eMBB and URLLC: Time is divided into slots, and further subdivided into minislots. eMBB traffic is scheduled at the beginning of slots (sharing frequency across two eMBB users), whereas URLLC traffic can be dynamically overlapped (superpose/puncture) at any minislot.

the time-frequency plane (see Figure 6.1). The sharing mechanism can be opportunistic (based on the channel states of various users); however, the eMBB shares are decided by the beginning, and fixed for the duration of a slot[2].

URLLC downlink traffic may arrive during an ongoing eMBB transmission;

[2]The sharing granularity among various eMBB users is at the level of Resource Blocks (RB), which are small time-frequency rectangles within a slot. In LTE today, these are (1 msec $\times$ 180 KHz), and could be smaller for 5G systems.

if tight latency constraints are to be satisfied, they cannot be queued until the next slot. Instead each eMBB slot is divided into minislots, each of which has a 0.125 msec duration[3]. Thus upon arrival URLLC demand can be immediately scheduled in the next minislot *on top of the ongoing eMBB transmissions.* If the Base Station (BS) chooses non-zero transmission powers for both eMBB and overlapping URLLC traffic, then this is referred to as *superposition.* If eMBB transmissions are allocated zero power when URLLC traffic is overlapped, then it is referred to as *puncturing* of eMBB transmissions. The superposed/punctured URLLC traffic is sufficiently protected (through coding and HARQ if necessary) to ensure that it is reliably transmitted. At the end of an eMBB slot, the BS can signal the eMBB users the locations, if any, of URLLC superposition/puncturing. The eMBB user can in turn use this information to decode transmissions, with some possible loss of rate depending on the amount of URLLC overlaps. We refer to [15, 16] for additional details.

A key problem in this setting is thus the *joint scheduling of eMBB and URLLC traffic over two time-scales.* At the slot boundary, resources are allocated to eMBB users based on their channel states and utilities, in effect, allocating long term rates to optimize high-level goals (e.g. utility optimization). Meanwhile, at each minislot boundary, the (stochastic) URLLC demands are overlapped (superposed/punctured) onto previously allocated eMBB transmissions. Decisions on the placement of such overlaps across scheduled eMBB user(s) will impact the rates they will see on that slot. Thus we have a coupled problem of jointly optimizing the scheduling of eMBB

[3]In 3GPP, the formal term for a 'slot' is eMBB TTI, and a 'minislot' is a URLLC TTI, where TTI expands to Transmit Time Interval.

users on slots with the placement of URLLC demands across minislots.

6.1.1 Main Contributions

This work is, to our knowledge, the first to formalize and solve the joint eMBB/URLLC scheduling problem described above. We consider various models for the eMBB rate loss associated with URLLC superposition/puncturing, for which we characterize the associated feasible throughput regions and online joint scheduling algorithms as detailed below.

(**Linear Model**): When the rate loss to eMBB is directly proportional to the fraction of superposed/punctured minislots, we show that the joint optimal scheduler has a nice decomposition: the stochastic URLLC traffic can be uniform-randomly scheduled in each minislot, and the eMBB scheduler can be scheduled via a greedy iterative gradient algorithm the only accounts for the expected rate loss due to the URLLC traffic.

(**Convex Model**): For more general models where the rate loss can be modeled through a convex function, we restrict to time homogeneous policies. In this setting, we characterize the capacity region and derive concavity conditions under which we can derive the effective rate seen by eMBB users (post-puncturing by URLLC traffic). We then develop a stochastic approximation algorithm jointly schedules eMBB and URLLC traffic, and show that it asymptotically maximizes utility for eMBB users while satisfying URLLC demands.

(**Threshold Model**): We finally consider a threshold model, where eMBB traffic is unaffected by puncturing until a threshold; beyond this threshold it suffers complete

throughput loss (a 0-1 rate loss model). We consider two broad classes of time homogeneous policies, where the URLLC traffic is placed in minislots proportional to either the eMBB allocated bandwidths (Rate Proportional) or the eMBB thresholds (Threshold Proportional). We motivate these policies (e.g. minimizes probability of eMBB loss in any slot) and derive the associated throughput regions. Finally, we utilize the additional structure imposed by the RP and TP Placement policies along with the shape of the threshold loss function and derive fast gradient algorithms that converge and provably maximize utility.

The related work for this chapter has been covered in detail in the previous chapter. Next we shall explain our system model.

6.2 System Model

Traffic model. We consider a wireless system supporting a fixed set of backlogged eMBB users $\mathcal{U}$ and stationary URLLC traffic demands. eMBB scheduling decisions are made across slots while URLLC demands arrive and are immediately scheduled across minislots. Each eMBB slot has an associated set of minislots where $\mathcal{M} = \{1, \ldots |\mathcal{M}|\}$ denotes there indices. URLLC demands across minislots are modeled as a independent and identically distributed (i.i.d.) random random process. We let the random variables $(D(m), m \in \mathcal{M})$ denote the URLLC demands per minislot for a typical eMBB slot. We let D be a random variable whose distribution is that of the aggregate URLLC demand per eMBB slot, i.e., $D \sim \sum_{m \in \mathcal{M}} D(m)$ with, cumulative distribution function $F_D()$ and mean $E[D] = \rho$. We assume demands have been normalized so the maximum URLLC demand per minislot is f and the

maximum aggregate demands per eMBB slot is $f \times |\mathcal{M}| = 1$ i.e., all the frequency-time resources are occupied. URLLC demands per minislot exceeding the system capacity are blocked by URLLC scheduler thus $D \leq 1$ almost surely. As mentioned earlier the system is engineered so that blocked URLLC traffic on a minislot is a rare event, i.e., satisfies the desired reliability on such traffic.

Wireless channel variations. The wireless system experiences channel variations each eMBB slot which are modeled as an i.i.d. random process over set of channel states $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$. Let S be a random variable modeling the distribution over the states in a typical eMBB slot with probability mass function $p_S(s) = P(S = s)$ for $s \in \mathcal{S}$. For each channel state s eMBB user u has a known peak capacity $\hat{r}_u^s$. The wireless system can choose what proportions of the frequency-time resources to allocate to each eMBB user on each minislot for each channel state. This is modeled by a matrix $\boldsymbol{\phi} \in \Sigma$ where

$$\Sigma := \left\{ \mathbf{x} \in \mathbb{R}_+^{|\mathcal{U}| \times |\mathcal{M}| \times |\mathcal{S}|} \mid \sum_{u \in \mathcal{U}} x_{u,m}^s = f, \forall m \in \mathcal{M}, s \in \mathcal{S} \right\} \tag{6.1}$$

and where the element $\phi_{u,m}^s$ represents the fraction of resources allocated to user u in mini slot m in channel state s. We also let $\phi_u^s = \sum_{m \in \mathcal{M}} \phi_{u,m}^s$, i.e., the total resources allocated to user u in an eMBB slot in channel state s. Now assuming no superposition/puncturing if the system is in channel state s and the eMBB scheduler chooses an allocation $\boldsymbol{\phi}$ the rate r_u allocated to user u would be given by $r_u = \phi_u^s \hat{r}_u^s$. The scheduler is assumed to know the channel state and can thus exploit such variations opportunistically in allocating resources to eMBB users. Note that for simplicity, we adopt a flat-fading model, namely, the rate achieved by an user is

directly proportional to the fraction of bandwidth allocated to it (the scaling factor is the peak rate of the user for the current channel state).

Class of joint eMBB/URLLC schedulers. We consider a class of stationary joint eMBB/URLLC schedulers denoted by Π satisfying the following properties. A scheduling policy combines a possibly state dependent eMBB *resource allocation* ϕ per slot with a URLLC *demand placement* strategy across minislots. The placement strategy may impact the eMBB users' rates since it affects the URLLC superposition/puncturing loads they will experience. As mentioned earlier in discussing the traffic model, in order to meet low latency requirements URLLC traffic demands are scheduled immediately upon arrival or blocked. The scheduler is assumed to be *causal* so it only knows the current (and past) channel states and achieved rates $\hat{r}_u^s, \forall, u \in \mathcal{U}, s \in \mathcal{S}$ but does not know the realization of future channels or URLLC traffic demands. In making superposition/puncturing decisions across minislots, the scheduler can use knowledge of the previous placement decisions that were made. In addition the scheduler is assumed to know (or can measure over time) the channel state distribution across eMBB slots and URLLC demand distributions per minislot i.e., that of $D(m)$, and per eMBB slot, i.e., D, and thus knows in particular $\rho = E[D]$.

In summary joint scheduling policy $\pi \in \Pi$ is thus characterized by the following:

- an eMBB resource allocation $\boldsymbol{\phi}^\pi \in \Sigma$ where $\phi_{u,m}^{\pi,s}$ denotes the fraction frequency-time slot resources allocated to eMBB user u on minislot m when the system

is in state s.

- the distributions of URLLC loads across eMBB resources induced by its URLLC placement strategy, denoted by random variables $\mathbf{L}^{\pi} = (L^{\pi,s}_{u,m} | u \in \mathcal{U}, m \in \mathcal{M}, s \in \mathcal{S})$ where $L^{\pi,s}_{u,m}$ denotes the URLLC load superposed/puncturing the resource allocation of user u on minislot m when the channel is in state s.

The distributions of $L^{\pi,s}_{u,m}$ and their associated means $l^{\pi,s}_{u,m}$ depend on the joint scheduling policy π, but for all states, users and minislots satisfy

$$L^{\pi,s}_{u,m} \leq \phi^{\pi,s}_{u,m} \quad \text{almost surely.}$$

In the sequel we let $L^{\pi,s}_{u} = \sum_{m \in \mathcal{M}} L^{\pi,s}_{u,m}$, i.e., the aggregate URLLC traffic superposed/puncturing user u in channel state s, and denote its mean by $l^{\pi,s}_{u}$ and note that

$$L^{\pi,s}_{u} \leq \phi^{\pi,s}_{u} \quad \text{almost surely.}$$

We shall also $L^{\pi,s} = \sum_{u \in \mathcal{U}} L^{\pi,s}_{u}$ denote the aggregate induced load and note that any policy π and any state s we have that

$$\rho = \mathrm{E}[D] = \mathrm{E}[L^{\pi}] = \mathrm{E}[\sum_{u \in \mathcal{U}} L^{\pi,s}_{u}] = \sum_{u \in \mathcal{U}} l^{\pi,s}_{u}.$$

Modeling superposition/puncturing and eMBB capacity regions. Under a joint scheduling policy π we model the rate achieved by an eMBB user u in channel state s by a random variable

$$R^{\pi,s}_{u} \quad = \quad f^{s}_{u}(\phi^{\pi,s}_{u}, L^{\pi,s}_{u}) \tag{6.2}$$

where the *rate allocation function* $f_u^s(\cdot,\cdot)$ models the impact of URLLC superposition/puncturing – one would expect it to be increasing the first argument (the allocated resources) and decreasing in the second argument (the amount superposition/puncturing by URLLC traffic). One would also expect such functions to satisfy

$$f_u^s(\phi_u^s, l_u^s) = 0$$

if $\phi_u^s = l_u^s$, i.e., if superposition/puncturing occurs across all of an eMBB users resources no data is successfully transmitted, however, perhaps under the superposition some rate might still be extracted from the transmission. Also under our system model we have that

$$R_u^{\pi,s} \leq f_u^s(\phi_u^{\pi,s}, 0) = \phi_u^{\pi,s}\hat{r}_u^s \text{ almost surely,}$$

with equality if there is no superposition/puncturing, i.e., when $l_u^s = 0$. We shall $r_u^{\pi,s} = E[R_u^{\pi,s}]$ denote the mean rates achieved by user u in state s under the URLLC superposition/puncturing distribution induced by scheduling policy π.

Models for Throughput Loss: In the sequel we shall consider specific forms of superposition/puncturing models: *(i)* linear, *(ii)* convex, and *(iii)* threshold models.

We rewrite the rate allocation function in (6.2) as the difference between the peak throughput and the loss due to URLLC traffic, and consider functions that can be decomposed as:

$$f_u^s(\phi_u^s, l_u^s) = \hat{r}_u^s\phi_u^s\left(1 - h_u^s\left(\frac{l_u^s}{\phi_u^s}\right)\right),$$

where $h_u^s : [0,1] \to [0,1]$ is the *rate loss function* and captures the relative rate loss due to URLLC overlap on eMBB allocations. The puncturing models we study now

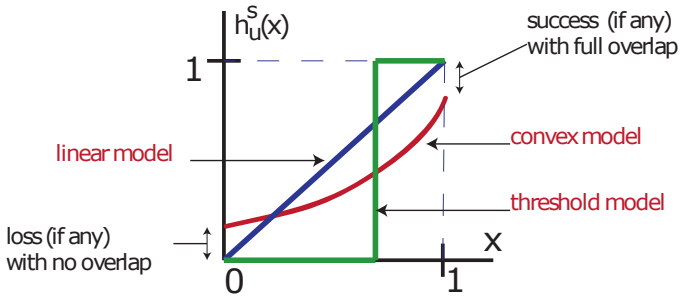


Figure 6.2: The illustration exhibits the rate loss function for the various models considered in this chapter, linear, convex and threshold.

map directly to structural assumptions on the rate loss function $h_u^s(\cdot)$; namely it is a non-decreasing function, and is one of *linear, convex, or threshold* as shown in Figure 6.2.

Linear Model: Under the linear model, the expected rate for user u in channel state s for policy π is given by

$$r_u^{\pi,s} = \mathrm{E}[f_u^s(\phi_u^{\pi,s}, L_u^{\pi,s})] = \hat{r}_u^s(\phi_u^{\pi,s} - l_u^{\pi,s}),$$

i.e., $h_u^s(x) = x$, and the resulting rate to eMBB users is a linear function of both the allocated resources and mean induced URLLC loads. This model is motivated by basic results for the channel capacity of AWGN channel with erasures, see [79] for more details. Our system in a given network state can be approximated as an AWGN channel with erasures, when the slot sizes are long enough so that the physical layer error control coding of eMBB users use long code-words. Further, there is a dedicated control channel through which the scheduler can signal to the eMBB receiver indicating the positions of URLLC overlap. Indeed such a control channel has been proposed in the 3GPP standards [15]. Note that under this model the

rate achieved by a given user depends on the aggregate superposition/puncturing it experiences, i.e., does not depend on which minislots and frequency bands it occurs. We discuss the policies for the linear model in Section 6.4.

