On the Power of (even a little) Flexibility in Dynamic Resource Allocation

by

Kuang Xu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature redacted

Author ....
Department of Electrical Engineering and Computer Science
May 19, 2014

Signature redacted

Certified by ....
John N. Tsitsiklis
Clarence J. Lebel Professor of Electrical Engineering
Thesis Supervisor

Signature redacted

Accepted by ....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

On the Power of (even a little) Flexibility in Dynamic Resource Allocation

by

Kuang Xu

B.S., University of Illinois at Urbana-Champaign (2009)

S.M., Massachusetts Institute of Technology (2011)

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

We study the role of *partial flexibility* in large-scale dynamic resource allocation problems, in which multiple types of processing resources are used to serve multiple types of incoming demands that arrive stochastically over time. Partial flexibility refers to scenarios where (a) only a small fraction of the total processing resources is flexible, or (b) each resource is capable of serving only a small number of demand types. Two main running themes are *architecture* and *information*: the former asks how a flexible system should be structured to fully harness the benefits of flexibility, and the latter looks into how information, across the system or from the future, may critically influence performance.

Overall, our results suggest that, with the right architecture, information, and decision policies, large-scale systems with partial flexibility can often *vastly outperform* their inflexible counterparts in terms of delay and capacity, and sometimes be *almost as good as* fully flexible systems. Our main findings are:

1. *Flexible architectures.* We show that, just like in fully flexible systems, a large capacity region and a small delay can be achieved even with very limited flexibility, where each resource is capable of serving only a *vanishingly* small fraction of all demand types. However, the system architecture and scheduling policy

need to be chosen more carefully compared to the case of a fully flexible system. (Chapters 3 and 4.)

2. *Future information in flexible systems.* We show that delay performance in a partially flexible system can be significantly improved by having access to predictive information about future inputs. When future information is sufficient, we provide an optimal scheduling policy under which delay stays *bounded* in heavy-traffic. Conversely, we show that as soon as future information becomes insufficient, delay diverges to *infinity* under any policy. (Chapters 5 and 6.)

3. *Decentralized partial pooling.* For the family of Partial Pooling flexible architectures, first proposed and analyzed by [84], we demonstrate that a decentralized scheduling policy can achieve the same heavy-traffic delay scaling as an optimal centralized longest-queue-first policy used in prior work. This demonstrates that *asymptotically optimal* performance can be achieved in a partially flexible system with *little information sharing.* Our finding, which makes use of a new technical result concerning the limiting distribution of an $M/M/1$ queue fed by a superposition of input processes, strengthens the result of [84], and provides a simpler line of analysis. (Chapter 7.)

Thesis Supervisor: John N. Tsitsiklis

Title: Clarence J. Lebel Professor of Electrical Engineering

Acknowledgments

First and foremost, I would like to thank my doctoral advisor, Professor John N. Tsitsiklis. There are no words adequate in expressing how instrumental Professor Tsitsiklis has been in my development as a researcher as well as a person, nor how grateful I am to having him as my advisor. It is only my hope that this dissertation marks the beginning of a new journey of interaction, collaboration, and learning.

I would like to thank Professors Bruce Hajek and Devavrat Shah for serving on my thesis committee, and for their feedback at various stages of my dissertation research. Special thanks go to Professors David Gamarnik and Devavrat Shah: even though I have not had the opportunity to work with them, we have had many interactions and discussions throughout the years, and they have been incredibly supportive of me in many way.

I have had the fortune to collaborate with a number of researchers during my doctoral study, and I would like to thank them all: Professors Carri W. Chan, Mor Harchol-Balter, Shie Mannor, and Joel Spencer, Doctors Christian Borgs, Jennifer T. Chayes, and Madhu Sudan, and Sherwin Doroudi.

I would like to thank the administrative staff at the Laboratory for Information and Decision Systems (LIDS) for their help, especially Jennifer Donovan, Brian E. Jones, and Debbie Wright. I want to say a special thank-you to Lynne R. Dell, who has been so patient, helpful, and supportive over the years.

Life at LIDS and MIT would not be the same without the student community. I would like to thank the members of SyNDeG and SPPIN, as well as other students in LIDS and ORC, for their friendship and enthusiasm. Special thanks go to Professors Mihalis Markakis and Yuan Zhong, who have survived countless research discussions and counseling sessions. I would also like to thank my friends at the MIT Graduate Student Council (GSC), as well as the MIT Ballroom Dance Team.

This dissertation has been supported by a number of research funds and fellowships, which I would like to acknowledge: the MIT Jacobs Presidential Fellowship, MIT-Xerox Fellowship, Siebel Foundation, MIT-RLE Claude E. Shannon Fellowship, and National Science Foundation grants CCF-0728554, CMMI-0856063, and CMMI-1234062.

Last, but not least, I would like to thank my parents, Xu Kangjie and Wu Yufang, for their love and support. I owe my deepest gratitude to them.

Dedicated to my parents, Xu Kangjie and Wu Yufang.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Imagine being the manager of a call center that supports a large number of product categories. With the goal of minimizing customers' waiting time in mind, which choice of staffing would you rather have?

1. *Flexible* agents, who have expertise in every product category.

2. *Inflexible* agents, who know about one (or a few) category only.

Intuitively, the flexible agents are much more desirable: if one category suddenly receives a burst of inquiries, there will be many other available agents that can come to help. However, flexibility also comes at a cost: having every agent well versed in all categories can be prohibitively expensive in terms of training cost, or simply infeasible. Is there a desirable scheme somewhere in between, involving a multitude of agents with different types of versatility? If so, what are the guiding principles for the system to be *structured*, and efficiently *operated* in real time?

The above example illustrates some aspects of the main focus of this report, that is, to understand the role of *partial flexibility* in large-scale dynamic resource allocation problems, where multiple processing resources (e.g., computer servers, human agents, medical resources, etc.) aim to serve multiple types of demands (e.g., computational tasks, customer inquiries, patients arriving to an emergency room, etc.) that arrive stochastically over time. In this context, by *flexibility* we mean a processing resource's capability to serve *multiple types of demands*, and *partial flexibility* refers to scenarios where:

1. only a small fraction of the total processing resources is flexible, or
2. each processing resource can only serve a small number of demand types.

Why are we interested in studying partially flexible systems? Here are some of the main reasons:

1. *Full flexibility can be expensive or infeasible.* While it is often desirable to have a *fully flexible* system, where each unit of processing resource is capable of serving all types of demand, it can often be too expensive, or simply impossible, to build or operate in practice. Such challenges are only exacerbated in large-scale systems. For instance, in content distribution systems, processing resources correspond to physical servers in server farms, and demand types correspond to different pieces of content (e.g., video clips). Having full flexibility means storing *all* pieces of content on *each* server, which can lead to high infrastructure costs. In the example we saw earlier, having a fully flexible staff at a large call center with hundreds of product categories can simply be impossible due to human skill limitations.

   In these settings, partial flexibility provides a viable, if not the only, alternative, where a small amount of flexibility can be built into the system *inexpensively*, and in a way that remains scalable as the system size grows.

2. *Partial flexibility is effective.* As we will see in subsequent chapters, even a small amount of flexibility can be surprisingly effective. In many scenarios, a partially flexible system can significantly outperform an *inflexible* counterpart, where processing resources are dedicated to serving only one specific demand type, and are sometimes almost as good as a fully flexible system. Viewed from this angle, partial flexibility is important not only as a "last resort" when full flexibility is unavailable. Rather, it can be an appealing design approach that delivers the most desirable cost-performance trade-off.

3. *Partial flexibility is challenging to understand and design.* Unfortunately, despite their practical importance, dynamical systems with partial flexibility appear to be much less understood than either fully flexible or inflexible systems. To a large extend, this is because by allowing for partial flexibility, we have substantially enlarged the space of possible design choices, and hence exposing a multitude of complexities that are unseen in either the fully flexible or inflexible settings, which are often significantly simpler.

The importance of partially flexible systems, compounded with a relatively poor understanding of their nature, has been the main motivation of our inquiries. That being said, the notion of partial flexibility in resource allocation remains rather general, whose manifestations can take on vastly differently forms depending on the angle through which we approach it. Therefore, the scope of our investigation is necessarily contained. In this report, we shall be primarily focusing on resource allocation systems that possess the following features:

1. Non-trivial dynamics. We will be studying queueing systems that involve non-trivial dynamics (as opposed to *static* models), where resource allocation decisions have to be made repeatedly over time. As a result, our metrics will also involve quantities measured in *time*, such as queueing delays and lengths of predictive windows.

2. Large-scale systems. We will be focusing on the regime where the size of the system (e.g., number of processing resources and demand types) grows to infinity. Indeed, most dynamical service systems are fairly intractable, not amenable to exact analytical solutions. On the other hand, *asymptotic analyses* (in the limit of a large number of resources) are often possible and can provide significant architectural insights. Furthermore, they often turn out to be quite accurate even for moderately sized systems. Finally, our interest in large systems is well motivated from many and diverse contexts, such as large data centers, server farms, call centers, etc.

3. Engineered systems. We study systems that are *engineered* or *designed*, and their operations are moderated by decision makers or managers with *collective* objectives, who follow decision *policies* that are prescribed beforehand. This is in contrast to game-theoretic models, where the structure and dynamics of the system can be the result of strategic interactions among different parties. Our modeling focus does not imply that the issue of *human incentives* or strategic interactions are unimportant in designing flexible systems; quite to the contrary, they are often essential ingredients in some of the applications we consider, such as a call center. We chose to focus on engineered systems because they are often easier to analyze, and can help us obtain fundamental understanding of the power and limitations of partially flexible systems. We believe that these insights can, in the future, be used to guide the design of flexible systems that do involve human incentives and interactions.

1.1 Main Themes: Architectures and Information

We outline in this section the main themes of the research presented in this report, as well as preview some of the main contributions. Before we do so, it will be useful to examine a simple example, which will help us gain some intuition about why flexibility should be *beneficial* in dynamic resource allocation problems, and in *what way* such benefits are measured. For the purpose of illustration, we will describe the model informally, and postpone the statements of exact mathematical assumptions till subsequent chapters.

Consider the queueing systems illustrated in Figure 1-1. Each system contains two servers that are capable of processing one job per unit time, and has two types of jobs that arrive at the average rates of λ_1 and λ_2 per unit time, respectively. Two queues, one for each job type, are used to store currently unprocessed jobs. We shall assume that the jobs arrive to each queue according to some *stochastic process* (e.g., a Poisson process), so that the exact times of arrivals of jobs are randomly disbursed across the time horizon, as opposed to arriving exactly at evenly spaced intervals. We say that system (a) is *flexible*, because each server is capable of processing jobs of both types, and system (b) is inflexible, or dedicated, because the servers can process jobs from one queue only.

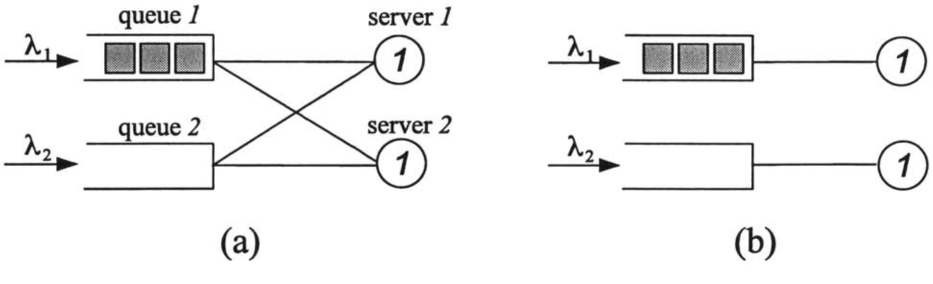

Figure 1-1: Flexible versus inflexible (dedicated) systems.

In what sense is the flexible system *better* than the inflexible one? The first benefit, that of **capacity**, is easy to see. In the flexible system, suppose that the arrival rate to the first queue, λ_1 is greater than 1 and hence exceeds the processing rate of the first server. The system will still be stable as long as the rate λ_2 is sufficiently small. This is because, thanks to being flexible, the second server can devote part of its processing power and *help* process jobs from the first queue, hence stabilizing the system. The inflexible system, however, does not enjoy this benefit: if λ_1 exceeds 1, the first queue in system (b) will inevitably become unstable, and the second server can do nothing to help.

To summarize, compared to an inflexible system, the presence of flexibility allows the system to admit a larger *capacity region*, captured, in our case, by the set of (λ_1, λ_2) pairs that the system is able to stabilize. This is achieved with the *same* amount of processing resources as in the inflexible system; the credit goes solely to flexibility.

While this notion of "flexible resources helping each other" may appear obvious at first, it turns out to have profound, and more subtle, implications in the dynamics of the system. Consider the situation of *uniform* arrival rates, with $\lambda_1 = \lambda_2 = \lambda$. Capacity is no longer a distinguishing factor between the flexible and inflexible systems, because both are stable if and only if $\lambda < 1$. Does this mean that we should expect identical performance from both systems?

Not quite. We shall argue that the flexible system still does better, but this time, in its **delay performance**. Because the arrival process is *stochastic*, it is inevitable that some periods of time will have more arrivals than others. Imagine the occurrence of such a "busy period," where a relatively large number of jobs have just arrived to queue 1, while queue 2 happens to be empty and will remain so for some time (see Figure 1-1). In a flexible system, the sensible thing to do is for both servers to focus on processing jobs from queue 1 and quickly reduce its backlog. In contrast, in an inflexible system, the long backlog in queue 1 will have to be cleared by server 1 alone (and hence more slowly), while server 2 remains idle and incapable of providing any assistance.

The more general phenomenon here is that, in a *dynamic* setting, flexibility also manifests itself in a form of *agility*, which allows for processing resources to be quickly dispatched to serving the *most congested* demand types. In the long run, the net effect of such actions helps avoid the frequent buildup of *large backlogs*, and will ultimately result in a smaller queueing delay. Fundamentally, this notion of agility is no different from the benefits of large capacity region in a flexible system that we saw earlier, because it is simply the result of the flexible processing resources being repeatedly re-purposed throughout the time horizon. However, unlike the more static property of capacity region, the resulting dynamics in a flexible system can often be considerably more difficult to characterize.

Flexible Architectures (Chapters 3 and 4). With the two-queue example of Figure 1-1 in mind, the first line of our research concerns the *structural* aspect of how *partially flexible architectures* should be designed, given only partially flexible processing resources. In particular, we would like to know:

1. With partially flexible resources, can we still harness the benefits of a *large capacity region* and *small delay*, similar to the case of a fully flexible system?

2. If so, how should the flexible architecture be designed, along with the appropriate scheduling polices?

We will study a class of multi-server multi-class queueing systems, with multiple queues (job types) and servers connected through a bipartite graph, where the level of flexibility is captured by the average number of job types a server is capable of processing, d (Figure 1-2). In this framework, the fully flexible system in Figure 1-1 corresponds to the case where the connectivity graph is a complete bipartite graph.

We focus on the scaling regime where the system size n tends to infinity, while the overall traffic intensity stays fixed. Our main finding is that, just like in the case of full flexibility, large capacity region and diminishing queueing delay can be simultaneously achieved even under very limited flexibility ($d \ll n$). However, the flexible architecture, as well as the associated scheduling policy, need be chosen carefully in order to harness the benefits of flexibility. These findings are conveyed through a collection of results, stated in Section 3.3 and summarized in Table 3.1, which characterize the delay and capacity performance for three families of partially flexible architectures: the Modular, Random Graph, and Expanded Modular architectures.

Because each server can be connected to a different set of queues, the family of partially flexible systems we study encompasses a rich set of architectures. Therefore, we will also examine and compare different flexibility architectures and scheduling policies, and examine the extent to which the objectives of a favorable capacity region and delay are possible for each architecture — its strengths, as well as its limitations.

While the analysis of capacity and delay is relatively straightforward for both fully flexible and inflexible systems, conventional techniques often fall short when applied to partially flexible systems with a more complex interconnection topology. As a result, some of our efforts will also go into developing novel problem formulations and analytical methodologies, mostly focusing on asymptotic scaling laws for large systems, which will help us rigorously study the delay and capacity performance of partially flexible systems with non-trivial structures.

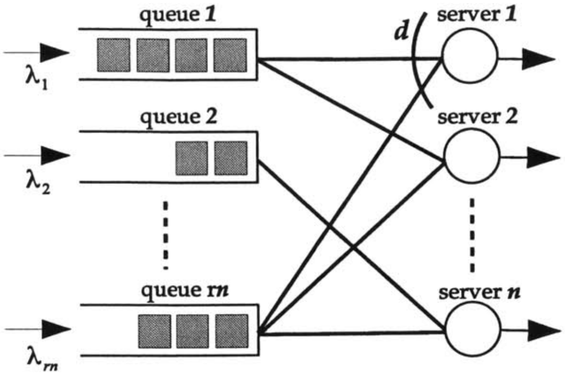

Figure 1-2: A parallel queueing system with multiple job types and flexible servers.

Importance of Information (Chapters 5 to 7). We next shift our focus to studying a less obvious, but equally important, aspect of flexible systems: the role of information. In the two-queue systems of Figure 1-1 that we saw earlier, this is reflected by the fact that the flexible servers must have up-to-date information of the lengths of the two queue at all times, in order to constantly focus the collective processing resource on serving the most needed demand type. Conversely, suppose that the queue length information is unknown, and that the server incurs a non-negligible delay when trying to serve an empty queue. The servers in the flexible system will then suffer from unnecessary idling because of not knowing which queue to serve, and it can be shown that the resultant delay performance can be comparable to that of an inflexible system, despite the presence of flexibility.

In some sense, having a well-designed flexible architecture is only half of the picture. The decision maker must also be equipped with *adequate information*, in the right place at the right time, in order to make optimal resource allocation decisions. Moreover, studying the information requirements of flexible systems also helps us understand *how much* information is necessary to achieve a target level of performance. This is of practical importance especially in large-scale systems, because a high level of information sharing can often lead to significant infrastructural or communications overhead.

For the second part of the report, we focus on a class of flexible queueing systems first proposed by [84], henceforth referred to as the Partial Pooling family, where a small fraction of the processing resources are *fully flexible* while the remaining resources are *dedicated*. We first show that having access to information *ahead of time* can provide substantial improvement in performance. We show that the decision maker can leverage a *finite* lookahead window, within which the times of future arrivals and services are revealed, and drastically decrease the resulting delay in heavy-traffic (Theorem 5.13), compared to that of an optimal *online* policy, which does not make use of future information (Theorem 5.8). Conversely, we quantify *how much* future information is *necessary* in order to improve performance, by proving a tight, *information lower bound*, which shows that with insufficient future information, delay performance cannot be improved by more than a *constant* factor over that of an online policy (Theorem 5.14).

We further demonstrate that in these Partial Pooling systems, a *decentralized* scheduling policy that uses only local queue length information achieves *optimal* delay, which is significantly smaller than that of an inflexible system, where all processing resources are dedicated (Theorem 7.1). In contrast, the scheduling scheme given in [84] requires real-time information for *all* queues in the systems.

1.2 Related Research

We review in this section some of the existing literature and prior research that is related to our work. We shall stay at a relatively general level, by highlighting the connections and differences in terms of philosophies and approaches. More extended discussions of related work concerning specific topics and techniques will be presented within subsequent chapters.

Earlier studies of flexible systems can be traced back to the 1980s, where the focus was largely on the performance and cost tradeoffs between fully flexible versus fully dedicated designs (see [72] for a survey). The seminal paper of Jordan and Graves [47] was the first to consider the design and performance of manufacturing systems with limited flexibility. It was shown, both empirically and through simulations, that when each plant is only capable of producing a small number of products (partial flexibility) a specific type of assignment architecture, called the "Long Chain," can offer performance comparable to that of a fully flexible system (where all products can be produced by all plants). A large body of literature has since followed, extending the idea of the Long Chain into other application domains [9, 13, 37, 38, 45, 46, 58, 88], and providing theoretical justifications for the effectiveness of the Long Chain and its variations [22, 23, 73]. With a few exceptions, the Long Chain model and its variants have been traditionally applied to *static* allocation problems (with a single or a small number of stages), and the results are often justified either empirically or via simulations. In contrast, we will be focusing on problems that involve non-trivial *dynamics*, where resource allocation decisions have to be made repeatedly over time, as well as on developing precise analytical results and *scaling laws.* We view our work as highly complementary to the above mentioned literature.

Another line of work concerns the design of load balancing systems and bears close intellectual ties to ours. Here, the general problem is to direct a stream of incoming tasks to a set of queues for processing. In the line of work initiated in [87] and [63] (popularly known as the "supermarket model"), it is shown that by routing tasks to the shorter queue among a small number ($d \geq 2$) of randomly chosen queues, the probability that a typical queue has at least i tasks decays as $\lambda^{\frac{d^i-1}{d-1}}$ (super-geometrically), as $i \to \infty$. The general idea has since been extended to various other settings [3, 18, 36, 56, 57, 61]; see also the survey paper [64] and references therein. Another line of work is concerned with the impact of service flexibility in routing problems, motivated by applications such as multilingual call centers, dating back to the seminal work in [32] which shows that the ability to route a portion of customers to a least-loaded station can lead to a constant-factor improvement in average delay under diffusion scaling. Similar models have been subsequently studied in [42, 68, 79] and more recently in [1, 2]. A major difference between our approach and the load balancing literature is our focus on resource flexibility (e.g., in scheduling and resource allocation), as opposed to to the demand flexibility in load-balancing and routing problems; in fact, our earlier work suggests that the system dynamics can be fundamentally different under these two flexibility types (see the discussion in Section 1.3 of [91]). Despite the differences, we expect that many of the concepts and techniques developed in the load-balancing literature, and in particular those for analyzing large-scale systems, will spark fruitful synergies with our theory and methodologies.