Convex Model: In the convex model, the rate loss function $h_u^s(\cdot)$ is convex (see Figure 6.2), and the resulting rate for eMBB user u in channel state s under policy π is given by

$$r_u^{\pi,s} = \mathrm{E}[f_u^s(\phi_u^{\pi,s}, L_u^{\pi,s})] = \hat{r}_u^s \phi_{\pi,s}^u \left(1 - E\left[h_u^s\left(\frac{L_u^{\pi,s}}{\phi_u^{\pi,s}}\right)\right]\right).$$

This covers a broad class of models, and is discussed in Section 6.5.

Threshold Model: Finally the threshold model is designed to capture a simplified packet transmission and decoding process in an eMBB receiver. The data is either received perfectly or it is lost depending on the amount of superposition/puncturing. With slight abuse of notation we shall let h_u^s also depend on both the relative URLLC load and the eMBB user allocation, i.e., $h_u^s(x, \phi_u^s) = \mathbf{1}(x \leq t_u^s(\phi_u^s))$ where the threshold in turn is an increasing function $t_u^s()$ satisfying and satisfy $x \geq t_u^s(x) \geq 0$. Such thresholds might reflect various engineering choices where codes are adapted when users are allocated more resources, so as to be more robust to interference/URLLC superposition/puncturing. The resulting rate for eMBB user u in channel state s and policy π is then given by

$$r_u^{\pi,s} = \hat{r}_u^s \phi_u^{\pi,s} P(L_{\pi,u}^s \leq \phi_u^{\pi,s} t_u^s(\phi_u^{\pi,s})).$$

While such a sharp falloff is somewhat extreme, it is nevertheless useful for modeling short codes that are designed to tolerate a limited amount of interference. In practice

one might expect a smoother fall off, perhaps more akin to the convex model, e.g., when hybrid ARQ (HARQ) is used. We discuss polices under the threshold based model in Section 6.6.

Capacity for eMBB traffic: We define the capacity $\mathcal{C} \subset \mathbb{R}_+^{|\mathcal{U}|}$ for eMBB traffic as the set of long term rates achievable under policies in Π. Let $\mathbf{c}^\pi = (c_u^\pi | u \in \mathcal{U})$ where

$$c_u^\pi = \sum_{s \in \mathcal{S}} r_u^{\pi,s} p_S(s).$$

Then the capacity is given by

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists\, \pi \in \Pi \text{ such that } \mathbf{c} \leq \mathbf{c}^\pi\}.$$

Note that this capacity region depends on the scheduling policies under consideration as well as the distributions of the channel states and URLLC demands.

Scheduling objective: URLLC priority and eMBB utility maximization: As mentioned earlier, URLLC traffic is immediately placed upon arrival, at the minislot scale, i.e, no queueing is allowed. Thus if demands exceed the system capacity on a given minislot such traffic is lost. The system is engineered so that such URLLC overloads are extremely rare, and thus URLLC traffic can meet extremely low latency requirements with high reliability. For eMBB traffic we adopt a utility maximization framework wherein each eMBB user u has an associated utility function $U_u(\cdot)$ which is a strictly concave, continuous and differentiable of the average rate c_u^π experienced by the user. Our aim is to characterize optimal rate allocations associated with the utility maximization problem:

$$\max_{\mathbf{c}} \{\sum_{u \in \mathcal{U}} U_u\left(c_u\right) \mid \mathbf{c} \in \mathcal{C}\}, \tag{6.3}$$

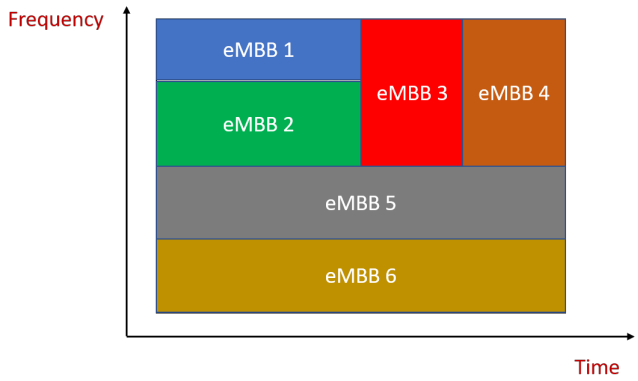


Figure 6.3: An example of eMBB resource allocations in 5G NR time-frequency plane.

and determine and associated scheduling policy π that will realize such allocations.

6.3 Optimal eMBB Placement in Time-Frequency Plane

3GPP New Radio frame structure allows flexible resource allocation for eMBB users in the time-frequency plane. In an eMBB slot, eMBB users can share resources in time or frequency . If only time is shared among eMBB users, the entire frequency is allocated to an eMBB user in a mini-slot. Similarly if only frequency resource are share among eMBB users, then a part of the bandwidth is allocated to an eMBB user for the entire eMBB slot. Sharing resources in the time and frequency domains are illustrated in Figures 6.4 and 6.5, respectively. In this section, we will show that sharing resources in the frequency domain results in a better average rate for the eMBB users if the loss functions $h_u^s(\cdot)$ are convex.

The essence of the problem can be captured in a setting with two eMBB users, i.e., $|\mathcal{U}| = 2$. We shall look two resource allocation configurations for eMBB users

given in Figures 6.4 and 6.5. In configuration 1, eMBB user 1 is allocated the entire bandwidth for m_1 mini-slots and the remaining $|\mathcal{M}| - m_1$ mini-slots are allocated to eMBB user 2. Define $\phi_1 := \frac{m_1}{|\mathcal{M}|}$ and $\phi_2 = \frac{|\mathcal{M}|-m_1}{|\mathcal{M}|}$. Instead in configuration 2, we shall allocate an eMBB user a fraction ϕ_u of the bandwidth for the entire eMBB slot. In configuration 1, the total puncturing observed by eMBB user u is given by $\sum_{m=1}^{\phi|\mathcal{M}|} D_m$. In configuration 2, under uniform URLLC placement, the total puncturing observed by eMBB user u is given by $\sum_{m=1}^{|\mathcal{M}|} \phi_u D_m$. Let us first define *exchangeable* set of random variables.

Definition An ordered set of random variables $\left\{D_1, D_2, \ldots, D_{|\mathcal{M}|}\right\}$ is said to be exchangeable if the probability distribution is same for any permutation of the set.

The main result of this section is given below:

Theorem 6.3.1. *Under the assumptions of exchangeable URLLC demands in an eMBB slot ($\left\{D_1, D_2, \ldots, D_{|\mathcal{M}|}\right\}$) and convex loss functions ($h_u(\cdot)$), if $\mathbb{E}\left[h_u(D_1)\right] < \infty$, we have that*

$$\mathbb{E}\left[h_u\left(\sum_{m=1}^{\phi_u|\mathcal{M}|} D_m\right)\right] \geq \mathbb{E}\left[h_u\left(\sum_{m=1}^{|\mathcal{M}|} \phi_u D_m\right)\right]. \tag{6.4}$$

Remarks: The above theorem shows that the expected loss suffered by an eMBB user due to URLLC puncturing in configuration 1 is higher than in configuration 2. Configuration 2 gives more flexibility in the URLLC placement as well as lesser variability in the total puncturing. Since the loss function is convex, this will naturally lead to a lower loss. Therefore, in any configuration if eMBB users

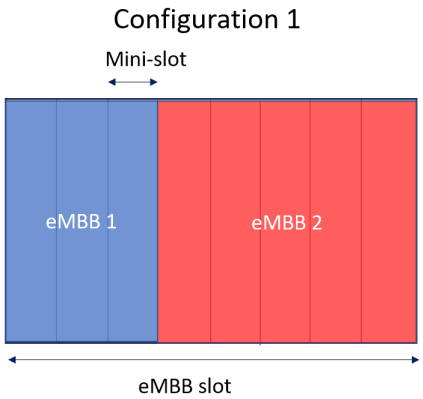


Figure 6.4: In this configuration, eMBB users share time and do not share the bandwidth in an eMBB slot.

are allocated resources for different duration in an eMBB slot, then we can replace the configuration with an equivalent configuration which allocates same amount of resources for all eMBB users but with same transmit duration and different bandwidths, while suffering lower losses from puncturing by URLLC traffic.

For more general configurations, for example the configuration given in Fig. 6.3, one can apply the Thm. 6.3.1 iteratively and show that sharing resource exclusively in the frequency domain is better than sharing resources in time domain. Therefore, we shall restrict ourselves to resource allocation schemes which share eMBB resources in the frequency domain in an eMBB slot.

6.4 Linear Model for Superposition/Puncturing

As a thought experiment, consider a two-user system, with users having the same utility function (say square root function), but i.i.d. (across time and users)

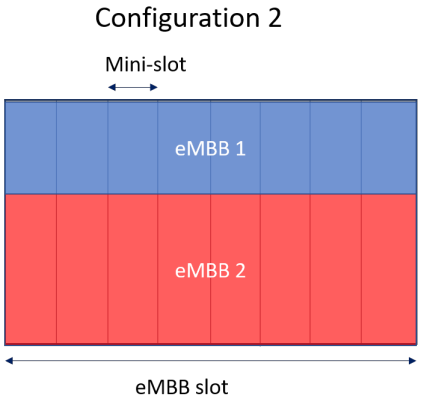


Figure 6.5: In this configuration, eMBB users share bandwidth and do not share the bandwidth in and eMBB slot.

channel states. Suppose that a naive eMBB scheduler ignores channel states and statically partitions the bandwidth between these users (symmetry implies half the bandwidth to each user). In this case, it is clear that an optimal URLLC scheduler needs to be both channel-state and eMBB aware – at each minislot, depending on the instantaneous demand and the channel states, it needs to puncture the two users' shares of bandwidths differently. For instance at a certain minislot, if one user has a really poor channel state, then the URLLC traffic in that minislot would be mostly loaded onto the frequency resources occupied by this user (as the total rate loss to eMBB traffic will be minimal).

In this section, we show a surprising result – if the eMBB scheduler is intelligent, then the URLLC scheduler can be *oblivious to the channel states, utility functions and the actual rate allocations of the eMBB scheduler.*

6.4.1 Characterization of capacity region

Let us consider the capacity region for a wireless system based on linear superposition/puncturing model under a restricted class of policies Π^{LR} that combine feasible eMBB allocations $\boldsymbol{\phi} \in \Sigma$ with random placement of URLLC demands across minislots. For any $\pi \in \Pi^{LR}$ with eMBB allocation $\boldsymbol{\phi}^{\pi}$ the mean induced loads for such randomization for each state $s \in \mathcal{S}$ and minislot $m \in \mathcal{M}$ will satisfy $l_{u,m}^{\pi,s} = \rho\phi_{u,m}^{\pi,s}$. Indeed randomization clearly leads to an induced loads that are proportional to the eMBB allocations on a per mini-slot basis, but also per eMBB slot, i.e., $l_u^{\pi,s} = \rho\phi_u^{\pi,s}$. Thus for our linear superposition/puncturing model we have that

$$r_u^{\pi,s} = \hat{r}_u^s(\phi_u^{\pi,s} - l_u^{\pi,s}) = \hat{r}_u^s\phi_u^{\pi,s}(1-\rho).$$

Hence the overall user rates achieved under such a policy are given by $\mathbf{c}^{\pi} = (c_u^{\pi}|u \in \mathcal{U})$ where

$$c_u^{\pi} = \sum_{s\in\mathcal{S}} \hat{r}_u^s\phi_u^{\pi,s}(1-\rho)p_S(s).$$

The capacity region associated with policies that use URLLC randomization is thus given by

$$\begin{aligned}\mathcal{C}^{LR} &= \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists\pi \in \Pi^{LR} \text{ s.t. } \mathbf{c} \leq \mathbf{c}^{\pi}\} \\ &= \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists\boldsymbol{\phi} \in \Sigma \text{ s.t. } \mathbf{c} \leq \mathbf{c}^{\boldsymbol{\phi}}\},\end{aligned}$$

where we have used abused notation by using $\mathbf{c}^{\boldsymbol{\phi}}$ to represent the throughput achieved by a policy π that uses eMBB resource allocation $\boldsymbol{\phi}$ and randomized URLLC demand placement. Finally note that for any fixed $\rho \in (0,1)$, $\mathcal{C}^{LR}$ is a closed and bounded convex region. This is because an affine map of a convex region remains convex;

hence multiplying the constraints on the capacity region defined by $\boldsymbol{\phi}$ by a constant $(1-\rho)$ preserves convexity of the rate region.

Theorem 6.4.1. *For a wireless system under the linear superposition/puncturing model we have that* $\mathcal{C} = \mathcal{C}^{LR}$.

The proof is deferred to the Appendix 6.10. In other words the throughput $\mathbf{c}^{\pi} \in \mathcal{C}$ achieved by any feasible policy $\pi \in \Pi$ can also be achieved by policy π', with a possibly different eMBB resource allocation policy than π but utilizing random placement of URLLC demands across mini-slots.

6.4.2 Utility maximizing joint scheduling

Given the result in Theorem 6.4.1 we now restate the utility maximization problem as optimizing solely over joint scheduling policies that use URLLC random placement policies, as below.

$$\begin{aligned} \max_{\phi \in \Sigma} \quad & \sum_{u \in \mathcal{U}} U_u(c_u^{\boldsymbol{\phi}}) \\ \text{s.t.} \quad & c_u^{\boldsymbol{\phi}} = \sum_{s \in \mathcal{S}} \hat{r}_u^s \phi_u^s (1-\rho) p_S(s), \quad \forall u \in \mathcal{U}. \end{aligned}$$

The above optimization problem has a strictly concave cost function, and convex constraints. Thus, at face-value, it appears that we can immediately apply the gradient scheduler introduced in [80], which is an online algorithm that converges and solves the optimization problem. This intuition is approximately correct, but subject to two modifications.