There are several other strands of the literature that touch upon some of the themes in this report, although the details of the models therein are quite different. Flexibility in the form of resource pooling is known to improve performance [11, 41, 59, 60], but much less is known on the impact of various degrees of pooling, or about scaling behaviors in large-system limits. Some recent work in this area [10] that studies limited pooling in a large-system limit is closer to our work in spirit, but still differs significantly in terms of critical modeling assumptions and dynamics. Regarding the role of information sharing on performance, the information required to control an unstable plant has been studied in the control theory community [15, 26, 65, 70, 82], but in a completely different context, and with a greater emphasis on stability rather than performance (e.g., delay or queue length). Finally, there have been studies of advanced reservations (a form of future information) in lossy networks [24, 55] and, more recently, in revenue management [54], although the motivation of and dynamics in these models are very different from ours.

Parts of the material presented in this report have also appeared in a number of earlier papers. Preliminary results from Chapters 3 and 4 appeared in SIGMETRICS 2013 [85]. Chapters 5 and 6 are based on [77] and [92], respectively. The material of Chapter 7 is new and has not appeared in any publication.

1.3 Organization of the Report

The remainder of report largely follows the logical development of the two themes described in Section 1.1. We begin by describing some of our main modeling assumptions and notation in Chapter 2. Chapters 3 to 4 examine the design and analysis of partially flexible architectures, and Chapters 5 to 7 are devoted to the role of information in partially flexible systems. We conclude the report in Chapter 8, where we also highlight several potential avenues for future research.

Chapter 2

Models and Notation

This chapter presents the queueing model that will form the basis of our analysis. We shall refrain from delving into great details of the mathematical formalism, which will be presented in subsequent chapters, and instead focus on highlighting the main modeling features, as well as connections between the more specific models adopted in different chapters.

2.1 Multi-Server Multi-Type Queueing Model with Flexible Servers

The general problem is that of allocating n units of processing resources to serve demands of m types, as depicted in Figure 2-1. We shall focus on the regime where the total number of demand types, m, is proportional to the total amount of processing resources, n, so that $m = rn$, where $r \in \mathbb{R}_+$ is a constant.[1] For simplicity of notation,

[1] Throughout, we shall avoid the excessive use of floors and ceilings, and assume that relevant quantities are appropriated rounded to an integer.

we will further assume that the number of demand types is equal to n ($r = 1$). As it will become clear in subsequent chapters, most results generalize to the case of other values of r as well.

Demands arrive to the system in the form of discrete *jobs*.[2] For the ith demand type, we assume that jobs arrive according to an independent Poisson process of rate $\lambda_i \in \mathbb{R}_+$, that is, the inter-arrival times between two adjacent jobs are independent and identically distributed (i.i.d.), according to an exponential distribution with mean $1/\lambda_i$. An infinite *buffer*, or *queue*, is associated with each job type, to store the jobs that are currently unprocessed.

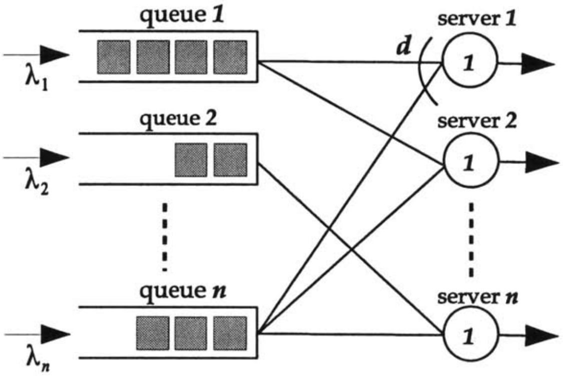

Figure 2-1: The multi-server multi-type queueing model.

We now turn to the modeling of processing resources. For most parts of this report, it suffices to think of the n units of total processing resources as a collection of n *servers*, each being capable of processing jobs at the average rate of 1 job per unit time. The **flexibility** of the processing resources is captured by the types of jobs each server is capable of processing, illustrated in Figure 2-1 by the bipartite graph that connects the servers to their corresponding compatible job types. We will

[2] We will use the terminology "demand" and "job" interchangeably from this point onward.

assume that such service flexibility is *fixed* over time, once the system is built.

Similar to the arrival process, the way jobs are being served is assumed to be *stochastic* to reflect the inherent variability of service times or server speeds in practical applications. Barring minor differences, the models for service stochasticities in this report fall under one of the following two types:

1. *Exponential service times (Service Time model).* In this setup, each job is associated with a random *job size* (a.k.a. workload), which is independently distributed as an exponential random variable with mean 1, regardless of its type. To initiate service, a job is transferred from the queue into a compatible server, and the corresponding server cannot accept a new job until the processing of the current job has been completed. We assume that each server works at a *constant* speed of 1, and therefore the *service time* to complete the processing of one job is equal to the job's size. A job departs the system as soon as it has received an amount of work that is equal to its size. The Service Time model will be used in Chapters 3 and 4.

2. *Poisson service token generation (Service Token model).* In this setup, each server constantly generates *service tokens* according to a Poisson process, whose rate is equal to that of the server (in this case, one). The generations of service tokens are independent across different servers. When a service token is generated, it is either "consumed" to instantly serve a job currently waiting in queue, in which case the job departs from the queues, or "wasted" (e.g., when all queues are empty) and causes no further change to the system. Note that because service speed variations are solely associated with the service token process rather than the job sizes, all jobs of the same type are essentially indistinguishable in the Service Token model.

As a rough analogy, the generation of a service token can be thought of as being equivalent to completing the service of a job in the Service Time model — both events lead to a new job being taken away from a queue. This analogy will be discussed in more detail in a subsequent paragraph that compares the two models. The Service Token model will be used in Chapters 5 to 7.

Note that in either model, we have not specified *which* jobs are to be served when processing resources become available. Indeed, these decisions, to be made dynamically by a *scheduling policy*, play a central role in the design of flexible systems, and shall be treated in detail in subsequent sections.

Service Time versus Token. By definition, the Service Time model attributes the source of service variability to the variation in job sizes, while the Service Token model postulates that the server's processing "speed," captured by the generation of service tokens, is stochastic.

However, there is, in fact, very little difference in the queueing dynamics and resulting performance from the two models, largely as a result of the properties of Poisson processes and exponential service times. First, it can be shown that the Service Time model is capable of *simulating* the *queue length dynamics* produced under a Service Token model, by allowing a server to stay idle, or process "dummy" jobs. To see why this is possible, consider a Service Time model with just one queue and one server. Upon the completion of a previous job, assume that the server uses the following rule:

1. If the queue is not empty, the server fetches a job from the queue and initiates its service.

2. If the queue is empty, the server initiates the service of a fictitious "dummy job",

whose size is an exponential random variable with mean 1, drawn independently from the rest of the system dynamics.

If we refer to the completion of both real and dummy jobs as a "service completion," then it is not difficult to verify that the *times of service completions* under the above-mentioned rule form a Poisson process of rate 1. Because one job *leaves* the queue at each of these service completions if the queue is non-empty, the above rule produces a queue length process that has the same distribution as one under a Service Token model, where the generation of a service token corresponds to that of a service completion. An analogous argument can be used to show such "simulation" for the general case with multiple queues and servers.

The above simulation argument implies that the Service Time model is strictly more powerful in terms of the set of queueing dynamics it is capable of producing. As a result, all queueing performance guarantees derived under the Service Token model can be achieved with Service Time as well.

On the other hand, the Service Token model is not much weaker than the Service Time model. Note that by viewing the generation of a service token as being equivalent to a service completion in the Service Time model, the evolution of the queues is essentially identical under both models when all queues are non-empty. Indeed, it is possible to show that the behavior of the two models are very similar when the system is *heavily loaded.* The reader is referred to [84, 91] for additional discussions on the relationship between the two server models.

Both approaches to server modeling have appeared in the literature, and the Service Time model is more common (cf., the model of $M/M/1$ queues). We chose to use the Service Token model in some of the chapters, because it often allows for more concise descriptions of the model, as well as simpler calculations. Nevertheless, all results derived in this report for the Service Token model can also be extended to the Service Time model, and the two models can be thought of as being interchangeable for most of our purposes.

2.2 Two Ways to Distribute Flexibility

Within the general class of multi-server multi-type queueing models described in Section 2.1, we shall further focus on two families of partially flexible systems, distinguished by how flexibility is being *measured* and *distributed* across the processing resources.

Family 1: Sparse Flexibility. The first family, illustrated in Figure 2-1, aims to capture situations where the system's flexibility spreads across the processing resources, so that *all* servers are *partially* flexible to some degree. In particular, each server is capable of processing *a few* job types, and the system's level of flexibility is measured by the average number of job types a server is able to serve, or, the average degree, d, of the bipartite graph that connects the queues and servers. A partially flexible system is one where d is significantly smaller than the system size, n — hence the name "sparse flexibility".

Family 2: Partial Pooling. The second family aims to model cases where the system's flexibility is *concentrated* on a small number of servers. In this family, a fraction, p, of the total processing resources is *fully flexible* (or *pooled*), while the remaining $1-p$ fraction of the resources is *inflexible* and is dedicated to serving specific demand types. The system's level of flexibility is captured by the fraction of fully flexible servers, p. A partially flexible system corresponds to one where the

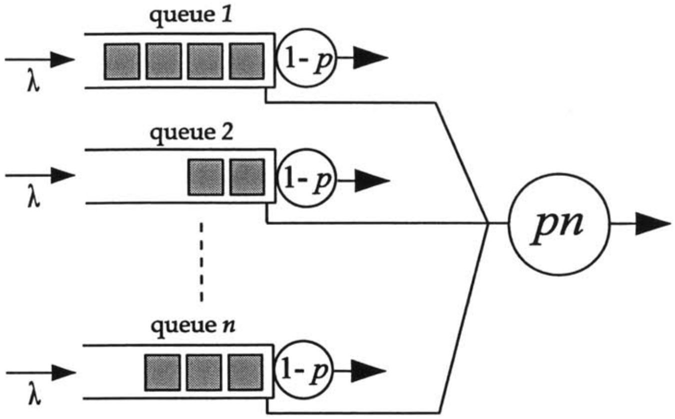

Figure 2-2: A Partial Pooling architecture, where a fraction p of processing resources is fully flexible, and the remaining $1-p$ fraction is dedicated.

value of p is small but positive, and hence a fraction of the resources are "partially pooled." The Partial Pooling model was first proposed and analyzed in [84, 91].

A typical system in the Partial Pooling family is illustrated in Figure 2-2, which consists of one flexible (central) server running at speed pn, and n inflexible (local) servers running at speed $1-p$. One may wonder whether the Partial Pooling model in Figure 2-2 can be captured by the multi-server multi-type model that we have seen in Section 2.1 and Figure 2-1. To see the relation between the two models, fix $p \in (0,1)$, and let $\delta > 0$ be a constant so that both p and $1-p$ are integer multiples of δ. Suppose that the n units of total processing resources consist of n/δ servers, each running at rate δ. Note that this is essentially the same as the original system (Figure 2-1), except that the servers now run at speed δ, instead of 1. Then, under the Service Token model in Section 2.1, it is not difficult to see that the system in Figure 2-2 produces the same dynamics as the case where:

(a) a fraction p of the n/δ servers are fully flexible, forming a *resource pool*, and

(b) the remaining $1-p$ fraction of the servers are inflexible, where each job type is served by $(1-p)/\delta$ such inflexible servers.

The Sparse Flexibility family will be analyzed in Chapters 3 and 4, and the Partial Pooling family in Chapters 5 through 7. While both families belong to the general class of multi-server multi-type queueing systems, they have fairly distinct structural properties as well as stochastic dynamics. As a result, our research will also have different emphases, among which the most prominent distinction lies between **architecture** versus **information**.

1. Within the Sparse Flexibility family, we will mainly be studying the *architectural* question of how the flexibility of different resources should be arranged, in a way that delivers the most desirable capacity and delay performance. In a large part, this is because in the Sparse Flexibility family every server can serve a different set of job types (cf. Figure 2-1), and hence it encompasses a considerably larger set of flexible architectures than the Partial Pooling family, which is parameterized by a single parameter, p.

2. With the Partial Pooling family, we will mainly be exploring the topic of *information*: what does the system operator *know* when making *dynamic* resource allocation and scheduling decisions, and how does that knowledge impact performance? The inherent symmetry in the Partial Pooling family provides sufficient structure for our models to be tractable, and enables us to drive sharp bounds and scaling laws. On the other hand, the mixture of flexible and dedicated resources manifests itself in queueing dynamics that are considerably richer and more complex than either fully flexible or inflexible systems, which allows us to obtain interesting, and deeper, insights on the relationship between information and the system's dynamic behavior.

2.3 Notation

We now introduce some of the terminology that will be used throughout the report. We shall postpone the definition of symbols and notation that are more restricted to a particular topic, which will be introduced in the corresponding chapters.

We will denote by $\mathbb{N}$, $\mathbb{Z}_+$, and $\mathbb{R}_+$, the sets of natural numbers, non-negative integers, and non-negative reals, respectively. The following short-hand notation for asymptotic comparisons will be used; here f and g are positive functions, and L is certain limit of interest in the set of extended reals, $\mathbb{R} \cup \{-\infty, +\infty\}$:

1. $f(x) \lesssim g(x)$ for $f(x) = \mathcal{O}(g(x))$, and $f(x) \gtrsim g(x)$ for $f(x) = \Omega(g(x))$;

2. $f(x) \gg g(x)$ for $\liminf_{x \to L} f(x)/g(x) = \infty$, and $\ll$ is defined analogously.

3. $f(x) \sim g(x)$ for $\lim_{x \to L} f(x)/g(x) = 1$.

Whenever possible, we will use upper-case letters for random variables, and lower-case letters for deterministic values. Let X and Y be two random variables.

1. $X \overset{d}{=} Y$ means that X and Y have the same distribution.

2. Suppose X and Y are real-valued. Then $X \preccurlyeq Y$ means that X is stochastically dominated by Y, i.e.,

$$\mathbb{P}(X > c) \le \mathbb{P}(Y > c), \quad \forall c \in \mathbb{R}. \tag{2.1}$$

We will use Expo(λ), Geo(p), Bino(n,p) as short-hands for the exponential, geometric and binomial distributions with the standard parameters, respectively. We will minimize the use of floor and ceiling throughout the report to avoid the cluttering of notation, and thus assume that all values of interest are appropriately rounded up or down to an integer, whenever doing so does not cause ambiguity or confusion.

Chapter 3

Queueing System Architectures with Limited Flexibility

In this chapter and the next, we will explore the design and operation of *partially flexible architectures.* We will study a multi-server multi-type queueing model, described in Section 2.1, with n *flexible* servers and n queues, connected through a bipartite graph, where the level of flexibility is captured by the graph's average degree, d_n. Applications in content replication in data centers, skill-based routing in call centers, and flexible supply chains are among our main motivations.

We focus on the scaling regime where the system size n tends to infinity, while the overall traffic intensity stays fixed. We show that a large capacity region and diminishing queueing delay are simultaneously achievable even under very limited flexibility ($d_n \ll n$). We also explore and compare different flexibility architectures and scheduling algorithms, and examine the extent to which the objectives of a favorable capacity region and delay are possible for each architecture.[1]

[1] A preliminary version of this chapter appeared at Sigmetrics 2013, [85].

3.1 Introduction

The class of multi-server multi-type queueing models, described in Section 2.1, lies at the heart of a number of modern queueing networks. In these systems, the designer is confronted with the problem of allocating processing resources (manufacturing plants, web servers, or call-center staff) to meet multiple types of demands that arrive dynamically over time (orders, data queries, or customer inquiries). It is often the case that a *fully flexible* or *completely resource-pooled* system, where every unit of processing resource is capable of serving all types of demands, delivers the best possible performance. Our inquiry is, however, motivated by the unfortunate reality that such full flexibility is often infeasible due to overwhelming implementation costs (in the case of a data center) or human skill limitations (in the case of a skill-based call center).

What are the key benefits of flexibility and resource pooling in such queueing networks? Can we harness the same benefits even when the degree of flexibility is *limited*, and how should the network be designed and operated? These are the main questions that we wish to address. While these questions can be approached from a few different angles, we will focus on the metrics of *capacity region* and *expected queueing delay*; the former measures the system's *robustness* against *demand uncertainties*, i.e., when the arrival rates for different demand types are unknown or likely to fluctuate over time, while the latter is a direct reflection of *performance*. Our main message is positive: in the regime where the system size is large, improvements in both the capacity region and delay are *jointly achievable* even under very limited flexibility, given a proper choice of the architecture (interconnection topology) and scheduling policy.

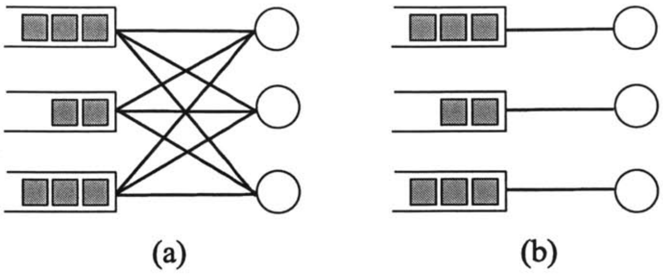

Figure 3-1: Extreme cases of flexibility: $d_n = n$ versus $d_n = 1$.

Benefits of Full Flexibility. We begin by illustrating the benefits of flexibility and resource pooling using two simple examples, which have been alluded to in the introductory chapter (cf. Figure 1-1 in Section 1.1). Consider a system of n servers, each running at rate 1, and n queues, where each queue stores jobs of a particular demand type. For each $i \in \{1, \ldots, n\}$, queue i receives an independent Poisson arrival stream of rate λ_i. The average arrival rate $\frac{1}{n}\sum_{i=1}^{n}\lambda_i$ is denoted by ρ, and is referred to as the *traffic intensity.* The sizes of all jobs are independent and exponentially distributed with mean 1.

For the remainder of this chapter, we will use a measure of flexibility given by the average number of servers that a demand type can receive service from, denoted by d_n. Let us consider the two extreme cases: a fully flexible system, with $d_n = n$ (Figure 3-1(a)), and an inflexible system, with $d_n = 1$ (Figure 3-1(b)). Fixing the traffic intensity $\rho < 1$, and letting the system size, n, tend to infinity, we observe the following qualitative benefits of full flexibility:

1. Large Capacity Region. In the fully flexible case and under any work-conserving scheduling policy[2], the *collection* of all jobs in the system evolves as an $M/M/n$

[2] A work-conserving policy mandates that a server be always busy whenever there is at least one job in the queues to which it is connected.

queue, with arrival rate $\sum_{i=1}^{n} \lambda_i$ and service rate n. It is easy to see that the system is stable for all arrival rates that satisfy, $\sum_{i=1}^{n} \lambda_i < n$, whereas in the inflexible system, since all $M/M/1$ queues operate independently, we must have $\lambda_i < 1$, for all i, in order to achieve stability. Comparing the two, we see that the fully flexible system attains a much larger capacity region, and is hence more robust to uncertainties or changes in the arrival rates.

2. Diminishing Delay. Let W be the steady-state average waiting time in queue (time from entering the queue to the initiation of service). As mentioned earlier, the total number jobs in the system for the fully flexible case evolves as an $M/M/n$ queue with traffic intensity $\rho < 1$. It is not difficult to verify that for any fixed value of ρ, the expected total number of jobs in the queues is *bounded* by a constant independent of n, and hence the expected waiting time in queue satisfies $\mathbb{E}(W) \to 0$, as $n \to \infty$.[3] In contrast, the inflexible system is simply a collection of n *unrelated* $M/M/1$ queues, and hence the expected waiting time is $\mathbb{E}(W) = \frac{\rho}{1-\rho} > 0$, for all n. In other words, the expected delay *diminishes* in a fully flexible system, as the system size increases, but stays bounded away from zero in the inflexible case.

Preview of Main Results. Will the above benefits continue to be present if the system is no longer fully flexible, that is, if $d_n \ll n$? The main results of the chapter show that a large capacity region and a diminishing delay can still be *simultaneously achieved*, even when the amount of flexibility in the system is limited ($d_n \ll n$), and the extent to which this is possible depends largely on the *architecture* of choice (c.f., Table 3.1). However, when flexibility is scarce, the architecture and scheduling

[3] The diminishing expected waiting time follows from the bounded expected total number of jobs in steady-state, the fact that the total arrival rate is ρn, which goes to infinity as $n \to \infty$, and Little's Law.

policy need be chosen with care: our solutions are based on connectivity topologies that range from simple Modular architectures to those based on Erdös-Rényi random bipartite graphs and expander graphs, combined with scheduling policies that range from greedy policies (for Modular architectures) to more sophisticated virtual-queue-based scheduling rules that utilize job-to-server assignments on the connectivity graph in a dynamic fashion (for architectures based on random graphs).