First, the setting in [80] has deterministic rates in each channel state. However, in our case, in each channel state, the rates are stochastic due to i.i.d. puncturing due to URLLC traffic (which accounts for the $(1-\rho)$ correction). This can be easily addressed by modifying the setting in [80]; the finite state and i.i.d. nature of puncturing implies that the proofs in [80] hold with minor modifications; we skip the details.

The second issue is somewhat more nuanced. In current wireless systems (e.g. LTE) and proposals for 5G systems, a slot is partitioned into a collection of Resource Blocks (RB), where each RB is a time-frequency rectangle (1 msec $\times$ 180 KHz in LTE). Importantly, these RBs can be individually allocated to different eMBB users. If we now apply the gradient scheduler in [80] to our setting, the result will be that all RBs in a slot will be allocated to the same user. While this is no-doubt asymptotically optimal, it seems intuitive that sharing RBs across users even within a slot will lead to better short-term performance. Indeed this intuition has been explored in the context of iterative MaxWeight algorithms to provide formal guarantees, see [81, 82]. The high level idea is that even within a slot, RB allocations are iterative, where future RB allocation need to account for prior rate allocations even within the same slot. This is formalized below, where we have fully described the joint eMBB-URLLC scheduler.

The URLLC scheduler: As explained in the previous section, the URLLC scheduler places the URLLC traffic uniformly at random over the minislots.

The eMBB scheduler: Let there be B resource blocks available for allocation every eMBB slot, indexed by $1, 2, \ldots, B$. Let $\overline{R}_u(t-1)$ be the random variable

denoting the average rate received by eMBB user up to eMBB slot $t-1$. In any eMBB slot t we schedule an user $u(b)$ in RB b such that

$$u(b) \in \text{argmax} \left\{ \hat{r}_u^s U_u^{'} \left(\overline{r}_u^{\epsilon} \left(b-1, t \right) \right), \, u = 1, 2, \ldots, \mathcal{U} \right\}, \tag{6.5}$$

where $\overline{r}_u^{\epsilon} \left(b-1, t \right)$ is an estimate of the average rate received by eMBB user u till slot t which is iteratively updated as follows:

$$\overline{r}_u^{\epsilon} \left(b, t \right) = \begin{cases} \overline{R}_u(t-1), & b = 0, \\ (1-\epsilon) \, \overline{r}_u^{\epsilon} \left(b-1, t \right) & \\ +\epsilon \left(\hat{r}_u^s \frac{1}{B} (1-\rho) \mathbb{1} \left(i = u(b) \right) \right), & b \neq 0. \end{cases} \tag{6.6}$$

In the above equation, ϵ is a small positive value. At the end of eMBB slot t, the eMBB scheduler receives feedback from the eMBB receivers indicating the actual rates received by the eMBB users due to allocations through (6.6). We denote this rate received eMBB user u in slot by the random variable $R_u(t)$. We finally update $\overline{R}_u(t)$ as follows:

$$\overline{R}_u(t) = (1-\epsilon) \, \overline{R}_u(t-1) + \epsilon R_u(t). \tag{6.7}$$

This update is analogous to the gradient algorithm [80] (see also iterative algorithms in [81,82]). The optimality proof of this algorithm follows (with minor modifications) from the analysis in [80]; we skip the details.

Remarks: (i) A natural decomposition of the joint eMBB+URLLC scheduling is now apparent. On one hand, the eMBB scheduler maximizes utilities based on the *expected* channel rates stemming from uniformly random puncturing of minislots (accounted for through the $(1-\rho)$ multiplicative factor), and does so using the iterative gradient scheduler. The URLLC scheduler, on the other-hand, is completely

agnostic to either the channel state or the actual eMBB allocations and simply punctures minislots based on the current instantaneous demand.

(ii) The fact that the URLLC traffic is completely agnostic to the channel state and eMBB utilities/allocation is surprising. Intuitively it seems plausible that one could load an eMBB user with a lower marginal utility with more URLLC traffic, while protecting a eMBB user with a higher marginal utility and achieve a better sum utility. Further, it seems reasonable that eMBB users with a worse channel state (and thus lower rate) could be loaded with additional URLLC traffic. However, Theorem. 6.4.1 implies that there exists an optimal solution that is achieved by channel and utility oblivious, uniform loading of URLLC traffic, thus providing a very simple algorithm for URLLC scheduling.

6.5 Convex Model – Time-Homogenous Policies

In this section we shall consider joint scheduling for wireless systems for a general superposition/puncturing model. This is a somewhat complex problem, whence we will focus our attention on a restricted, but still rich, class of scheduling policies which we refer to as time-homogeneous eMBB/URLLC schedulers. We identify a key concavity requirement in Condition 1 (that is satisfied by convex loss functions) that enables a stochastic approximation approach for utility maximization.

6.5.1 Time-homogeneous eMBB/URLLC Scheduling policies

We shall define time-homogeneous eMBB/URLLC schedulers as follows. First, feasible eMBB allocations $\boldsymbol{\phi} \in \Sigma$ will be restricted such that for any eMBB slot in

channel state $s \in \mathcal{S}$ allocations are *time-homogeneous* across minislots across the slot, i.e., $\phi^s_{u,1} = \phi^s_{u,m}, \forall m \in \mathcal{M}$ and its overall allocation for the slot is given by $\phi^s_u = |\mathcal{M}|\phi^s_{u,1}$. The set of time-homogeneous eMBB allocations is thus given by

$$\Sigma^U \quad := \quad \left\{\mathbf{x} \in \Sigma \mid \forall s \in \mathcal{S}, u \in \mathcal{U},\ x^s_{u,m} = x^s_{u,1} \ \ \forall m \in \mathcal{M}\right\}.$$

Second, URLLC demand placement per minislot are done proportionally to pre-specified weights, and these weights are assumed to be time-homogeneous across minislots. In particular such policies are parametrized by a weight matrix $\gamma \in \Sigma^U$, where induced load on user u under channel state s and slot m is given by

$$L^s_{u,m} = \frac{\gamma^s_{u,m}}{\sum_{u' \in \mathcal{U}} \gamma^s_{u',m}} D(m) = \frac{\gamma^s_{u,1}}{f} D(m).$$

The eMBB and URLLC allocations are however coupled together since it must be the case that for all $u \in \mathcal{U}$ $L^s_{u,m} \leq \phi^s_{u,m} = \phi^s_{u,1}$ almost surely, i.e., one can not induce more superposition/puncturing on a user than the resources it has been allocated on that slot. so the following condition must be satisfied. Thus we must have that for all $u \in \mathcal{M}$

$$D(m) \leq \min_{u \in \mathcal{U}} \frac{\phi^s_{u,1}}{\gamma^s_{u,1}} f.$$

Note we have assumed that $D(m) \leq f$ almost surely, thus if $\frac{\phi^s_{u,1}}{\gamma^s_{u,1}} \geq 1$ this may not hold.

Assumption 5. *We say a system satisfies a $(1-\delta)$ URLLC sharing factor per minislot if $D(m) \leq f(1-\delta)$ almost surely for all $m \in \mathcal{M}$.*

Under a $(1-\delta)$ URLLC demand backoff a time-homogeneous eMBB resource allocation $\boldsymbol{\phi}$ and URLLC allocation $\boldsymbol{\gamma}$ is will be feasible if for all $s \in \mathcal{S}$ we have

$$(1-\delta) \leq \min_{u \in \mathcal{U}} \frac{\phi_{u,1}^s}{\gamma_{u,1}^s},$$

which is satisfied as long as $(1-\delta)\gamma_{u,1}^s \leq \phi_{u,1}^s$ for all $u \in \mathcal{U}$. This motivates the following definition.

Definition *Under a $(1-\delta)$ sharing factor, the feasible time-homogeneous eMBB/URLLC scheduling policies are parameterized by $\boldsymbol{\phi}, \boldsymbol{\gamma} \in \Sigma^U$ such that $(1-\delta)\boldsymbol{\gamma} \leq \boldsymbol{\phi}$. We shall denote the set of such policies as follows:*

$$\Pi^{U,\delta} := \{(\boldsymbol{\phi}, \boldsymbol{\gamma}) \mid \boldsymbol{\phi}, \boldsymbol{\gamma} \in \Sigma^U \text{ and } (1-\delta)\boldsymbol{\gamma} \leq \boldsymbol{\phi}\},$$

where $\Pi^{U,\delta}$ is a convex set.

6.5.2 Characterization of throughput region

In this section we characterize the throughput regions achievable under time-homogeneous scheduling.

Theorem 6.5.1. *Under a $(1-\delta)$ sharing factor and time-homogeneous scheduler $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{U,\delta}$ the probability of induced throughput for user r $u \in \mathcal{U}$ in channel state $s \in \mathcal{S}$ is given by*

$$r_u^{\boldsymbol{\pi},s} = \mathrm{E}[f_u^s(\phi_u^{\boldsymbol{\pi},s}, \gamma_u^{\boldsymbol{\pi},s} D)],$$

and the overall user throughputs are given by $\mathbf{c}^{\boldsymbol{\pi}} = (c_u^{\boldsymbol{\pi}} : u \in \mathcal{U})$ where $c_u^{\boldsymbol{\pi}} = \sum_{u \in \mathcal{U}} r_u^{\boldsymbol{\pi},s} p_S(s)$.

The proof is available in Appendix 6.11. Based on the above we can define feasible throughput region constrained to the time-homogeneous policies in $\Pi^{U,\delta}$. First let us define

$$\mathcal{C}^{U,\delta} \quad = \quad \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists \boldsymbol{\pi} \in \Pi^{U,\delta} \text{ s.t. } \mathbf{c} \leq \mathbf{c}^{\boldsymbol{\pi}}\}.$$

We shall let $\hat{\mathcal{C}}^{U,\delta}$ denote the convex hull of $\mathcal{C}^{U,\delta}$. Note that throughputs rates in the convex hull are achievable through policies that do time sharing/randomization amongst time-homogeneous scheduling policies in $\Pi^{U,\delta}$.

Condition 1. *For all $s \in \mathcal{S}$ and $u \in \mathcal{U}$ the functions $g_u^s(,)$ given by*

$$g_u^s(\phi_u^s, \gamma_u^s) = \mathrm{E}[f_u^s(\phi_u^s, \gamma_u^s D)], \tag{6.8}$$

are jointly concave on $\Pi^{U,\delta}$.

Lemma 6.5.2. *Condition 1 is satisfied for systems where superposition/puncturing of each user is modelled via either a*

1. *Convex loss function,*
2. *Threshold loss function with fixed relative thresholds, i.e., $t_u^s(\phi_u^s) = \alpha_u^s$ for $\phi \in [0,1]$ and the URLLC demand distribution $F_D(\cdot)$ is such that $F_D(\frac{1}{x})$ is concave in x (satisfied by the truncated Pareto distribution).*

The proof is available in Appendix 6.11. With this condition in place, we now describe the throughput region.

Theorem 6.5.3. *Suppose that Condition 1 holds. then $\mathcal{C}^{U,\delta} = \hat{\mathcal{C}}^{U,\delta}$, i.e., there is no need to consider time-sharing/randomization amongst time-homogeneous eMBB/URLLC policies.*

The proof is available in Appendix 6.11. Thus, with time-homogeneous policies and imposing concavity of from Condition 1, the above result sets up a convex optimization problem in (ϕ, γ), i..e, we have a concave cost function with convex constraints. Thus, by iteratively updating (ϕ, γ), we can develop an online algorithm that asymptotically maximizes utility. Below, we formally develop a stochastic approximation algorithm to achieve this objective.

6.5.3 Stochastic approximation based online algorithm

We first restate the utility maximization problem for time-homogeneous URLLC/eMBB scheduling policies:

$$\max_{\phi,\gamma \in \Pi^{U,\delta}} \quad \sum_{u \in \mathcal{U}} U_u \left(\sum_{s \in \mathcal{S}} p_{\mathcal{S}}(s) g_u^s \left(\phi_u^s, \gamma_u^s \right) \right). \tag{6.9}$$

Observe that the objective function consists of a sum of compositions of non-decreasing concave function $(U_u(\cdot))$, and supposing Condition 1 holds, a concave function $g_u^s(\cdot, \cdot)$ in ϕ and γ. Further, the constraint set is convex. Therefore, the above problem fits in the framework of standard convex optimization problems. However, solving the above problem requires the knowledge of all possible network states and its probability distribution, resulting in an *offline* optimization problem. In this section, we develop a stochastic approximation based online algorithm to solve the above problem.

Online algorithm: Let $\overline{R}_u(t-1)$ be the random variable denoting the average rate received by eMBB user up to eMBB slot $t-1$. Let s be the network state in slot t. Define vectors $\phi^s := \{\phi^s_u, \mid u \in \mathcal{U}\}$ and $\gamma^s := \{\gamma^s_u \mid u \in \mathcal{U}\}$. At the beginning of eMBB slot t, we compute the vectors $\left(\tilde{\phi}(t), \tilde{\gamma}(t)\right)$ as the solution to the following optimization problem.

$$\max_{\phi^s, \gamma^s} \quad \sum_{u \in \mathcal{U}} U_u^{'}\left(\overline{R}_u(t-1)\right) g_u^s(\phi_u^s, \gamma_u^s), \tag{6.10}$$

$$\text{s.t.} \quad \phi^s \geq (1-\delta)\,\gamma^s, \tag{6.11}$$

$$\sum_{u \in \mathcal{U}} \phi_u^s = 1 \text{ and } \sum_{u \in \mathcal{U}} \gamma_u^s = 1, \tag{6.12}$$

$$\phi^s \in [0,1]^{|\mathcal{U}|} \text{ and } \gamma^s \in [0,1]^{|\mathcal{U}|}. \tag{6.13}$$

This optimization problem is a convex optimization problem and can be solved numerically using standard convex optimization techniques. Using $\left(\tilde{\phi}(t), \tilde{\gamma}(t)\right)$, we schedule URLLC and eMBB traffic as follows:

The eMBB scheduler: For notational ease, we fluidize the bandwidth. Specifically, we assume that the bandwidth of a resource block is very small when compared to the total bandwidth available. Hence, the bandwidth can be split into arbitrary fractions and we allocate $\tilde{\phi}_u(t)$ fraction of the total bandwidth to eMBB user u.