3.1.1 Motivating Applications

We describe here several motivating applications for our model; Figure 3-2 illustrates the overall architecture that they share. **Content replication** is commonly used in data centers for bandwidth intensive operations such as database queries [76] or video streaming [53], by hosting the same piece of content on multiple servers. Here, a server corresponds to a physical machine in the data center, and each queue stores incoming demands for a particular piece of content (e.g., a video clip). A server j is connected to queue i if there is a copy of content i on server j, and d_n corresponds to the average number of replicas per piece of content across the network. Similar structures also arise in **skill-based routing (SBR) in call centers**, where agents (servers) are assigned to answer calls from different categories (queues) based on their domains of expertise [88], and in **process-flexible supply chains** [22, 38, 46, 47, 73], where each plant (server) is capable of producing multiple product types (queues). In many of these applications, demand rates can be unpredictable and may change significantly over time; for instance, unexpected "spikes" in demand traffic are common in modern data centers [48]. These demand uncertainties make *robustness* an important criterion for system design. These practical concerns have been our primary motivation for studying the *joint trade-off* between robustness, performance, and the level of flexibility.

3.1.2 Related Research

Bipartite graphs provide a natural model for capturing the relationships between demand types and service resources. It is well known in the supply chain literature that limited flexibility, corresponding to a sparse bipartite graph, can be surprisingly effective in resource allocation even when compared to a fully flexible system [22, 38, 46, 47, 73]. The use of sparse random graphs or expanders as flexibility structures to improve robustness has recently been studied in [23] in the context of supply chains, and in [53] for content replication. Similar to the robustness results reported in this chapter, both works show that random graphs or expanders can accommodate a large set of demand rates. However, in contrast to our work, nearly all analytical results in this literature focus on static allocation problems, where one tries to match supply with demand in a single slot, as opposed to our model, where resource allocation decisions need to be made dynamically over time.

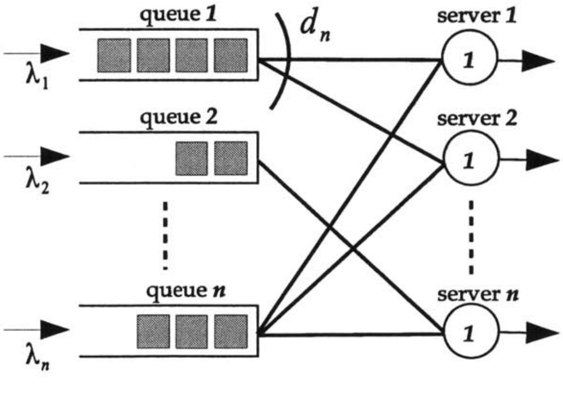

Figure 3-2: A processing network with n queues and n servers.

In the queueing theory literature, the models that we consider fall under the umbrella of the so-called multi-class multi-server systems, where a set of servers are connected to a set of queues through a bipartite graph. Under these (and similar) settings, complete resource pooling (full flexibility) is known to improve system performance [11, 41, 59]. However, much less is known when only limited flexibility is available: systems with a non-trivial connectivity graph have proven to be extremely difficult to analyze, even under seemingly simple scheduling policies (e.g, first-come-first-serve) [81, 86]. Simulations in [88] show empirically that limited cross-training can be highly effective in a large call center under a skill-based routing algorithm. Using a very different set of modeling assumptions, [10] proposes a specific chaining structure with limited flexibility, which is shown to perform well under heavy traffic. Closer to the spirit of the current work is [84], which studies a partially flexible system where a fraction $p > 0$ of all processing resources are fully flexible, while the remaining fraction, $1-p$, is dedicated to specific demand types, and which shows an exponential improvement in delay scaling under heavy-traffic. However, both [10] and [84] focus on the heavy-traffic regime, which is different from the current setting where traffic intensity is assumed to be fixed, and the analytical results in both works apply only to uniform demand rates. Furthermore, with a constant fraction of fully flexible resources, the average degree in [84] scales linearly with the system size n, whereas here we are interested in the case of a much smaller (sub-linear) degree scaling.

At a higher level, our work is focused on the joint trade-off between robustness, delay, and the degree of flexibility in a queueing network, which is much less studied in the existing literature, and especially for networks with a non-trivial interconnection topology.

On the technical end, we build on several existing ideas. The techniques of batching (cf. [67, 83]) and the use of virtual queues (cf. [52, 62]) have appeared in many contexts in queueing theory, but the specific models considered in the literature bear little resemblance to ours. The study of perfect matchings on a random bipartite graph dates back to the seminal work in [29]; while it has become a rich topic in combinatorics, we will refrain from giving a thorough literature survey because only some elementary and standard properties of random graphs are used in the current chapter.

Organization of the Chapter We describe the model in Section 3.2 along with the notation to be used throughout. The main theorems, as well as constructions for the corresponding flexibility architectures and scheduling policies, are stated in Section 3.3. The construction and the analysis associated with the virtual-queue-based scheduling algorithm, designed for the Random Graph architecture, is relatively more complex and will be presented separately, in Chapter 4. We conclude the chapter in Section 3.5 with a further discussion of the results as well as directions for future research.

3.2 Model and Metrics

3.2.1 Queueing Model and Interconnection Toplogies

The Model. We consider a sequence of systems operating in *continuous time*, indexed by the integer n, where the nth system consists of rn queues and n servers, where r is a positive constant that is fixed as n varies (Figure 3-2). To simplify notation, we will mostly focus on the case of $r = 1$ for the remainder of the chapter, while noting that *all results* and arguments in this chapter can be extended to the case of an arbitrary $r > 0$ without significant difficulty.

The *flexible architecture* is represented by an $n \times n$ undirected bipartite graph $g_n = (E, I \cup J)$, where I and J represent the set of queues and servers, respectively, and E the set of edges between them.[4] We will also refer to I and J as the set of left and right vertices, respectively. A server $j \in J$ is *capable* of serving a queue $i \in I$, if and only if $(i, j) \in E$. We will use the following notation.

1. Let $\mathcal{G}_n$ be set of all $n \times n$ bipartite graphs.

2. For $g_n \in \mathcal{G}_n$, let $\deg(g_n)$ be the average degree among the n left vertices. (Since $r = 1$, this is the same as the average degree of the right vertices.)

3. For a subset of vertices, $M \subset I \cup J$, let $g|_M$ be the graph induced by g on the vertices in M.

4. Denote by $\mathcal{N}(i)$ the set of servers in J connected to queue i, and similarly, by $\mathcal{N}(j)$ the set of queues in I connected to server j.

In the nth system, each queue i receives a stream of incoming jobs according to a Poisson process of rate $\lambda_{n,i}$, independent of all other streams, and we define $\lambda_n = (\lambda_{n,1}, \lambda_{n,2}, \ldots, \lambda_{n,n})$, which is the **arrival rate vector**.[5] The sizes of the jobs are exponentially distributed with mean 1, independent from each other and from the arrival processes. All servers are assumed to be running at a constant rate of 1. The system is assumed to be empty at time $t = 0$. Note that, the assumption of exponential service times and uniform-speed servers corresponds to the Service Time model introduced in Section 2.1.

Jobs arriving at queue i can be assigned (immediately, or in the future) to an idle server $j \in \mathcal{N}(i)$ to receive service. The assignment is *binding*, in the sense that once

[4] For notational simplicity, we omit the dependence of E, I, and J on n.

[5] When referring to a specific arrival rate vector, we may omit the dependence on n and write $\lambda = (\lambda_1, \ldots, \lambda_n)$ instead.

the assignment is made, the job cannot be transferred to, or simultaneously receive service from, any other server. Moreover, service is *non-preemptive*, in the sense that once service is initiated for a job, the assigned server has to dedicate its full capacity to this job until its completion.[6] Formally, if a server j has just completed the service of a previous job at time t or is idle, its available actions are: **(a) Serve a new job**: Server j can choose to fetch a job from any queue in $\mathcal{N}(j)$ and immediately start service. The server will remain occupied and take no other actions until the processing of the current job is completed. **(b) Remain idle**: Server j can choose to remain idle. While in the idling state, it will be allowed to initiate a service (Action (a)) at any point in time.

Given the limited set of actions available to the server, the performance of the system is fully determined by a *scheduling policy*, π, which specifies for each server $j \in J$, (a) when to remain idle, and when to serve a new job, and (b) from which queue in $\mathcal{N}(j)$ to fetch a job when initiating a new service. We only allow policies that are causal, in the sense that the decision at time t depends only on the history of the system (arrivals and service completions) up to t. We allow the scheduling policy to be *centralized* (i.e., to have full control over all server actions) based on the knowledge of all *queue lengths* and server states. On the other hand, the policy does *not* observe the actual sizes of the jobs before they are served.

[6] While we restrict ourselves to only binding and non-preemptive scheduling polices in this chapter, other common architectures where (a) a server can serve multiple jobs concurrently (processor sharing), (b) a job can be served by multiple servers concurrently, or (c) jobs sizes are revealed upon entering the system, are clearly more powerful than the current setting, and are therefore capable of implementing the scheduling policy considered in this chapter. As a result, the performance upper bounds developed in this chapter also apply to these more powerful variations.

3.2.2 Performance Metrics

Characterization of Arrival Rates. Throughout the chapter, we will restrict ourselves to arrival rate vectors with average *traffic intensity* ρ, i.e.,

$$\sum_{i=1}^{n} \lambda_i \leq \rho n, \tag{3.1}$$

where $\rho < 1$ is a fixed constant. To quantify the *level of variability* or *uncertainty* of a set of arrival rate vectors, $\mathbf{\Lambda}$, we introduce a *fluctuation parameter*, denoted by u_n, defined as the *maximum arrival rate* to any single queue among arrival rate vectors in $\mathbf{\Lambda}$:

$$u_n = \sup_{\lambda \in \mathbf{\Lambda}} \max_{i \in I} \lambda_i. \tag{3.2}$$

Note that, for a graph with maximum degree d_n, the fluctuation parameter should not exceed d_n, because otherwise at least one queue could be unstable. Therefore, the best we can hope for is a flexible architecture that can accommodate arrival rate vectors with a u_n that is close to d_n. The following condition formally characterizes the range of arrival rate vectors we will be interested in, parameterized by the fluctuation parameter, u_n, and traffic intensity, ρ.

Condition 3.1. (Rate Condition) *Fix $n \geq 1$ and some $u_n > 0$. We say that a (non-negative) arrival rate vector λ satisfies the rate condition if the following hold:*

1. $\max_{1 \leq i \leq n} \lambda_i \leq u_n$.

2. $\sum_{i=1}^{n} \lambda_i \leq \rho n$.

We denote by $\mathbf{\Lambda}_n(u_n)$ the set of all arrival rate vectors that satisfy the above conditions.

Capacity Region. The capacity region for a given architecture is defined as the set of all arrival rate vectors that it can handle. As mentioned in the Introduction, a largcr capacity rcgion indicates that the architecture is more robust against uncertainties or changes in the arrival rates. More formally, we have the following definition.

Definition 3.2 (Feasible Demands and Capacity Region). *Let $G = (I \cup J, E)$ be an $n \times n$ bipartite graph. An arrival rate vector (demand), $\lambda = (\lambda_1, \ldots, \lambda_n)$, is said to be feasible (or admissible), if there exists a flow, $\mathcal{F} = \{f_{ij} : (i,j) \in E\}$, such that*

$$\begin{aligned} \lambda_i = \sum_{j \in \mathcal{N}(i)} f_{ij}, &\quad \forall i \in I, \\ \sum_{i \in \mathcal{N}(j)} f_{ij} < 1, &\quad \forall j \in J, \\ f_{ij} \geq 0, &\quad \forall (i,j) \in E. \end{aligned} \tag{3.3}$$

In this case, we say that the flow $\mathcal{F}$ satisfies *the demand λ. The* capacity region *of G, denoted by $\mathbf{R}(G)$, is defined as set of all feasible demand vectors of G.*

For thc remainder of the chapter, we will use the fluctuation parameter u_n (Condition 3.1) to gauge the size of the capacity region of an architecture, $\mathbf{R}(g_n)$. For instance, if $\mathbf{\Lambda}_n(u_n) \subset \mathbf{R}(g_n)$, then the architecture g_n is able to handle all arrival rate vectors with a maximum arrival rate of u_n.

Diminishing Delay. We define the *expected average delay*, $\mathbb{E}(W|\lambda, g, \pi)$, as the expected qucueing delay under the arrival rate vector λ, flexible architecture g, and scheduling policy π. Specifically, let $W_{i,m}$ be the waiting time in queue experienced by the mth job arriving to queue i, and let

$$\mathbb{E}(W|\lambda, g, \pi) = \frac{1}{\sum_{i\in I}\lambda_i}\sum_{i\in I}\lambda_i\mathbb{E}(W_i), \tag{3.4}$$

where $\mathbb{E}(W_i) = \limsup_{m\to\infty}\mathbb{E}(W_{i,m})$.[7] For the remainder of the chapter, we may omit the mentioning of g and π, and write $\mathbb{E}(W|\lambda)$ instead to emphasize the dependence of delay on the arrival rate.

The delay performance of the system is measured by the following criteria: (a) for what ranges of arrival rates, λ, is *diminishing delay* achieved as the system size tends to infinity, i.e., $\mathbb{E}(W|\lambda) \to 0$ as $n \to \infty$, and (b) at what *speed* does the delay diminish, as a function of n.

3.3 Main Results: Capacity and Delay Performance for Flexible Architectures

The statements of our main theorems are given in this section, and focus on the performance of three flexible architectures: Random Graph, Modular and Expanded Modular. For each case, we also provide the scheduling policy associated with the flexible architecture.

Our results show that *all three* flexible architectures are able to achieve the joint objective of a large capacity region and diminishing delay, under limited flexibility ($d_n \ll n$). However, they do so to different degrees, and the associated scheduling policies also vary in complexity. Below is a high-level summary of our results, and a

[7] Note that $\mathbb{E}(W|\lambda)$ captures a *worst-case* expected waiting time across all jobs in the long run, and is always well defined, even under scheduling policies that do not induce a steady-state distribution.

more complete comparison is given in Table 3.1.

1. The Random Graph architecture is based on an interconnection topology generated by an Erdös-Rényi bipartite random graph. It admits a capacity region that is essentially optimal, with high probability (Theorem 3.5). Using a virtual-queue-based scheduling policy that utilizes dynamic assignments of jobs to servers over the connnectivity graph, we show that one can achieve diminishing delay for "most" arrival rate vectors in the capacity regime.

2. A Modular architecture consists of collection of *separate* small subnetworks, and the queues and servers within each subnetwork are fully connnected. Since the subnetworks are disconnected from one another, a Modular architecture does not admit a large capacity region: there always exists an *infeasible* arrival rate vector even when the fluctuation parameter is of constant order (Theorem 3.7). Nevertheless, we show that with proper randomization in the construction of the subnetworks, a simple greedy scheduling policy is able to deliver diminishing delay for "most" arrival rate vectors with essentially optimal fluctuation parameters, with high probability (Theorem 3.8).

3. The Expanded Modular architecture can be thought of as a combination of the Random Graph and Modular architectures. By construction, it devotes the system's flexibility *separately* in achieving the performance goal of capacity and delay. As a result, the Expanded Modular architecture admits a smaller capacity region compared to that of a Random Modular architecture, but it is able to ensure a diminishing delay for *all* arrival rates, uniformly across the capacity region (Theorem 3.11).

Based on these considerations, the Random Graph architecture appears to be best performing one. Whether some even better performance is achievable, however, remains an open problem; cf. the "Ideal graph" in Table 3.1 and Conjecture 3.15 in Section 3.5.

Flexible architectures	Rate Conditions	Capacity	Delay
Random Graph (w.h.p.) (Theorems 3.5, 3.6)	$d_n \gtrsim \ln n$, $u_n \lesssim d_n/\ln n$	Good for all λ	Good for most λ with[a] $\mathbb{E}(W) \lesssim \ln^2 n/d_n$, $d_n \gg \ln^2 n$ Unknown whether "for all λ"
Modular (Theorems 3.7, 3.9)	$d_n \gg 1$, $u_n > 1$	Bad for many λ (even if $u_n \lesssim 1$)	Good for uniform λ, with $\mathbb{E}(W) \lesssim \exp(-c \cdot d_n)$
Random Modular (w.h.p.) (Theorems 3.8, 3.9)	$d_n \gtrsim \ln n$, $u_n \lesssim d_n/\ln n$	Good for most λ, Bad for some λ	Good for most λ, with $\mathbb{E}(W) \lesssim \exp(-c \cdot d_n)$, Bad for some λ
Expanded Modular with $d_n = d_1(n) \cdot d_2(n)$ (Theorem 3.11)	$d_n \gg 1$, $u_n \lesssim d_1(n)$	Good for all λ	Good for all λ, with slower rate $\mathbb{E}(W) \lesssim 1/d_2(n)$
Ideal Graph (Conjecture 3.15)	$d_n \gg 1$, $u_n \lesssim d_n$	Good for all λ	Good for all λ, with $\mathbb{E}(W) \lesssim \exp(-c \cdot d_n)$

Table 3.1: This table summarizes and compares the flexibility architectures that we study, along with the metrics of capacity and delay. We say that capacity is "good" for λ if λ falls within the capacity region of the architecture, and that delay is "good" if the expected delay is vanishingly small for large n. When describing the size of the set of λ for which a statement applies, we use the following (progressively weaker) quantifiers:

1. "for all" means that the statement holds for all $\lambda \in \mathbf{\Lambda}_n(u_n)$;

2. "for most" means that the statement holds with high probability when λ is drawn from an arbitrary distribution over $\mathbf{\Lambda}_n(u_n)$, independently from any randomization in the construction of the flexibility architecture;

3. "for many" means that the statement is true for a non-empty set of λs, even when the degree of fluctuation u_n is small or constant.

The label "w.h.p." means that all statements in the corresponding row hold with high probability with respect to the randomness in generating the flexibility architecture. The statement marked "a" is based on an alternative interpretation of Theorem 3.6, given in Eq. (3.9).

3.3.1 Preliminaries

The notion of an expander graph will be used in some of our constructions.

Definition 3.3. *An $n \times n$ bipartite graph $(I \cup J, E)$ is an (α, β)-expander, if for all $S \subset I$ that satisfy $|S| \leq \alpha n$, we have that $|\mathcal{N}(S)| \geq \beta |S|$, where $\mathcal{N}(S) = \bigcup_{i \in S} \mathcal{N}(s)$.*

The usefulness of expanders in our context comes from the following lemma, which relates an expander's expansion parameters to the size of its capacity region, measured by the fluctuation parameter, u_n. The proof is elementary and is given in Appendix A.1.1.

Lemma 3.4 (Capacity of Expanders). *Fix $n \in \mathbb{N}$ and $\gamma \in [\rho, 1)$. Suppose that g_n is an $(\gamma / u_n, u_n)$-expander. Then $\mathbf{\Lambda}_n(u_n) \subset \mathbf{R}(g_n)$.*

3.3.2 Random Graph Architectures

The Random Graph architecture is an $n \times n$ Erdös-Rényi random bipartite graph G, where each of the n^2 edges is present with probability p, independently of all other edges. We will refer to it as an (n, p) random bipartite graph, and use $\mathbb{P}_{n,p}(\cdot)$ to denote the corresponding probability measure on $\mathcal{G}_n$, i.e.,

$$\mathbb{P}_{n,p}(g) = p^{|E|} (1-p)^{n^2 - |E|}, \quad \forall g \in \mathcal{G}_n. \tag{3.5}$$

Construction of the Architecture. We will simply use a $(n, d_n/n)$ random bipartite graph, so that each queue-server pair is connected with probability d_n/n.[8]

[8] Note that even though the *process* for generating the interconnection topology involves randomization, the topology itself remains *fixed* once generated.

Scheduling Policy. We will employ a class of virtual-queue-based scheduling policies, which chooses job-to-server assignments in a dynamic fashion. The details of the scheduling policy are described in the proof of Theorem 3.6 in Chapter 4.

The following theorem states that with high probability, the Random Graph architecture has a large capacity region. This stems from the fact that a random graph is also a good expander with high probability.

Theorem 3.5 (Capacity of Random Graph Architectures). *Suppose that* $d_n \geq \frac{2}{1-\rho}\ln n$*, and* $u_n \leq \frac{1-\rho}{8} d_n / \ln n$*. Let* G_n *be an* (n,p) *random bipartite graph, with* $p = d_n/n$*. We have*

$$\lim_{n\to\infty} \mathbb{P}_{n,p}\left(\mathbf{\Lambda}_n(u_n) \subset \mathbf{R}(G_n)\right) = 1. \tag{3.6}$$

Proof. See Section 3.4.1. □

Note that when $d_n \gtrsim \ln n$, a (n,p) bipartite random graph with $p = d_n/n$ has average degree of order d_n with high probability. Therefore, Theorem 3.5 shows that, with high probability, the size of the capacity region of the Random Graph architecture is the best possible, within a logarithmic factor, because the fluctuation parameter, u_n, can be of order $\mathcal{O}(d_n/\ln n)$.