The URLLC Scheduler: We load different eMBB users with URLLC traffic according to the vector $\tilde{\gamma}(t)$.

At the end of eMBB slot t, the eMBB scheduler receives feedback from the eMBB receivers indicating the rates received by the eMBB users. Let us denote the

rate received eMBB user u in slot by the random variable $R_u(t)$. We update $\overline{R}_u(t)$ as follows:

$$\overline{R}_u(t) = (1 - \epsilon_t)\,\overline{R}_u(t-1) + \epsilon_t R_u(t), \tag{6.14}$$

where $\{\epsilon_t \mid t = 1, 2, 3, \ldots\}$ is a sequence of positive numbers which satisfy the following (standard) condition:

Condition 2. *The averaging sequence $\{\epsilon_t\}$ satisfies:*

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \quad \textit{and} \quad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

.

Finally, we state the main result of this section, which is the optimality of the stochastic approximation based online algorithm.

Theorem 6.5.4. *Let $\mathbf{r}^*$ be the optimal average rate vector received by eMBB users under the solution to the offline optimization problem. Suppose that Conditions 1 and 2 hold. Then we have that:*

$$\lim_{t\to\infty} \overline{\mathbf{R}}(t) = \mathbf{r}^* \quad \textit{almost surely.} \tag{6.15}$$

The proof is available in the Appendix 6.11.

6.6 Threshold Model and Placement Policies

In the previous section, we developed a stochastic approximation algorithm for time-homogeneous policies. This algorithm iteratively solves an optimization

problem described in (6.43). This optimization problem jointly optimizes over a pair of row vectors (ϕ^s, γ^s). While this convex optimization problem can be solved using standard methods, it could become computationally challenging as the number of users scale up.

In this section, we shall restrict our attention to a threshold model for superposition/puncturing, and look at policies that impose structural conditions on the puncturing matrix γ. We will show that the resulting class of policies have nice theoretical properties that lead to simpler online algorithms (solving (6.5), which is an one-dimensional search).

We consider two types of structural conditions on the puncturing matrix γ, resource proportional and threshold proportional placement policies, described below.

(i) Resource Proportional (RP) Placement: The first is based on allocating URLLC demands in proportion to eMBB user slot allocations, i.e., $\gamma_u^s = \phi_u^s$. We refer to this as Resource Proportional (RP) Placement and denote such policies by

$$\Pi^{RP,\delta} := \{(\boldsymbol{\phi}, \boldsymbol{\gamma}) \in \Pi^{U,\delta} \mid \boldsymbol{\gamma} = \boldsymbol{\phi}\},$$

and define the associated achievable throughput region

$$\mathcal{C}^{RP,\delta} \quad = \quad \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists \boldsymbol{\pi} \in \Pi^{RP,\delta} \text{ s.t. } \mathbf{c} \leq \mathbf{c}^{\boldsymbol{\pi}}\}.$$

The motivation for RP Placement comes from the optimality of random placement for the linear model in Section 6.4. Observe that if puncturing occurs uniformly randomly, then the expected number of punctures is directly proportional to the

fraction of bandwidth allocated to an eMBB user. Thus, RP Placement has the interpretation of a *determinized version* of the policy we previously studied with linear loss functions.

(ii) Threshold Proportional (TP) Placement: The second policy allocates URLLC demands in proportion to the eMBB users associated loss thresholds so as to avoid losses,

$$\gamma_u^s = \frac{\phi_u^s t_u^s(\phi_u^s)}{\sum_{u' \in \mathcal{U}} \phi_{u'}^s t_{u'}^s(\phi_{u'}^s)}.$$

We refer to this as Threshold Proportional (TP) Placement and denote such policies by

$$\Pi^{TP,\delta} := \\ \{(\boldsymbol{\phi}, \boldsymbol{\gamma}) \in \Pi^{U,\delta} \mid \gamma_u^s = \frac{\phi_u^s t_u^s(\phi_u^s)}{\sum_{u' \in \mathcal{U}} \phi_{u'}^s t_{u'}^s(\phi_{u'}^s)} \forall s \in \mathcal{S}, u \in \mathcal{U}\}.$$

The associated achievable throughput region is denoted

$$\mathcal{C}^{TP,\delta} \quad = \quad \{\mathbf{c} \in \mathbb{R}_+^{|\mathcal{U}|} \mid \exists \boldsymbol{\pi} \in \Pi^{TP,\delta} \text{ s.t. } \mathbf{c} \leq \mathbf{c}^{\boldsymbol{\pi}}\}.$$

The following theorem provides a formal motivation for TP Placement,. The main takeaway here is that the *probability of any loss in an eMBB slot under TP Placement policy is a lower bound over all other strategies.*

Theorem 6.6.1. *Consider a system with* $(1-\delta)$ *sharing factor. Consider a joint scheduling policy based on the TP URLLC placement i.e,* $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{TP,\delta}$. *Then* $\boldsymbol{\pi}$ *achieves the minimum probability of eMBB loss amongst all joint scheduling policies using the same eMBB resource allocation* $\boldsymbol{\phi}^{\boldsymbol{\pi}}$.

The proofs (along with characterizations of the capacity region for RP and TP Placement policies) are available in Appendix 6.11.1.

6.6.1 Online scheduling for RP and TP Placement

In this section, we consider online algorithms that implement the RP and TP Placement policies. While the stochastic approximation algorithm developed in Section 6.5.3 can clearly be used, the additional structure imposed by the RP and TP Placement policies, and the shape of the threshold loss function (discussed below) can result in much simpler algorithms (with optimality guarantees).

We consider the case where $t_u^s(\phi)$ is a (state dependent but ϕ independent) constant, i.e., $t_u^s(\phi) = \alpha^s$, where $\alpha^s \in (0, 1)$. Intuitively, this means that eMBB traffic which has a higher share of the bandwidth is more resilient to losses (e.g. through coding over larger fraction of resources). Then, by substituting this loss function in (6.32) and (6.35) (where we also use the fact that $\sum_{u \in \mathcal{U}} \phi_u^s = 1$), we have that

$$r_u^{\boldsymbol{\pi},s} = \hat{r}_u^s \phi_u^s F_D(\alpha^s).$$

Comparing with the development in Section 6.4.2, we observe that the cost and constraints are identical if $F_D(\alpha^s)$ replaces $(1 - \rho)$. Note that a small difference is that $F_D(\alpha^s)$ is state and user dependent, whereas $(1 - \rho)$ does not depend on either; however, it is easy to see that the development in Section 6.4.2 immediately generalizes to this setting. Hence, we can interpret $F_D(\alpha^s)$ as the state and user dependent average rate loss due to puncturing via the RP or TP Placement policies.

We can now employ the rate-based iterative gradient scheduler developed

in Section 6.4.2 (by replacing $(1-\rho)$ in (6.6) by a user-dependent $F_D(\alpha^s)$), and the theoretical guarantees directly carry over. As this algorithm only minimizes over users at each slot in (6.5), this is easier to implement when compared to the stochastic approximation algorithm developed in Section 6.5.3.

6.7 Optimality of Mini-slot Homogeneous Policies

In this section we derive conditions under which *mini-slot homogeneous* URLLC placement polices are optimal.

With slight abuse of notation, we introduce the following additional assumption on loss function $(h_u^s(\cdot))$.

Assumption 6. *Let the total URLLC demand in an eMBB slot be d and γd be the total URLLC puncturing on eMBB user u where $\gamma \in [0,1]$, then for any $\phi \in [0,1]$ $h_u(\cdot,\cdot)$ can be split as follows:*

$$h_u^s\left(\frac{\gamma d}{\phi}\right) = f(d)\tilde{h}_u^s\left(\frac{\gamma}{\phi}\right), \tag{6.16}$$

where $f(\cdot)$ is a non-zero non-decreasing function, and $\tilde{h}_u^s(\cdot)$ is a non-decreasing convex function.

We shall first state the following definitions.

Definition A scheduler is said to be *non-anticipative* and *causal* if at the beginning of a mini-slot m, 1) scheduler knows the realizations of D_1, $D_2, \ldots, D_{m-1}$ and 2), scheduler is unaware of the realization of D_m, but knows only its distribution.

Definition A scheduling policy is said to be *mini-slot dependent* if the URLLC placement policy can vary with the mini-slot index m in an eMBB slot.

We shall describe a non-anticipative, causal, and mini-slot dependent joint scheduling policy π.

1. At the beginning of an eMBB slot, the scheduler chooses $\phi_u^{s,\pi}, u \in \mathcal{U}$ such that

$$\sum_{u \in \mathcal{U}} \phi_u^{s,\pi} = 1 \text{ and } \phi_u^{s,\pi} \in [0,1] \quad \forall u. \tag{6.17}$$

2. In each mini-slot m, the total puncturing on eMBB user u is given by $\gamma_u^{s,\pi}\left(m, \mathbf{D}(m-1)\right) D_m$, where $\gamma_u^{s,\pi}\left(\cdot,\cdot\right)$ is the URLLC placement factor, $\mathbf{D}(m-1) := [D_1, D_2, \ldots, D_{m-1}]$ is the vector of URLLC demands till mini-slot $m-1$ in a given eMBB slot. For any m and $\mathbf{d}$, $\gamma_u^{s,\pi}\left(m, \mathbf{d}\right)$ has to satisfy the following constraints.

$$\sum_{u \in \mathcal{U}} \gamma_u^{s,\pi}(m, \mathbf{d}) = 1, \quad \gamma_u^{s,\pi}(m, \mathbf{d}) \in [0,1]\,, \tag{6.18}$$

$$\gamma_u^{s,\pi}(m, \mathbf{d}) \leq \frac{\phi_u^{s,\pi}}{|\mathcal{M}|\,(1-\delta)}. \quad \forall u \in \mathcal{U}. \tag{6.19}$$

Observe that the URLLC placement factor for non-anticipative, causal, and mini-slot dependent scheduling policy is a function of both the mini-slot index and the past URLLC demands. Let $\tilde{\Pi}$ be the set of all non-anticipative, causal, and mini-slot dependent scheduling policies. For any eMBB slot t, we would like find the policy which solves the following optimization problem.

$$\mathcal{OP}_1: \quad \max_{\pi \in \tilde{\Pi}} : \sum_{u \in \mathcal{U}} w_u g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}\left(\cdot,\cdot\right)\right), \tag{6.20}$$

where s is the current network state and $g_u^{s,\pi}(\cdot,\cdot)$ is the average rate experienced by eMBB user u under policy π. $g_u^{s,\pi}(\cdot,\cdot)$ is given by the following expression:

$$g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}(\cdot,\cdot)\right) := r_u^s \phi_u^{s,\pi} \mathbb{E}\left[1 - h_u^s\left(\sum_{m=1}^{|\mathcal{M}|} \gamma_u^{s,\pi}\left(m, \mathbf{D}(m-1)\right) D_m, \phi_u^{s,\pi}\right)\right], \quad (6.21)$$

where the expectation is computed with respect to the joint distribution of D_1, $D_2, \ldots, D_{|\mathcal{M}|}$.

The main result on the optimality of mini-slot homogeneous policies is stated below.

Theorem 6.7.1. *Under Assumptions 6, there exists an optimal solution* $(\phi^{s,*}, \gamma^{s,*}(\cdot,\cdot))$ *for* $\mathcal{OP}_1$ *with a mini-slot homogeneous URLLC placement policy.*

The following corollary directly follows from the previous theorem.

Corollary 6.7.2. *There exists an optimal mini-slot homogeneous policy when* $h_u^s(\cdot)$ *satisfies:*

1. **Linear:** $h_u^s(\frac{\gamma D}{\phi}) = k_u^s\left(\frac{\gamma D}{\phi}\right)$, *where* $k_u^s \geq 0$.
2. **Monomial:** $h_u^s(\frac{\gamma D}{\phi}) = k_u^s\left(\frac{\gamma D}{\phi}\right)^q$ *where* $k_u^s \geq 0$ *and* $q \geq 0$.

6.8 Simulations

We consider a system with a total of 100 RBs available per eMBB slot, with 8 minislots per eMBB slot. In an eMBB slot, $\hat{r}_u^s$ for an eMBB user is drawn from

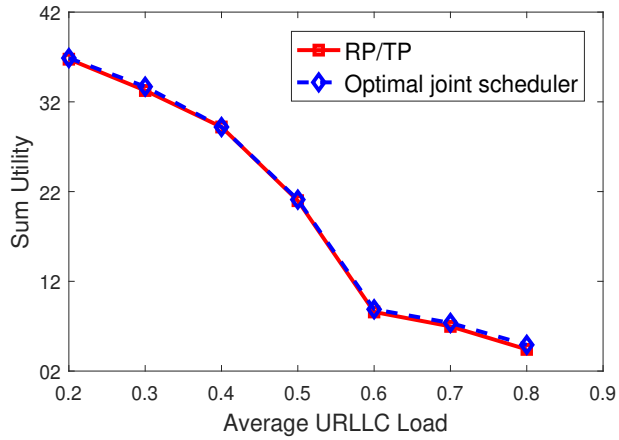


Figure 6.6: Sum utility as a function of URLLC load ρ for the optimal and TP Placement policies under threshold model ($\delta = 0.1$).

the finite set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ Mbps with equal probability and i.i.d. across users and slots. Our system consists of 20 users, and with 100 channel states (all equally likely). The (20 users $\times$ 100 states) rate matrix is one-time synthesized by independently and uniformly sampling a rate from the finite rate set for each matrix element.