We next turn to the delay in a Random Graph architecture. The following theorem states that, when $u_n \lesssim d_n/\ln n$, for any arrival rate vector $\lambda_n \in \mathbf{\Lambda}_n(u_n)$, the Random Graph architecture can achieve a diminishingly small delay, with high probability. The first part of the theorem states that if λ_n is known, then a "good graph" with desirable delay performance can be constructed. The second part states that the random graph construction will be able to produce such a "good graph" with high probability. The proof of the theorem will be presented in Chapter 4.

Theorem 3.6 (Delay of Random Graph Architectures). *Fix* $\gamma > 0$*, and* $n \geq 1$*.*

Suppose that $d_n \geq \frac{4}{1-\rho} \ln^{2.1} n$, *and* $u_n \leq \frac{1-\rho}{8} d_n / \ln n$.[9]

(a) For any $\lambda_n \in \mathbf{\Lambda}_n(u_n)$, *there exists a bipartite graph,* $g_n \in \mathcal{G}_n$, *with* $\deg(g_n) \leq (1+\gamma) d_n$, *and a scheduling policy,* π_n, *under which* [10]

$$\mathbb{E}(W \mid \lambda_n) \leq K \frac{\ln^2 n}{d_n}, \tag{3.7}$$

where $K > 0$ *is a constant independent of* n, g_n, *and* λ_n.

(b) For any $\lambda_n \in \mathbf{\Lambda}_n(u_n)$, *there exists* $\mathcal{H}_n \subset \mathcal{G}_n$, *with*[11]

$$\mathbb{P}_{n, d_n/n}(\mathcal{H}_n) \geq 1 - \delta_n, \tag{3.8}$$

such that there exists a scheduling policy π_n, *under which Eq.* (3.7) *holds for every* $g_n \in \mathcal{H}_n$. *Here* $\{\delta_n\}_{n\geq 1}$ *is a sequence of non-increasing constants with* $\lim_{n\to\infty} \delta_n = 0$.

(c) The scheduling policy, π_n, *only depends on* g_n *and an upper bound on the traffic intensity,* ρ. *It has no additional dependencies on the arrival rate vector* λ_n.

We can interpret Theorem 3.6 as a statement for "most" arrival rate vectors in $\mathbf{\Lambda}_n(u_n)$, as follows. Consider the case where the flexible architecture G is drawn according to the probability measure $\mathbb{P}_{n,d_n/n}$, and the arrival rate vector λ_n is chosen at random, according to some arbitrary probability measure μ_n on $\mathbf{\Lambda}_n(u_n)$, but still independently from G. It can be shown, through an easy application of Fubini's

[9] The theorem holds if $d_n \gg \ln^2 n$. Here, we have chosen to let $d_n \geq \frac{4}{1-\rho} \ln^{2.1} n$ for concreteness.

[10] The choice of g_n can depend on λ_n.

[11] $\mathbb{P}_{n,p}$ is the probability measure induced by an (n,p) random bipartite graph, defined in Eq. (3.5).

Theorem, that

$$\left(\mathbb{P}_{n,\frac{d_n}{n}} \times \mu_n\right)\left(\mathbb{E}\left(W|\lambda_n\right) \leq K\tfrac{\ln^2 n}{d_n}, \text{ under } G \text{ and } \pi_n\right) \to 1, \tag{3.9}$$

for any sequence of measures, $\{\mu_n\}_{n\geq 1}$, where each μ_n has support on $\mathbf{\Lambda}_n(u_n)$, and $\times$ is used to denote product measure. In other words, with high probability, the Random Graph architecture yields small delay for "most" $\lambda_n \in \mathbf{\Lambda}_n(u_n)$ (as in Table 3.1).

Remark. The delay characterization given by Theorem 3.6 is weaker than that for the capacity in Theorem 3.5. We do not know whether it is possible to find an architecture, under which we can guarantee a small delay for *all* $\lambda \in \mathbf{\Lambda}_n(u_n)$ (see Conjecture 3.15 in Section 3.5).

The requirement of $d_n \gtrsim \ln^{2.1} n$ in Theorem 3.6 is stronger than that of Theorem 3.5 by a $\ln n$ factor. This is, however, not a hard limitation. Using a more refined analysis, one can extend Theorem 3.6 to showing that, for any $\theta \in [0,1)$, it is possible to achieve a delay scaling of order

$$\mathbb{E}\left(W \mid \lambda_n\right) \lesssim \frac{\ln^{2-\theta} n}{d_n}, \tag{3.10}$$

whenever $d_n \gg \ln^{2-\theta} n$ and $u_n \lesssim d_n/\ln^{1+\theta} n$. While this extension provides an additional trade-off among the system parameters, it does not change the qualitative conclusions of Theorem 3.6, and we have hence excluded it from the statement of the theorem for simplicity.

3.3.3 Modular Architectures

In a Modular architecture, the designer partitions the network into n/d_n *separate* sub-networks. Each sub-network consists of d_n queues and servers that are fully connected (Figure 3-3), but disconnected from queues and servers in other subnetworks.

Construction of the Architecture. More formally, the construction is as follows.

1. Partition the set of servers, J, into n/d_n clusters of d_n servers each, in some arbitrary manner (e.g., assign d_n servers to the first cluster, the next d_n servers to the second cluster, etc). Let $s(j)$ be the index of the cluster to which server j belongs.

2. Let $\sigma_n : \{1, \ldots, n\} \to \{1, \ldots, n/d_n\}$ be a partition of the set of queues, I, into n/d_n queue clusters, so that each cluster has exactly d_n elements. That is, $\sigma_n^{-1}(q)$ has cardinality d_n for every $q \in \{1, \ldots, n/d_n\}$. Let $q(i)$ be the index of the cluster to which queue i belongs.

3. To construct the interconnection topology, we connect queue i to server j if they belong to queue and server clusters with the same index, i.e., $s(j) = q(i)$. A pair of queue and server clusters with the same index will be referred to as a subnetwork.

Note that any choice of σ_n yields an isomorphic architecture. In the case where σ_n is drawn uniformly at random from the set of possible partitions, we call the resulting topology a *Random Modular* architecture. Note also that by construction, the degree of all nodes in a Modular architecture is equal to the size of the cluster, d_n.

Scheduling Policy. We will use a simple greedy policy that is equivalent to running each subnetwork as an $M/M/d_n$ queue. When server j become available, it starts serving a job from any non-empty queue in $\mathcal{N}(j)$. Similarly, when a job arrives at queue i, it is immediately served by an arbitrary idle server in $\mathcal{N}(i)$ if such a server exists, and waits in queue i, otherwise.

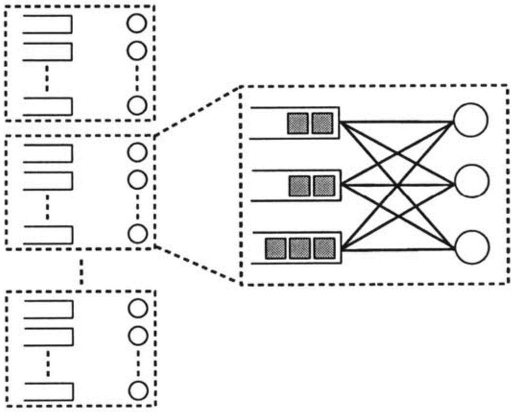

Figure 3-3: A Modular architecture consisting of n/d_n subnetworks, each with d_n queues and servers. Within each subnetwork, all servers are connected to all queues.

Our first result shows that a Modular architecture does not have a large capacity region in the worst-case sense: for any partition σ_n, there always exists an inadmissible arrival rate vector, even if u_n is small, of order $\mathcal{O}(1)$.

Theorem 3.7 (Capacity for Deteministic Modular Architectures). *Fix $n \geq 1$. Suppose that $d_n \leq \frac{\rho}{2}n$, and $u_n > 1$. Let g_n be a Modular architecture associated with the permutation σ_n. Then, there exists $\lambda \in \mathbf{\Lambda}_n(u_n)$ such that $\lambda \notin \mathbf{R}(g_n)$.*

Proof. See Section 3.4.2.

However, if we are willing to consider a weaker characterization of the capacity region, the next theorem states that the Random Modular can handle any arrival rate vector, with high probability, if the fluctuation parameter, u_n, is of order $\mathcal{O}(d_n/\ln n)$, but no more than that.

Theorem 3.8 (Capacity of Random Modular Architectures). *Let σ_n be drawn uniformly at random from the set of all partitions and independent of λ, and let G be the Modular architecture generated by σ_n. Suppose that $d_n \geq c_1 \ln n$, for some $c_1 > 0$.*

1. *There exists $c_2 > 0$, so that if $u_n \leq c_2 d_n/\ln n$, then*

$$\lim_{n\to\infty} \inf_{\lambda\in\Lambda_n(u_n)} \mathbb{P}_G(\lambda \in \mathbf{R}(G)) = 1. \tag{3.11}$$

2. *Conversely, for any $c_1 > 0$, there exists $c_3 > 0$, such that if $u_n \geq c_3 d_n/\ln n$ and $d_n \leq n^{0.3}$, then*

$$\lim_{n\to\infty} \inf_{\lambda\in\Lambda_n(u_n)} \mathbb{P}_G(\lambda \in \mathbf{R}(G)) = 0, \tag{3.12}$$

Proof. See Section 3.4.3.

We now turn to delay. The following theorem shows that in any Modular architecture, delay is vanishingly small for essentially all arrival rate vectors in the capacity region.

Theorem 3.9 (Delay of Modular Architectures). *Fix some $\gamma \in (0,1)$, and let g_n be a Modular architecture. Suppose that $\lambda \in \mathbf{R}(g_n)$. Then, there exists a constant $c > 0$, independent of n, so that*

$$\mathbb{E}(W \mid \gamma\lambda_n) \lesssim \exp(-c \cdot d_n). \tag{3.13}$$

Proof. See Section 3.4.4.

3.3.4 Expanded Modular Architectures

The Expanded Modular architecture combines the features of a Modular architecture and an expander graph via a graph product. However, we will start by providing the more general version of the construction, without an explicit use of expander graphs.

Construction of the Architecture. We first express the average degree as a product, $d_n = d_1(n) \cdot d_2(n)$, where the magnitudes of $d_1(n)$ and $d_2(n)$ relative to each other are a design choice. The architecture is constructed as follows.