We first consider a threshold model with $\alpha^s = 0.3$ for 50% of eMBB states and $\alpha^s = 0.7$ for the rest. We use the utility function $U_u(r) = \log(r) + 6.5$ for all eMBB users, where r is measured in Mbps (constant added to ensure non-negativity of the sum utility). URLLC load in an eMBB slot (D) is generated form the truncated Pareto distribution with tail exponent $\eta = 2$. We compare the optimal policy (stochastic approximation algorithm, see Section 6.5.3) with that from the TP Placement policy (the simpler gradient algorithm in Section 6.6.1). In this case, as the threshold functions are (state-dependent) constants, the RP and TP Placement poli-

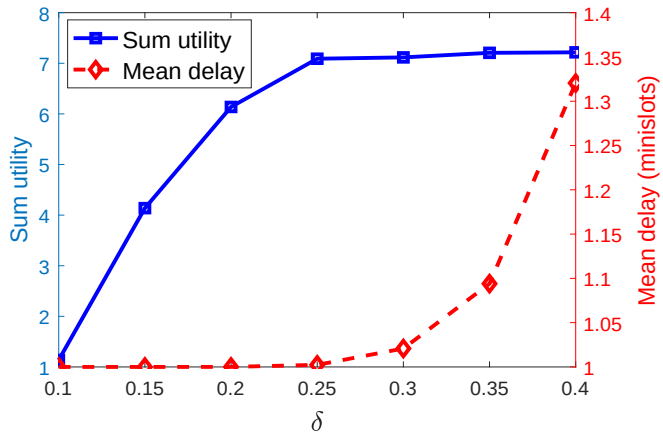


Figure 6.7: Sum utility and mean URLLC delay as a function of δ.

cies are the same. As we can see in Figure 6.6, the TP Placement policy tracks the optimal policy very well.

In Figure 6.8, we study the trade-off between achieving a higher eMBB utility and lowering the mean delay of URLLC traffic for different values of the sharing factor $1-\delta$. Figure 6.8 plots the corresponding probability that the URLLC traffic delay exceeds two minislots ($0.125 \times 2 = 0.25$ msec). To study this trade-off we generate URLLC arrivals in each minislot from an uniform distribution between $[0, 1/8]$ (recall there are 8 minislots). In each minislot, we can serve at most $\frac{1-\delta}{8}$ units of URLLC traffic. If the URLLC load in a given minislot is more than $\frac{1-\delta}{8}$, the remaining URLLC traffic is queued and served in the next minislot on a FCFS basis. For the eMBB users we use a convex model with $h_u^s(s) = e^{\kappa_u(x-1)}$ where κ_u determines the sensitivity of an eMBB user to an URLLC load. We have chosen $\kappa = 0.2$ for 50 % of the users and $\kappa = 0.7$ for the rest. We also set $\forall u\ U_u(x) = \log(x) + 4.2$ (constant added to ensure positive sum utility). In summary, a larger value of δ

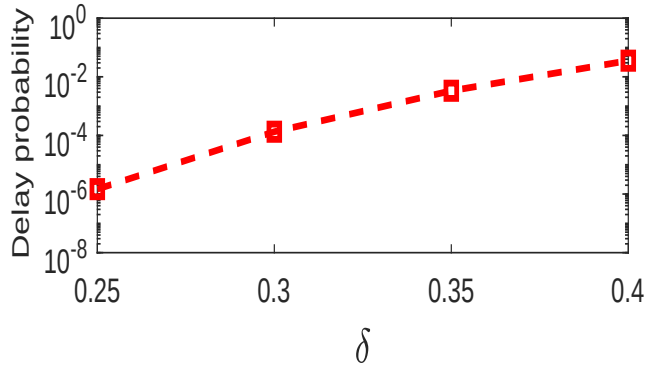


Figure 6.8: Log-scale plot of the probability that URLLC traffic is delayed by more than two minislots (0.25 msec) for various values of δ.

limits the amount of URLLC traffic than can be served in a minislot. However, a larger δ enlarges the constraint set $\Pi^{U,\delta}$ in the eMBB utility maximization problem, and hence we get higher eMBB utility.

6.9 Conclusion

In this chapter, we have developed a framework and algorithms for joint scheduling of URLLC (low latency) and eMBB (broadband) traffic in emerging 5G systems. Our setting considers recent proposals where URLLC traffic is dynamically multiplexed through puncturing/superposition of eMBB traffic. Our results show that this joint problem has structural properties that enable clean decompositions, and corresponding algorithms with theoretical guarantees.

Appendix

6.10 Proofs from Section 6.4

Theorem 6.4.1. *For a wireless system under the linear superposition/puncturing model we have that* $\mathcal{C} = \mathcal{C}^{LR}$.

Proof. Clearly since $\Pi^{LR} \subset \Pi$ we have that $\mathcal{C}^{LR} \subset \mathcal{C}$

Now consider any policy $\pi \in \Pi$ with eMBB user allocations $\boldsymbol{\phi}^{\pi}$ and URLLC loads $\mathbf{l}^{\pi}$ and associated long term throughput is $\mathbf{c}^{\pi}$ given by

$$c_u^{\pi} = \sum_{s \in \mathcal{S}} \hat{r}_u^s (\phi_u^{\pi,s} - l_u^{\pi,s}) p_S(s).$$

Let us define a π' based on π to have per mini-slot eMBB user allocations given by

$$\phi_{u,m}^{\pi',s} = \frac{\phi_u^{\pi,s} - l_u^{\pi,s}}{\sum_{u' \in \mathcal{U}} \phi_{u'}^{\pi,s} - l_{u'}^{\pi,s}} f = \frac{\phi_u^s - l_u^{\pi,s}}{1-\rho} f,$$

for $s \in \mathcal{S}$, $u \in \mathcal{U}$ and $m \in \mathcal{M}$. Since induced mean loads on an eMBB user can not exceed its allocation we have that $\boldsymbol{\phi}^{\pi} \geq \mathbf{l}^{\pi}$ so the above allocations are positive. Note also that this allocation is not mini-slot dependent, but normalized so that per mini-slot they sum to f and over the whole eMBB slot sum to 1, i.e., $\boldsymbol{\phi}^{\pi'} \in \Sigma$. Thus for such an allocation we have that

$$\phi_u^{\pi',s} = \frac{\phi_u^s - l_u^{\pi,s}}{1-\rho}.$$

Also suppose that π' uses randomized URLLC placement across mini-slots which induces mean URLLC loads proportional to the allocations, i.e., $l_u^{\pi',s} = \rho\phi_u^{\pi',s}$. It follows that

$$\begin{aligned}\phi_u^{\pi',s} - l_u^{\pi',s} &= \phi_u^{\pi',s} - \rho\phi_u^{\pi',s} \\ &= (1-\rho)\phi_u^{\pi',s} \\ &= \phi_u^{\pi,s} - l_u^{\pi,s},\end{aligned}$$

and so $c_u^{\pi,s} = c_u^{\pi',s}$ for all $s \in \mathcal{S}$ and $u \in \mathcal{U}$. Thus for any policy π there is a policy π' which uses randomized URLLC placement and achieves the same long term throughputs. It follows that $\mathcal{C} \subset \mathcal{C}^{LR}$ and so $\mathcal{C} = \mathcal{C}^{LR}$. □

6.11 Proofs from Section 6.5

Theorem 6.5.1. *Under a $(1-\delta)$ sharing factor and time-homogeneous scheduler $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{U,\delta}$ the probability of induced throughput for user r $u \in \mathcal{U}$ in channel state $s \in \mathcal{S}$ is given by*

$$r_u^{\boldsymbol{\pi},s} = \mathrm{E}[f_u^s(\phi_u^{\boldsymbol{\pi},s}, \gamma_u^{\boldsymbol{\pi},s} D)].$$

and the overall user throughputs are given by $\mathbf{c}^{\boldsymbol{\pi}} = (c_u^{\boldsymbol{\pi}} : u \in \mathcal{U})$ where $c_u^{\boldsymbol{\pi}} = \sum_{u\in\mathcal{U}} r_u^{\boldsymbol{\pi},s} p_S(s)$.

Proof. Under a policy $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{U,\delta}$ we have that the induced loads are given by

$$L_{u,m}^{\boldsymbol{\pi},s} = \frac{\gamma_{u,1}^{\boldsymbol{\pi},s}}{f} D(m),$$

so we have that

$$L_u^{\boldsymbol{\pi},s} = \sum_{u\in\mathcal{U}} L_{u,m}^{\boldsymbol{\pi},s} = \frac{\gamma_{u,1}^{\boldsymbol{\pi},s}}{f}\sum_{u\in\mathcal{U}} D(m) = \frac{\gamma_{u,1}^{\boldsymbol{\pi},s}}{f} D = \gamma_u^{\boldsymbol{\pi},s} D.$$

where the last equality follows from the uniformity of URLLC splits and normalization it follows that

$$r_u^{\boldsymbol{\pi},s} = \mathrm{E}[f_u^s(\phi_u^{\boldsymbol{\pi},s}, L_u^{\boldsymbol{\pi},s})] = \mathrm{E}[f_u^s(\phi_u^{\boldsymbol{\pi},s}, \gamma_u^{\boldsymbol{\pi},s} D)].$$

□

Lemma 6.5.2. *Condition 1 is satisfied for systems where superposition/puncturing of each user is modelled via either a*

1. *convex loss function,*
2. *threshold-based loss function with fixed relative thresholds, i.e.,* $t_u^s(\phi_u^s) = \alpha_u^s$ *for* $\phi \in [0.1]$ *and the URLLC demand distribution* F_D *is such that* $F_D(\frac{1}{x})$ *is concave in* x *(satisfied by the truncated Pareto distribution).*

Proof. Recall that convex loss functions are specified as follows

$$f_u^s(\phi_u^s, l_u^s) = \hat{r}_u^s \phi_u^s (1 - h_u^s(\frac{l_u^s}{\phi_u^s})),$$

with $h_u^s : [0,1] \to [0,1]$ a convex increasing function. For time-homogenous policies we have defined

$$\begin{aligned} g_u^s(\phi_u^s, \gamma_u^s) &= \mathrm{E}[f_u^s(\phi_u^s, \gamma_u^s D)] \\ &= \hat{r}_u^s E[\phi_u^s - \phi_u^s h_u^s(\frac{\gamma_u^s}{\phi_u^s} D)]. \end{aligned}$$

Recall that convex function $h()$ one can define a function $l(\phi, \gamma) = \phi h(\frac{\gamma}{\phi})$ known as the perspective of $h()$ which is known to be jointly convex in its arguments. It follows that $\phi - \phi h(\frac{\gamma}{\phi})$ is jointly concave, and so is $g_u^s()$ since it is a weighted aggregation of jointly concave functions.

For threshold-based loss functions where $t_u^s(\phi_u^s) = \alpha_u^s$ we have that

$$\begin{aligned} g_u^s(\phi_u^s, \gamma_u^s) &= \mathrm{E}[f_u^s(\phi_u^s, \gamma_u^s D)] \\ &= \hat{r}_u^s \phi_u^{\pi,s} P(\gamma_u^s D \leq \phi_u^{\pi,s} \alpha_s^u) \\ &= \hat{r}_u^s \phi_u^{\pi,s} F_D(\frac{\phi_u^{\pi,s} \alpha_s^u}{\gamma_u^s}). \end{aligned}$$

Now using the same result on the perspective functions of variables the result follows. The truncated Pareto case can be easily verified by taking derivatives. □

Theorem 6.5.3. *Suppose that Condition 1 holds, then $\mathcal{C}^{U,\delta} = \hat{\mathcal{C}}^{U,\delta}$, i.e., there is no need to consider time-sharing/randomization amongst time-homogeneous eMBB/URLLC policies.*

Proof. Clearly $\mathcal{C}^{U,\delta} \subset \mathcal{C}^{U,\delta}$. We will show that $\mathbf{c} \in \hat{\mathcal{C}}^{U,\delta}$ then their exists $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{U,\delta}$ such that $\mathbf{c} \leq \mathbf{c}^{\boldsymbol{\pi}}$ from which it follows that $\mathcal{C}^{U,\delta} \subset \mathcal{C}^{U,\delta}$.

Suppose $\mathbf{c} \in \hat{\mathcal{C}}^{U,\delta}$, then it can be represented as a convex combination of policies $\Pi^{U,\delta}$, in each channel state. For example suppose for simplicity that for that in channel state $s \in \mathcal{S}$ we have that $\lambda \in [0, 1]$ one time shares between two policies $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ to achieve throughput for $u \in \mathcal{U}$ given by

$$r_u^s = \lambda r_u^{\boldsymbol{\pi}_1,s} + (1 - \lambda) r_u^{\boldsymbol{\pi}_2,s}.$$

Consider u we have

$$\begin{aligned} r_u^s &= \lambda r_u^{\boldsymbol{\pi}_1,s} + (1-\lambda) r_u^{\boldsymbol{\pi}_2,s} \\ &= \lambda g_u^s(\phi_u^{\boldsymbol{\pi}_1,s}, \gamma_u^{\boldsymbol{\pi}_1,s}) + (1-\lambda) g_u^s(\phi_u^{\boldsymbol{\pi}_2,s}, \gamma_u^{\boldsymbol{\pi}_2,s}) \\ &\leq g_u^s(\lambda \phi_u^{\boldsymbol{\pi}_1,s} + (1-\lambda)\phi_u^{\boldsymbol{\pi}_2,s},\ \lambda \gamma_u^{\boldsymbol{\pi}_1,s} + (1-\lambda)\phi_u^{\gamma_2,s}) \\ &= g_u^s(\phi_u^{\boldsymbol{\pi},s}, \gamma_u^{\boldsymbol{\pi},s}), \end{aligned}$$

where $\phi_u^{\boldsymbol{\pi},s} = \lambda \phi_u^{\boldsymbol{\pi}_1,s} + (1-\lambda)\phi_u^{\boldsymbol{\pi}_2,s}$ and $\gamma_u^{\boldsymbol{\pi},s} = \lambda \gamma_u^{\boldsymbol{\pi}_1,s} + (1-\lambda)\gamma_u^{\boldsymbol{\pi}_2,s}$. Clearly $\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}$ as given above correspond to a policy $\boldsymbol{\pi}$ such that $\boldsymbol{\pi} \in \Pi^{U,\delta}$ since the set is convex. It also follows that $r_u^s \leq r_u^{\boldsymbol{\pi},s}$, so $c_u^s \leq c_u^{\boldsymbol{\pi},s}$ and so $\mathbf{c} \leq \mathbf{c}^{\boldsymbol{\pi}}$. □

Theorem 6.5.4. *Let* $\mathbf{r}^*$ *be the optimal average rate vector received by eMBB users under the solution to the offline optimization problem. Suppose that Conditions 1 and 2 hold. Then we have that:*

$$\lim_{t\to\infty} \overline{\mathbf{R}}(t) = \mathbf{r}^* \quad \textit{almost surely.} \tag{6.22}$$

The proof requires intermediate lemmas, detailed below. For the ease of exposition, let us define $U(\mathbf{r}) := \sum_{u\in\mathcal{U}} U_u(r_u)$ and

$$\nabla U(\mathbf{r}) := \left[\frac{\partial U_1(x)}{\partial x}\Big|_{x_1=r_1}, \frac{\partial U_2(x)}{\partial x}\Big|_{x_2=r_2}, \ldots, \frac{\partial U_1(x)}{\partial x}\Big|_{r_{x_{|\mathcal{U}|}}=|\mathcal{U}|} \right]^T$$

. First we have the following important lemma regarding the stochastic approximation algorithm.

Lemma 6.11.1. $\mathbf{R}(t) = \left[R_1(t), R_2(t), \ldots, R_{|\mathcal{U}|}\right]^T$ *is an unbiased estimator of* $\underset{\mathbf{c}\in\mathcal{C}^{\mathcal{U},\delta}}{\textit{argmax:}}\ \nabla U\left(\overline{\mathbf{R}}(\mathbf{t})\right)^T \mathbf{c}$, *i.e.,*

$$\mathbb{E}\left[\mathbf{R}(t)\right] = \underset{\mathbf{c}\in\mathcal{C}^{\mathcal{U},\delta}}{\textit{argmax:}}\ \nabla U\left(\overline{\mathbf{R}}(\mathbf{t})\right)^T \mathbf{c}. \tag{6.23}$$

Proof. Based on the definition of $\mathcal{C}^{\mathcal{U},\delta}$ we can re-write $\underset{\mathbf{c}\in\mathcal{C}^{\mathcal{U},\delta}}{\max:}\ \nabla U\left(\overline{\mathbf{R}}(\mathbf{t})\right)^T \mathbf{c}$ as follows:

$$\max_{\phi,\gamma} \quad \sum_{u\in\mathcal{U}} U_u^{'}\left(\overline{R}_u(t)\right)\left(\sum_{s\in\mathcal{S}} p_{\mathcal{S}}(s) g_u^s\left(\phi_u^s, \gamma_u^s\right)\right) \tag{6.24}$$

$$\text{s.t.} \quad \phi \geq (1-\delta)\,\gamma, \tag{6.25}$$

$$\phi,\ \gamma \in \Pi^{U,\delta}. \tag{6.26}$$

Observe that the above optimization problem can be solved separately for each network state $s \in \mathcal{S}$. The de-coupled problem for any state s is same as the optimization problem (6.43) in our online algorithm. With a slight abuse of notation, let $\left(\tilde{\phi}(s), \tilde{\gamma}(s)\right)$ be the optimal solution to the online problem when $S(t) = s$. Conditioned on $S(t) = s$, we have that:

$$\mathbb{E}\left[R_u(t) \mid S(t) = s\right] = \mathbb{E}\left[f_u^s\left(\tilde{\phi}_u^s, \tilde{\gamma}_u^s D\right) \mid S(t) = s\right] = g_u^s\left(\tilde{\phi}_u^s, \tilde{\gamma}_u^s\right) \quad \forall u \in \mathcal{U}. \tag{6.27}$$

Computing $\mathbb{E}\left[\mathbb{E}\left[R_u(t) \mid S(t)\right]\right]$ gives the desired result (6.23). □

The main intuition behind the proof of optimality is that for large t, the trajectories of $\overline{\mathbf{R}}(t)$ can be approximated by the solution to the following differential equation in $\mathbf{x}(t)$ with continuous time t:

$$\frac{d\mathbf{x}(t)}{dt} = \underset{\mathbf{c}\in\mathcal{C}^{\mathcal{U},\delta}}{\text{argmax:}}\ \nabla U\left(\mathbf{x}(t)\right)^T \mathbf{c} - \mathbf{x}(t). \tag{6.28}$$

Let us define $q(\mathbf{x}) := \underset{\mathbf{c}\in\mathcal{C}^{\mathcal{U},\delta}}{\text{argmax:}}\ \nabla U\left(\mathbf{x}\right)^T \mathbf{c}$. To show the optimality of our online algorithm, we shall also require the following result on the above differential equation.

Lemma 6.11.2. *The differential equation* (6.28) *is globally asymptotically stable. Furthermore, for any initial condition* $\mathbf{x}(0) \in C^{\mathcal{U},\delta}$*, we have that* $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{r}^*$.

Proof. To prove this lemma it is enough to show that there exists a Lyapunov function $L(\mathbf{x}(t))$ such that it has a negative drift when $x(t) \neq \mathbf{r}^*$ and has zero drift when $x(t) = \mathbf{r}^*$. Define $L(\mathbf{x}) = U(\mathbf{r}^*) - U(\mathbf{x})$. Observe that under our assumption of strictly concave $U_u(\cdot)$, the offline optimization problem is guaranteed to have an unique optimal solution, which is $\mathbf{r}^*$. Therefore, $\forall \mathbf{x} \in C^{\mathcal{U},\delta}$ and $\mathbf{x} \neq \mathbf{r}^*$ $L(\mathbf{x}) > 0$. Next we will compute the drift of $L(\mathbf{x}(t))$ with respect to time.

$$\frac{dL(\mathbf{x}(t))}{dt} = -\nabla U\left(\mathbf{x}(t)\right)^T \frac{d\mathbf{x}(t)}{dt}, \tag{6.29}$$

$$= -q\left(\mathbf{x}(t)\right) + \nabla U\left(\mathbf{x}(t)\right)^T \mathbf{x}(t), \tag{6.30}$$

$$< 0 \qquad \forall \mathbf{x}(t) \neq \mathbf{r}^*. \tag{6.31}$$

To get inequality (6.31), first observe that from the definition of $q(\mathbf{x}(\mathbf{t}))$ and (6.30), we get that $\frac{dL(\mathbf{x}(t))}{dt} \leq 0$. However, we have to show that this inequality is strict for $\mathbf{x}(t) \neq \mathbf{r}^*$. Observe that $q(\mathbf{x}) = \mathbf{x}$ is a necessary and sufficient condition for optimality of the offline optimization problem, see [83] for more details. From strict concavity of the utility functions, we have an unique optimal point $\mathbf{r}^*$. Therefore, $\frac{dL(\mathbf{x}(t))}{dt} < 0$ for $\mathbf{x}(t) \neq \mathbf{r}^*$ and $\frac{dL(\mathbf{x}(t))}{dt} = 0$ at $\mathbf{x}(t) = \mathbf{r}^*$. □

To conclude the proof, Lemmas 6.11.1 and 6.11.2 along with the condition 2 satisfy all the conditions necessary to apply Theorem 2.1 in Chapter 5, [51] which

states that $\overline{\mathbf{R}}(t)$ converges to $\mathbf{r}^*$ almost surely.

6.11.1 Proofs and Additional Results from Section 6.6

First we state is a corollary to Theorem 6.5.1 for systems having threshold model for superposition/puncturing.

Corollary 6.11.3. *Under a $(1-\delta)$ sharing factor and time-homogeneous scheduler $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{U,\delta}$ the probability of induced eMBB loss for user $u \in \mathcal{U}$ in channel state $s \in \mathcal{S}$ is given by*

$$\epsilon_u^{\boldsymbol{\pi},s} = 1 - F_D(\frac{\phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})}{\gamma_u^{\boldsymbol{\pi},s}}).$$

where F_D denotes the cumulative distribution function of the URLLC demands on a typical eMBB slot. Then the associated user throughput is given by

$$r_u^{\boldsymbol{\pi},s} = \hat{r}_u^s \phi_u^{\boldsymbol{\pi},s} F_D(\frac{\phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})}{\gamma_u^{\boldsymbol{\pi},s}}).$$

and the overall user throughputs are given by $\mathbf{c}^{\boldsymbol{\pi}} = (c_u^{\boldsymbol{\pi}} : u \in \mathcal{U})$ where

$$c_u^{\boldsymbol{\pi}} = \sum_{u \in \mathcal{U}} \hat{r}_u^s \phi_u^s F_D(\frac{\phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})}{\gamma_u^{\boldsymbol{\pi},s}}) p_S(s).$$

The following two corollaries are direct consequences of Corollary 6.11.3 and Theorem 6.5.3 restricted to RP and TP Placement strategies, and characterize the throughput regions under these policies.

Corollary 6.11.4. *Consider a wireless system with full sharing factor and time-homogeneous scheduler based on the RP URLLC Placement policy $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{RP,\delta}$. Then any eMBB resource allocation $\boldsymbol{\phi}$ combined with a RP URLLC demand*

placement policy, $\boldsymbol{\gamma} = \boldsymbol{\phi}$ is feasible. The probability of loss for user $u \in \mathcal{U}$ in channel state $s \in \mathcal{S}$ is given by

$$\epsilon_u^{\boldsymbol{\pi},s} = 1 - F_D(t_u^s(\phi_u^{\boldsymbol{\pi},s})),$$

with associated user throughput

$$r_u^{\boldsymbol{\pi},s} \quad = \quad \hat{r}_u^s \phi_u^s F_D(t_u^s(\phi_u^{\boldsymbol{\pi},s})). \tag{6.32}$$

Further if for all $s \in \mathcal{S}$ and $u \in \mathcal{U}$ the functions $g_u^s(,)$ given by

$$g_u^s(\phi_u^s) \quad = \quad \phi_u^s F_D(t_u^s(\phi_u^{\boldsymbol{\pi},s})), \tag{6.33}$$

are concave then $\mathcal{C}^{RP,\delta} = \hat{\mathcal{C}}^{RP,\delta}$.

Corollary 6.11.5. *Under a $(1-\delta)$ sharing factor and jointly uniform scheduler based on the TP URLLC Placement policy $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{TP,\delta}$, the probability of induced eMBB loss user $u \in \mathcal{U}$ in channel state $s \in \mathcal{S}$ is given by*

$$\epsilon_u^{\boldsymbol{\pi},s} \quad = \quad 1 - F_D(\sum_{u \in \mathcal{U}} \phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})), \tag{6.34}$$

with associated user throughput

$$r_u^{\boldsymbol{\pi},s} = \hat{r}_u^s \phi_u^s F_D(\sum_{u \in \mathcal{U}} \phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})). \tag{6.35}$$

Further if for all $s \in \mathcal{S}$ and $u \in \mathcal{U}$ the functions $g_u^s(,)$ given by

$$g_u^s(\phi_u^s, \gamma_u^s) \quad = \quad \phi_u^s F_D(\sum_{u \in \mathcal{U}} \phi_u^{\boldsymbol{\pi},s} t_u^s(\phi_u^{\boldsymbol{\pi},s})), \tag{6.36}$$

are jointly concave then $\mathcal{C}^{TP,\delta} = \hat{\mathcal{C}}^{TP,\delta}$.

Finally, using the above corollary, we show the optimality of TP Placement with respect to probability of loss on a given eMBB slot.

Theorem 6.6.1. *Consider a system with* $(1-\delta)$ *sharing factor. Consider a joint scheduling policy based on the TP URLLC Placement i.e,* $\boldsymbol{\pi} = (\boldsymbol{\phi}^{\boldsymbol{\pi}}, \boldsymbol{\gamma}^{\boldsymbol{\pi}}) \in \Pi^{TP,\delta}$. *Then* $\boldsymbol{\pi}$ *achieves the minimum probability of eMBB loss amongst all joint scheduling policies using the same eMBB resource allocation* $\boldsymbol{\phi}^{\boldsymbol{\pi}}$.

Proof. Clearly the probability of loss depends on the minislot demands and the users thresholds. If one relaxes the sequential constraint on URLLC allocations, one can consider aggregating the the minislot demands and pooling together the users superposition/puncturing thresholds. The probability of loss for this relaxed system is simply the probability the demand exceeds the size of the superposition/puncturing pool, i.e., The probability of loss under the pooled resources is given by

$$P(D \geq \sum_{u \in \mathcal{U}} \phi_u^s t_u^s(\phi_u^s)).$$

This is clearly a lower bound for any placement policy. Note however that the threshold proportional strategy meets this bound from Corollary 6.11.5 (see Equation 6.34) so it indeed minimizes the probability of loss on a given eMBB slot. □

Theorem 6.11.6. *Under Assumptions 6, there exists an optimal solution* $(\phi^{s,*}, \gamma^{s,*}(\cdot,\cdot))$ *for* $\mathcal{OP}_1$ *with a mini-slot homogeneous URLLC placement policy.*

Proof. The proof has the following three steps.

1. We shall first upper bound the optimal value of $\mathcal{OP}_1$ by the solution to a hypothetical *non-causal* scenario described in the sequel.

2. We show that for the hypothetical non-casual scenario there exists an optimal joint scheduling policy with mini-slot homogeneous URLLC placement policy which in general is a function of the aggregate URLLC load in an eMBB slot.

3. Lastly, under Assumption 6 on the loss functions, we show that there exists an URLLC placement policy policy which is still mini-slot homogeneous but independent of the aggregate URLLC load.

6.11.2 Upper bound on $\mathcal{OP}_1$

At the beginning of each eMBB slot, first the scheduler chooses $\phi^{s,\pi}$. Next the total URLLC demand in each mini-slot is revealed, i.e., the realizations of $D_1, D_2, \ldots, D_{|\mathcal{M}|}$ are revealed. Therefore, this setting is not causal as it assumes exact knowledge about future events. In general the URLLC placement under the non-causal setting is dependent on the mini-slot index m and $\mathbf{D}\left(|\mathcal{M}|\right) := \left[D_1,\, D_2, \ldots, D_{|\mathcal{M}|}\right]$. With slight abuse of notation, we shall denote it by $\gamma_u^s\left(m, \mathbf{D}\left(|\mathcal{M}|\right)\right)$. The joint scheduling policy has to satisfy the constraints (6.17), (6.18), and (6.19). We have the following lemma on the *non-causal* setting.

Lemma 6.11.7. *There exists an optimal mini-slot homogeneous policy for the non-casual setting such that the URLLC placement depends only on the total URLLC demand in an eMBB slot, i.e.,* $\sum_{m=1}^{|\mathcal{M}|} D_m$.

Proof. Let $\left(\tilde{\phi}^{\pi}, \tilde{\gamma}^{s,\pi}\left(\cdot,\cdot\right)\right)$ be the decision variables under an optimal joint scheduling

policy π in the non-causal setting. Let $d_1, d_2, \ldots, d_{|\mathcal{M}|}$ be realizations of $D_1, D_2, \ldots, D_{|\mathcal{M}|}$ such that $\sum_{m=1}^{|\mathcal{M}|} d_m = d$. Define the following:

$$\nu_u^s := \frac{\sum_{m=1}^{|\mathcal{M}|} \tilde{\gamma}_u^{s,\pi}\left(m, \mathbf{d}\left(|\mathcal{M}|\right)\right) d_m}{d}. \tag{6.37}$$

Note that with the definition of ν_u^s, the total puncturing experienced by an eMBB user u in an eMBB slot is $\nu_u^s d$. From this one can construct an equivalent mini-slot homogeneous URLLC placement policy. For all mini-slots, use ν^s as the URLLC placement factor. This satisfies the constraints (6.17), (6.18), and (6.19). In general ν^s could depend on d_1, d_2, $\ldots, d_{|\mathcal{M}|}$. However, we will show that the optimal solution depends only on the sum $\sum_{m=1}^{|\mathcal{M}|} d_m$.

Let d'_1, $d'_2, \ldots,$ $d'_{|\mathcal{M}|}$ be such that $\sum_{m=1}^{|\mathcal{M}|} d'_m = d$ and there exists an m such that $d'_m \neq d_m$. Define the following:

$$\nu_u'^s := \frac{\sum_{m=1}^{|\mathcal{M}|} \tilde{\gamma}_u^{s,\pi}\left(m, \mathbf{d}'\left(|\mathcal{M}|\right)\right) d'_m}{d}. \tag{6.38}$$

Therefore, the total puncturing observed by $\nu_u'^s d$. Observe that ν'^s is also a feasible URLLC policy for the case when the URLLC demand realizations are d_1, d_2, $\ldots$, $d_{|\mathcal{M}|}$. Similarly ν^s is also a feasible URLLC placement policy for the case with d'_1, d'_2, $\ldots$, $d'_{|\mathcal{M}|}$. Therefore, the optimal solution has to be independent of the realizations of D_1, $D_2, \ldots, D_{|\mathcal{M}|}$ and depends only on the sum $\sum_{m=1}^{|\mathcal{M}|} D_m$. □

Therefore, we shall restrict ourselves to mini-slot homogeneous policies in the non-causal setting with the URLLC placement as a function of the total URLLC demand for that eMBB slot. With slight abuse of notation we shall denote a URLLC

placement policy in this setting by $\gamma_u^s(\cdot)$ with the only argument as the total URLLC demand in that eMBB slot. This procedure is formally described next.

1. At the beginning of an eMBB slot, the joint scheduler chooses $\phi_u^{s,\pi}, u \in \mathcal{U}$ such that
$$\sum_{u\in\mathcal{U}} \phi_u^{s,\pi} = 1 \text{ and } \phi_u^{s,\pi} \in [0,1] \quad \forall u. \tag{6.39}$$
2. The total URLLC demand $D = \sum_{m=1}^{|\mathcal{M}|} D_m$ in that eMBB slot is revealed.
3. For an URLLC demand of D, $\gamma_u^{s,\pi}(D)$ is chosen such that
$$\sum_{u\in\mathcal{U}} \gamma_u^{s,\pi}(D) = 1, \quad \text{and} \quad \gamma_u^{s,\pi}(D) \in [0,1]\,. \tag{6.40}$$

Let us denote the feasible policies for this hypothetical non-causal scenario by $\Pi^\dagger$. $(\phi^{s,\pi}, \gamma^{s,\pi}(\cdot))$ is chosen as the solution to the following optimization problem.

$$\mathcal{OP}_2: \quad \max_{\pi\in\Pi^\dagger} : \sum_{u\in\mathcal{U}} w_u g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}(\cdot)\right), \tag{6.41}$$

where $g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}(\cdot)\right) = r_u^s \phi_u^{s,\pi} \mathbb{E}\left[1 - h_u^s\left(\gamma_u^{s,\pi}(D)D, \phi_u^{s,\pi}\right)\right]$. First we have the following important lemma.

Lemma 6.11.8.

$$\max_{\pi\in\Pi^\dagger} : \sum_{u\in\mathcal{U}} w_u g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}(\cdot)\right) \geq \max_{\pi\in\tilde{\Pi}} : \sum_{u\in\mathcal{U}} w_u g_u^{s,\pi}\left(\phi_u^{s,\pi}, \gamma_u^{s,\pi}(\cdot,\cdot)\right). \tag{6.42}$$

Proof. This directly follows from the proof of Lemma 6.11.7 where we have shown that any URLLC placement factor $\gamma_u^{s,\pi}(\cdot,\cdot)$ can be transformed into a mini-slot homogeneous policy which depend only on the total URLLC demand in an eMBB slot, and hence, any feasible solution to $\mathcal{OP}_1$ is a feasible solution for $\mathcal{OP}_2$. □

In general the optimal URLLC placement policy under $\mathcal{OP}_2$ may depend on the total URLLC demand in an eMBB slot. However, under the Assumption 6 it is independent of the total URLLC demand. This is stated formally in the following lemma.

Lemma 6.11.9. *Under Assumption 6, there exists an optimal solution* $(\phi^{s,*}, \gamma^{s,*}(\cdot))$ *for* $\mathcal{OP}_2$ *with URLLC placement policy* $(\gamma^{s,*}(\cdot))$ *independent of* D.

Proof. If $(\phi^{s,*}, \gamma^{s,*}(\cdot))$ is an optimal solution to $\mathcal{OP}_2$, then $\gamma^{s,*}(\cdot)$ must also be an optimal solution to the following optimization problem in $\gamma^s(\cdot)$.

$$\max_{\gamma^s} \quad \sum_{u\in\mathcal{U}} w_u g_u^s(\phi_u^{s,*}, \gamma_u^s(\cdot)), \tag{6.43}$$

$$\text{s.t.} \quad \phi_u^{s,*} \geq (1-\delta)\,\gamma_u^s(d) \quad \forall u,\, d, \tag{6.44}$$

$$\sum_{u\in\mathcal{U}} \gamma_u^s(d) = 1 \text{ and } \gamma_u^s(d) \in [0,1] \quad \forall u,\, d. \tag{6.45}$$

$$\tag{6.46}$$

For any d and u, from the K.K.T. conditions for the above optimization problem, we have that

$$-w_u r_u^s f(d) h_u^{s'}\left(\frac{\gamma_u^{s,*}(d)}{\phi_u^{s,*}}\right) + \beta(d) + \eta_u(d) - \nu_u(d) - \lambda_u(d) = 0. \tag{6.47}$$

where $\beta(d)$ is an arbitrary constant (function of d) and $\eta_u(d)$, $\nu_u(d)$ and $\lambda_u(d)$ are constants such that

$$\lambda_u(d)\left(\phi_u^{s,*} - \gamma_u^{s,*}(1-\delta)\right) = 0 \quad \text{and} \quad \lambda_u(d) \geq 0 \quad \forall u, \tag{6.48}$$

$$\eta_u(d)\gamma_u^{s,*} = 0 \quad \text{and} \quad \eta_u(d) \geq 0 \quad \forall u, \tag{6.49}$$

$$\nu_u(d)\left(1 - \gamma_u^{s,*}\right) = 0 \quad \text{and} \quad \nu_u(d) \geq 0 \quad \forall u. \tag{6.50}$$

For any $d' \neq d$, if we choose $\beta(d') = \beta(d)\frac{f(d')}{f(d)}$, $\eta_u(d') = \eta_u(d)\frac{f(d')}{f(d)}$, $\nu_u(d') = \nu_u(d)\frac{f(d')}{f(d)}$, and $\lambda_u(d') = \lambda_u(d)\frac{f(d')}{f(d)}$, then $\gamma_u^{s,*}(d)$ and $\phi_u^{s,*}$ satisfy the K.K.T. condition

$$-w_u r_u^s f(d') h_u^{s'}\left(\frac{\gamma_u^{s,*}(d)}{\phi_u^{s,*}}\right) + \beta(d') + \eta_u(d') - \nu_u(d') - \lambda_u(d') = 0. \tag{6.51}$$

Note that we have used the non-zero property of $f(\cdot)$ when we multiply with $\frac{f(d')}{f(d)}$. Hence, $\gamma_u^{s,*}(d)$ and ϕ_u^s are optimal for d' too. Hence, we have a constructed an optimal solution with URLLC placement policy independent of D. □

We have shown in Lemma 6.11.9 that there exists an optimal policy $(\phi^{\mathbf{s},*}, \gamma^{\mathbf{s},*}(\cdot))$ which is a mini-slot homogeneous policy and independent of the realization of D. In Lemma 6.11.8, we have also shown that the optimal value of $\mathcal{OP}_2$ is an upper bound for $\mathcal{OP}_1$. Hence, there exists a mini-slot homogeneous policy which achieves an upper bound for $\mathcal{OP}_1$. Therefore, there exists a mini-slot homogeneous policy which is optimal for $\mathcal{OP}_1$.

□

Theorem 6.11.10. *Under the assumptions of exchangeable URLLC arrivals (D_m) in every mini-slot and convex loss functions ($h_u(\cdot)$), if $\mathbb{E}[h_u(D_1)] < \infty$, then we have that*

$$\mathbb{E}\left[h_u\left(\sum_{m=1}^{\phi_u|\mathcal{M}|} D_m\right)\right] \geq \mathbb{E}\left[h_u\left(\sum_{m=1}^{|\mathcal{M}|} \phi_u D_m\right)\right]. \tag{6.52}$$

Proof. We shall assume that $k := \phi_u|\mathcal{M}|$ is an integer. Let $\mathcal{S}_k$ be the set of all subsets with k elements chosen from the set $\{1, 2, \ldots, |\mathcal{M}|\}$. For example, if $|\mathcal{M}| = 3$ and

$k = 2$, then $\mathcal{S}_k = \{\{1,2\},\{2,3\},\{1,3\}\}$. Note that $|\mathcal{S}_k| = \binom{|\mathcal{M}|}{k}$. Using the above definitions, we can re-write the R.H.S. of (6.52) as follows:

$$\mathbb{E}\left[h_u\left(\sum_{m=1}^{|\mathcal{M}|}\phi_u D_m\right)\right] = \mathbb{E}\left[h\left(\frac{1}{\binom{|\mathcal{M}|}{k}}\sum_{q\in\mathcal{S}_k}\left(\sum_{m\in q} D_m\right)\right)\right]. \tag{6.53}$$

Using the above expression one can apply Jensen's inequality on the R.H.S. of (6.52), we have that

$$\mathbb{E}\left[h_u\left(\sum_{m=1}^{|\mathcal{M}|}\phi_u D_m\right)\right] \leq \frac{1}{\binom{|\mathcal{M}|}{k}}\mathbb{E}\left[h\left(\sum_{q\in\mathcal{S}_k}\left(\sum_{m\in q} D_m\right)\right)\right]. \tag{6.54}$$

Since D_m's are exchangeable, the R.H.S. of the above expression is same as the L.H.S. of (6.52). Hence, proved. $\square$

Chapter 7

Conclusions

In this thesis we have focused on the design of schedulers for next generation wireless networks which support heterogeneous application mixes, characterized by different, possibly complex, application/user Quality of Experience (QoE) metrics. The central problem underlying resource allocation for such systems is realizing QoE trade-offs among various applications/users given the dynamic loads and capacity variability they would typically see. We optimized various flow-level delay based metrics based which are directly related to the QoE of users. This approach is different from the traditional approach of using rate and packet based metrics which do not directly relate to user experience.

We have shown that using apriori information on flow sizes/distributions as well as on application QoE requirements from higher OSI layers like the application and transport layers can help us realize complex trade-offs in QoE. In future we envision network protocols which provide more higher layer information to the schedulers and QoE-aware scheduler designs which can exploit such information for better QoE management. We have also developed robust scheduler designs, which can learn and adapt to the changing traffic conditions like system loads, flow size distributions etc. and in principle do not need any intervention from network operators. Application

of such learning techniques and in general state-of-the-art Machine Learning (ML) techniques to design self-adapting wireless systems would be an interesting future research direction.

URLLC traffic with its stringent reliability requirements has its own specific design challenges, for example, 'tall' vs 'wide' transmissions, 'one shot' vs 'multiple transmissions' etc. which have been discussed in Chapter 5. From the point of a view wireless system design, one has to quantify such trade-offs so as to optimize system parameters. We have developed a queuing network based analytical framework to capture such trade-offs as well as to dimension the system appropriately to support URLLC requirements.

Since wireless spectrum is scarce and expensive it is of practical interest to find efficient multiplexing schemes to share radio resources between URLLC and other types of traffic like the eMBB traffic. We have developed a joint scheduling framework for eMBB and URLLC traffic in a downlink setting based on preemptive puncturing/superposition of eMBB transmissions by URLLC traffic. We then identified scenarios where it was necessary to do joint scheduling of URLLC and eMBB with selective puncturing/superposition of eMBB users based on the robustness of their transmissions as well as scenarios where one could completely de-couple URLLC and eMBB scheduling. In this thesis we have not addressed some of the issues related to URLLC traffic, for example, interaction between the HARQ processes of URLLC and eMBB traffic, exploiting periodicity in URLLC arrivals, and provisioning uplink channel for URLLC QoS requirements. We hope to address these issues in our future research.

Bibliography

[1] Y. Polyanskiy, H. V. Poor, and S. Verdu. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, May 2010.

[2] Bilal Sadiq, Ritesh Madan, and Ashwin Sampath. Downlink scheduling for multiclass traffic in lte. *EURASIP J. Wirel. Commun. Netw.*, 2009:14:9–14:9, Mar. 2009.

[3] Magnus Proebster. Improving the quality of experience with size-based and opportunistic scheduling. In *Proc. International Symposium on Wireless Communications Systems (ISWCS)*, pages 443–448, Aug. 2014.

[4] Tobias Hobfeld, Sebastian Biedermann, Raimund Schatz, Alexander Platzer, Sebastian Egger, and Markus Fiedler. The memory effect and its implications on web qoe modeling. In *Proc. Int. Teletraffic Cong.(ITC)*, pages 103–110, Sep. 2011.

[5] Ricky K. P. Mok, Edward W. W. Chan, and Rocky K. C. Chang. Measuring the quality of experience of http video streaming. In *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 485–492, May 2011.

[6] Pablo Ameigeiras, Juan J. Ramos-Munoz, Jorge Navarro-Ortiz, Preben Mogensen, and Juan M. Lopez-Soler. Qoe oriented cross-layer design of a resource allocation algorithm in beyond 3g systems. *Comput. Commun.*, 33(5):571–582, Mar. 2010.

[7] S. Ben Fred, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts. Statistical bandwidth sharing: A study of congestion at flow level. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 111–122, NY, USA, 2001. ACM.

[8] S. Sesia, I. Toufik, and M. Baker. *LTE – The UMTS Long Term Evolution, From Theory to Practice*. John Wiley and Sons, 2009.

[9] Magnus Proebster, Matthias Kaschub, Thomas Werthmann, and Stefan Valentin. Context-aware resource allocation for cellular wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–19, Jul. 2012.

[10] John C. Gittins, Kevin D. Glazebrook, and Richard Weber. *Multi-armed Bandit Allocation Indices*. Qiley, 2nd edition, 2011.

[11] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari. Wireless communication for factory automation: an opportunity for LTE and 5G systems. *IEEE Communications Magazine*, 54(6):36–43, June 2016.

[12] O. N. C. Yilmaz, Y. P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs. Analysis of ultra-reliable and low-latency 5G communication

for a factory automation use case. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1190–1195, June 2015.

[13] M. Gidlund, T. Lennvall, and J. Akerberg. Will 5G become yet another wireless technology for industrial automation? In *IEEE Int. Conf. on Industrial Technology (ICIT)*, pages 1319–1324, March 2017.

[14] K. I. Pedersen, G. Berardinelli, F. Frederiksen, P. Mogensen, and A. Szufarska. A flexible 5G frame structure design for frequency-division duplex cases. *IEEE Communications Magazine*, 54(3):53–59, March 2016.

[15] 3GPP TSG RAN WG1 Meeting 87, November 2016.

[16] Chairman's notes 3GPP: 3GPP TSG RAN WG1 Meeting 88bis, Available at `http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_88b/Report/`, April 2017.

[17] Chih-Ping Li, Jing Jiang, W. Chen, Tingfang Ji, and J. Smee. 5G ultra-reliable and low-latency systems design. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–5, June 2017.

[18] L. You, Q. Liao, N. Pappas, and D. Yuan. Resource Optimization with Flexible Numerology and Frame Structure for Heterogeneous Services. *ArXiv e-prints*, January 2018.

[19] A. Anand and G. de Veciana. Invited paper: Context-aware schedulers: Realizing quality of service/experience trade-offs for heterogeneous traffic mixes. In *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8, May 2016.

[20] Pablo Ameigeiras, Juan J. Ramos-Munoz, Jorge Navarro-Ortiz, Preben Mogensen, and Juan M. Lopez-Soler. Qoe oriented cross-layer design of a resource allocation algorithm in beyond 3g systems. *Computer Communications*, 33(5):571–582, Mar. 2010.

[21] Kalpana Seshadrinathan and AlanConrad Bovik. Automatic prediction of perceptual quality of multimedia signals—a survey. *Multimedia Tools and Applications*, 51(1):163–186, 2011.

[22] A. Asadi and V. Mancuso. A survey on opportunistic scheduling in wireless communications. *IEEE Commn. Surveys Tutorial*, 15(4):1671–1688, Jan. 2013.

[23] Matthew Andrews. A survey of scheduling theory in wireless data networks. In *Wireless Commun.*, volume 143 of *The IMA Volumes in Mathematics and its Applications*, pages 1–17. Springer New York, 2007.

[24] V. Joseph and G. de Veciana. Nova: Qoe-driven optimization of dash-based video delivery in networks. In *Proc. INFOCOM*, pages 82–90, Apr. 2014.

[25] H. Kowshik, P. Dutta, M. Chetlur, and S. Kalyanaraman. A quantitative framework for guaranteeing QoE of video delivery over wireless. In *Proc. INFOCOM*, pages 290–294, Apr. 2013.

[26] D. Bethanabhotla, G. Caire, and M.J. Neely. Utility optimal scheduling and admission control for adaptive video streaming in small cell networks. In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, pages 1944–1948, Jul. 2013.

[27] Samuli Aalto and Pasi Lassila. Impact of size-based scheduling on flow level performance in wireless downlink data channels. In *Managing Traffic Performance in Converged Networks*, volume 4516 of *Lecture Notes in Computer Science*, pages 1096–1107. Springer Berlin Heidelberg, 2007.

[28] Samuli Aalto and Urtzi Ayesta. Optimal scheduling of jobs with a dhr tail in the m/g/1 queue. In *Proc. Int. Conf. on Performance Evaluation Methodologies and Tools*, ValueTools '08, pages 50:1–50:8, 2008.

[29] Samuli Aalto, Urtzi Ayesta, and Rhonda Righter. On the gittins index in the m/g/1 queue. *Queueing Systems*, 63(1–4):437–458, 2009.

[30] Konstantin Avrachenkov, Urtzi Ayesta, Patrick Brown, and Eeva Nyberg. Differentiation between short and long tcp flows: Predictability of the response time. In *Proc. INFOCOM*, volume 2, pages 762–773, Mar. 2004.

[31] K. Avrachenkov, U. Ayesta, and P. Brown. Batch arrival processor-sharing with application to multi-level processor-sharing scheduling. *Queueing Systems*, 50(4):459–480, 2005.

[32] S.F. Yashkov. Mathematical problems in the theory of shared-processor systems. *Journal of Soviet Mathematics*, 58(2):101–147, 1992.

[33] Mor Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[34] Leonard Kleinrock. *Queueing Systems*, volume II: Computer Applications. Wiley Interscience, 1976.

[35] Ianire Taboada, Jose Oscar Fajardo, Fidel Liberal, and Bego Blanco. Size-based and channel-aware scheduling algorithm proposal for mean delay optimization in wireless networks. In *Proc. ICC*, pages 6596–6600, Jun. 2012.

[36] Ying Zhang and Ake AArvidsson. Understanding the characteristics of cellular data traffic. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, CellNet '12, pages 13–18, NY, USA, 2012.

[37] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402, Nov. 2003.

[38] Arjun Anand and Gustavo de Veciana. Measurement-based scheduler for multi-class qoe optimization in wireless networks. In *Proc. INFOCOM*, pages 1–9, May 2017.

[39] David Tse and Pramod Vishwanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[40] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[41] Kibeom Seong, M. Mohseni, and J.M. Cioffi. Optimal resource allocation for ofdma downlink systems. In *Information Theory, 2006 IEEE International*

Symposium on, pages 1394–1398, Jul. 2006.

[42] Sem Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. In *Proc. INFOCOM*, volume 1, pages 321–331, Mar. 2003.

[43] Sem Borst and Matthieu Jonckheere. Flow-level stability of channel-aware scheduling algorithms. In *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, Apr. 2006.

[44] H. Wu, X. Lin, X. Liu, and Y. Zhang. Application-level scheduling with deadline constraints. In *Proc. INFOCOM*, pages 2436–2444, Apr. 2014.

[45] J. Ghaderi, T. Ji, and R. Srikant. Connection-level scheduling in wireless networks using only mac-layer information. In *Proc. INFOCOM*, pages 2696–2700, Mar. 2012.

[46] Bilal Sadiq and Gustavo de Veciana. Balancing srpt prioritization vs opportunistic gain in wireless systems with flow dynamics. In *Proc. Int. Teletraffic Cong. (ITC)*, pages 1–8, Sep. 2010.

[47] Samuli Aalto, Aleksi Penttinen, Pasi Lassila, and Prajwal Osti. Optimal size-based opportunistic scheduler for wireless systems. *Queueing Systems*, 72(1):5–30, Oct. 2012.

[48] Chi-ping Li and Michael J. Neely. Delay and Power-Optimal Control in Multi-Class Queueing Systems. *ArXiv e-prints*, January 2011.

[49] B. M. Hochwald, T. L. Marzetta, and V. Tarokh. Multiple-antenna channel hardening and its implications for rate feedback and scheduling. *IEEE Trans. Inf. Theory*, 50(9):1893–1909, Sept. 2004.

[50] Nan E, Xiaoli Chu, Weisi Guo, and Jie Zhang. User data traffic analysis for 3g cellular networks. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 468–472, Aug. 2013.

[51] Harold J. Kushner and G. George Yin. *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[52] Arjun Anand and Gustavo de Veciana. A whittle's index based approach for qoe optimization in wireless networks. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(1):15:1–15:39, 2018.

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

[54] Jan A. van Mieghem. Dynamic scheduling with convex delay costs: The generalized c—mu rule. *The Annals of Applied Probability*, 5(3):809–833, 1995.

[55] Avishai Mandelbaum and Alexander L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized c—mu rule. *Operations Research*, 52(6), Dec. 2004.

[56] Itay Gurvich and Ward Whitt. Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing & Service Operations Management*, 11(2):237–253, 2009.

[57] Carlos F. Bispo. Single server scheduling problem: Optimal policy for convex costs depends on arrival rates. In *Proc. Multidisciplinary Int. Conf. on Scheduling : Theory and Applications (MISTA 2011)*, pages 275–296, Aug. 2011.

[58] Rhonda Righter and Susan H. Xu. Scheduling jobs on non-identical ifr processors to minimize general cost functions. *Advances in Applied Probability*, 23(4):909–924, Dec. 1991.

[59] Alexander L. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.*, 14(1):1–53, Feb. 2004.

[60] P. S. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O'Keeffe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57(1):21–39, Apr. 2003.

[61] Samuli Aalto, Pasi Lassila, and Prajwal Osti. Whittle index approach to size-aware scheduling with time-varying channels. In *Proc. ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, pages 57–69. ACM, 2015.

[62] Samuli Aalto, Aleksi Penttinen, Pasi Lassila, and Prajwal Osti. On the optimal trade-off between srpt and opportunistic scheduling. In *Proc. ACM SIGMETRICS Joint Int. Conf. on Measurement and Modeling of Computer Systems*, SIGMETRICS '11, pages 185–196, 2011.

[63] Urtzi Ayesta, Martin Erausquin, and Peter Jacko. A modeling framework for optimizing the flow-level scheduling with time-varying channels. *Perform. Eval.*, 67(11):1014–1029, Nov. 2010.

[64] Peter Jacko. Value of information in optimal flow-level scheduling of users with markovian time-varying channels. *Perform. Eval.*, 68(11):1022–1036, Nov. 2011.

[65] Ianire Taboada, Peter Jacko, Urtzi Ayesta, and Fidel Liberal. Opportunistic scheduling of flows with general size distribution in wireless time-varying channels. In *Proc. Teletraffic Cong. (ITC)*, pages 1–9, Sep. 2014.

[66] Urtzi Ayesta, Martin Erausquin, Matthieu Jonckheere, and Maaike Verloop. Scheduling in a random environment: Stability and asymptotic optimality. *IEEE/ACM Transactions on Networking*, 21(1):258–271, Feb. 2013.

[67] Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.

[68] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 4 edition, 2012.

[69] Shailesh Patil and Gustavo de Veciana. Measurement-based opportunistic scheduling for heterogenous wireless systems. *IEEE Trans. Commun.*, 57(9):2745–2753, Sep. 2009.

[70] A. Anand and G. de Veciana. Resource Allocation and HARQ Optimization for URLLC Traffic in 5G Wireless Networks. *ArXiv e-prints*, 2018.

[71] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. G. Ström, K. F. Trillingsgaard, A. Bana, D. Kim, R. Kotaba, J. Park, and R. B. Sørensen. Ultra-reliable low-latency communication (URLLC): principles and building blocks. *CoRR*, abs/1708.07862, 2017.

[72] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim. Introduction to ultra reliable and low latency communications in 5g. *CoRR*, abs/1704.05565, 2017.

[73] S. Ashraf, Y. P. E. Wang, S. Eldessoki, B. Holfeld, D. Parruca, M. Serror, and J. Gross. From radio design to system evaluations for ultra-reliable and low-latency communication. In *Proc. European Wireless Conference*, pages 1–8, May 2017.

[74] G. Durisi, T. Koch, and P. Popovski. Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proceedings of the IEEE*, 104(9):1711–1726, Sept 2016.

[75] G. Durisi, T. Koch, J. Ostman, Y. Polyanskiy, and W. Yang. Short-packet communications over multiple-antenna rayleigh-fading channels. *IEEE Trans. on Comm.*, 64(2):618–629, Feb 2016.

[76] B. Singh, Z. Li, O. Tirkkonen, M. A. Uusitalo, and P. Mogensen. Ultra-reliable communication in a factory environment for 5G wireless networks: Link level and deployment study. In *IEEE Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5, Sept 2016.

[77] Leonard Kleinrock. *Queueing Systems*, volume I. Wiley, 1975.

[78] A. Anand, G. de Veciana, and S. Shakkottai. Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks. *ArXiv e-prints*, 2017.

[79] D. Julian. Erasure networks. In *Proc. IEEE International Symposium on Information Theory,*, Jul. 2002.

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

[81] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant. Low-complexity scheduling algorithms for multi-channel downlink wireless networks. In *Proceedings of IEEE Infocom*, 2010.

[82] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant. Scheduling for small delay in multi-rate multi-channel wireless networks. In *Proceedings of IEEE Infocom*, 2011.

[83] S. Boyd and L. Vandenberge. *Convex Optimization*. Cambridge University Press, 2003.

Vita

Arjun Anand was born in Kerala, India. He received the Bachelor of Technology degree in Electronics and Communication Engineering from National Institute of Technology, Calicut, India in May 2011. Subsequently, he received his Masters' of Engineering in Telecommunication Engineering degree from Indian Institute of Science, Bangalore, India in June 2013. He joined The University of Texas at Austin for his PhD. under the guidance of Prof. Gustavo de Veciana in August 2014.

Permanent address: arjun_anand@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.