1. Partition I and J into equal-sized clusters of size $d_1(n)$. We will refer to the index set of the queue and server clusters as $\mathcal{Q}$ and $\mathcal{S}$, respectively. For all $i \in I$ and $j \in J$, denote by $q(i) \in \mathcal{Q}$ and $s(j) \in \mathcal{S}$ the indices of the queue and server clusters to which i and j belong, respectively.

2. Let g_n^e be a bipartite graph of maximum degree $d_2(n)$ defined on the set of queue and server clusters, $\mathcal{Q} \cup \mathcal{S}$. Let E^e be the set of edges of g_n^e.

3. To construct the interconnection topology $g_n = (I \cup J, E)$, let $(i, j) \in E$ if and only if their corresponding queue and server clusters are connected in g_n^e, i.e., $(q(i), s(j)) \in E^e$.

Note that by the above construction, each queue is connected to at most $d_2(n)$ server clusters through g_n^e, and within each connected cluster, $d_1(n)$ servers. Therefore, the maximum degree of g_n is at most $d_1(n) \cdot d_2(n) = d_n$.

Scheduling Policy. The scheduling policy consists of two stages, and the policy requires the knowledge of the arrival rate vector, λ. The computation in the first stage is performed only once for any given λ, while the second stage is repeated throughout the operation of the system. We assume that $\lambda \in \mathbf{R}(g_n)$.

1. Compute a feasible flow, $\{f_{q,s}\}_{(q,s)\in E^e}$, over the graph g_n^e, where the demand at each queue cluster $q \in \mathcal{Q}$ is equal to $\kappa_q = \sum_{i\in q} \lambda_i$, and the supply at each server cluster $s \in \mathcal{S}$ is equal to $\frac{1+\rho}{2} d_1(n)$. Note that such a flow exists as long as $\lambda \in \mathbf{R}(g_n)$. Denote by $f_{q,s}$ the total rate of flow from the queue cluster q to the server cluster s.

2. When a server becomes available, it chooses a neighboring queue cluster (w.r.t. the topology of g_n^e) with probability roughly proportional to the flow between the clusters. In particular, a server in cluster s chooses the queue cluster q with probability

$$p_{s,q} = \frac{f_{q,s}}{\sum_{q'\in\mathcal{N}(s)} f_{q',s}} \cdot \frac{1+\rho}{2} + \frac{1}{\deg(s)} \cdot \frac{1-\rho}{2}, \tag{3.14}$$

where $\deg(s)$ is the degree of s in g_n^e. Within the chosen cluster, the server starts serving a job from an arbitrary non-empty queue, or, if all queues in the cluster are empty, the server initiates an idling period whose length is exponentially distributed with mean 1.

When the graph g_n^e is an **expander graph**, we refer to the topology created via the above procedure as an *Expanded Modular architecture generated by* g_n^e. The following lemma ensures that such expander graphs exist for the range of parameters we are interested in. The lemma is a simple consequence of a standard result on the existence of expander graphs, and its proof is given in Appendix A.1.3.

Lemma 3.10. *Suppose that $d_2(n) \to \infty$ as $n \to \infty$. Let $\beta_n = \frac{1}{2}\left(\ln^{-1}\frac{1}{\rho} + 1\right)^{-1} d_2(n)$, and $\alpha_n = \frac{1+\rho}{2\beta_n}$. There exists $n' > 0$, such that for all $n \geq n'$, there exists an (α_n, β_n)-expander with maximum degree $d_2(n)$.*

Note that an Expanded Modular architecture is constructed as a "product" between an expander graph across the queue and server clusters, and a fully connected graph for each pair of connected clusters. As a result, its performance is also of a "hybrid" nature: the expansion properties of g_n^e guarantee a large capacity region, while a diminishing delay is obtained as a result of the growing size of the server and queue clusters. We summarize this in the following theorem. Here we assume that $d_2(n)$ is sufficiently large so that the expander graph described in Lemma 3.10 exists.

Theorem 3.11 (Capacity and Delay of Expanded Modular Architectures). *Suppose that $d_n = d_1(n) \cdot d_2(n)$, and let $\beta_n = \frac{1}{2}\left(\ln^{-1}\frac{1}{\rho} + 1\right)^{-1} d_2(n)$, and $\alpha_n = \frac{1+\rho}{2} \cdot \frac{1}{\beta_n}$. Let g_n^e be an (α_n, β_n)-expander with maximum degree $d_2(n)$, and let g_n be an Expanded Modular architecture generated by g_n^e. If*

$$u_n \leq \frac{1+\rho}{2}\beta_n = \frac{1+\rho}{4}\left(\ln^{-1}\frac{1}{\rho} + 1\right)^{-1} d_2(n), \tag{3.15}$$

then

$$\sup_{\lambda \in \Lambda_n(u_n)} \mathbb{E}(W|\lambda) \lesssim \frac{c}{d_1(n)}, \tag{3.16}$$

under the scheduling policy described above, where c is a constant that does not depend on n.

Proof. See Section 3.4.5. □

A Capacity-Delay Trade-off. For the Expanded Modular architecture, the relative values of $d_1(n)$ and $d_2(n)$ reflect a design choice: a larger value of $d_2(n)$ ensures a larger capacity region, while a larger value of $d_1(n)$ yields smaller delays. Therefore, while the Expanded Modular architecture is able to provide a strong delay guarantee that applies to *all* arrival rate vectors in $\Lambda_n(u_n)$, it comes at the expense of either a slower rate of diminishing delay (small $d_1(n)$) or a smaller capacity region (small $d_2(n)$).

3.4 Proofs of Main Results

3.4.1 Proof of Theorem 3.5

Proof. The following useful lemma shows that a random graph is w.h.p. an expander graph.

Lemma 3.12. (Expanders from Random Graphs) *Fix $\gamma \in (0,1)$. Let $d_n \geq \frac{1}{1-\gamma}\ln n$, and $\beta(n) = \frac{1-\gamma}{4}d_n/\ln n$. Let G be an $(n, d_n/n)$ random bipartite graph. We have*

$$\lim_{n\to\infty} \mathbb{P}_{n,d_n/n}\left(G \textit{ is a } \left(\tfrac{\gamma}{\beta_n}, \beta_n\right)\textit{-expander}\right) = 1. \tag{3.17}$$

Proof. See Appendix A.1.2.

Let $\gamma = 1 - (1-\rho)/2 > \rho$. We have that

$$\begin{aligned} &\lim_{n\to\infty} \mathbb{P}\left(\lambda \in \mathbf{R}(G_n),\ \forall \lambda \in \Lambda_n(u_n)\right) \\ &\overset{(a)}{\geq} \lim_{n\to\infty} \mathbb{P}\left(G_n \text{ is an } (\gamma/u_n, u_n)\text{-expander}\right) \\ &\overset{(b)}{=} 1, \end{aligned} \tag{3.18}$$

where steps (a) follows from Lemma 3.4, and step (b) from Lemma 3.12, with $\beta_n = u_n$. □

3.4.2 Proof of Theorem 3.7

Proof. Since the arrival rate vector we choose can depend on the architecture, without loss of generality, we assume that servers and queues are clustered in the same manner: server i and queue i belong to the same cluster. Since all servers have capacity 1, and each cluster has exactly d_n servers, it suffices to show that there exists $\lambda = (\lambda_1, \ldots, \lambda_n) \in \Lambda_n(u_n)$, such that the total arrival rate to the first queue cluster exceeds d_n, i.e.,

$$\sum_{1 \leq i \leq d_n} \lambda_i > d_n, \tag{3.19}$$

To this end, consider the vector λ where $\lambda_i = \min\{u_n, 2\}$ for all $i \in \{1, \ldots, d_n\}$ and $\lambda_i = 0$ for all $i \geq d_n + 1$. We have that

$$\max_{1 \leq i \leq n} \lambda_i = \min\{2, u_n\} \leq u_n, \tag{3.20}$$

and

$$\sum_{1 \leq i \leq n} \lambda_i = d_n \min\{2, u_n\} \overset{(a)}{\leq} \frac{\rho}{2} n \cdot 2 = \rho n, \tag{3.21}$$

where step (a) follows from the assumption that $d_n \leq \frac{\rho}{2} n$. Eqs. (3.20) and (3.21) together ensure that $\lambda \in \Lambda_n(u_n)$. Since we had assumed that $u_n > 1$, Eq. (3.19) holds for this λ. We thus have that $\lambda \notin \mathbf{R}(g_n)$, which proves our claim. □

3.4.3 Proof of Theorem 3.8

Proof. **Upper Bound (Eq.** (3.11)**).** We will use the following classical result due to Hoeffding (adapted from Theorem 3 in [44]).

Lemma 3.13. *Fix m and n, so that $0 < m < n$. Let $X_1, X_2, \ldots, X_m$ be drawn uniformly at random from a finite collection $C = \{c_1, \ldots, c_n\}$ without replacement.*

Suppose that $0 \leq c_i \leq b$ *for all* $1 \leq i \leq n$, *and* $\operatorname{Var}(X_1) = \sigma^2$. *Letting* $\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$, *we have*

$$\mathbb{P}\left(\bar{X} \geq \mathbb{E}\left(\bar{X}\right) + t\right) \leq \exp\left(-\frac{mt}{b}\left[\left(1 + \frac{\sigma^2}{bt}\right)\ln\left(1 + \frac{bt}{\sigma^2}\right) - 1\right]\right), \tag{3.22}$$

for all $t \in (0, b)$.

Fix $\epsilon \in \left(0, \frac{1}{\rho} - 1\right)$ and $k \in \{1, \ldots, n/d_n\}$. Let $A_k \subset I$ be the set of d_n queues to which the servers $kd_n + 1$ through $(k+1)d_n$ are connected under the architecture G, generated by the partition σ_n. Define the event E_k as

$$E_k = \left\{ \sum_{i \in A_k} \lambda_i > (1 + \epsilon)\rho d_n \right\}. \tag{3.23}$$

Since σ_n is drawn uniformly at random from all possible partitions, it is not difficult to see that $\sum_{i \in A_k} \lambda_i \overset{d}{=} \sum_{i=1}^{d_n} X_i$, where $X_1, X_2, \ldots, X_m$ are m elements drawn uniformly at random without replacement from the set $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Applying Lemma 3.13, with $m = d_n$ and $b = u_n$ we have that

$$\begin{aligned}
\mathbb{P}(E_1) =& \mathbb{P}\left(\sum_{i=1}^{d_n} X_i > (1+\epsilon)\rho d_n\right) \\
\overset{(a)}{\leq}& \mathbb{P}\left(\frac{1}{d_n}\sum_{i=1}^{d_n} X_i > \mathbb{E}\left(\frac{1}{d_n}\sum_{i=1}^{d_n} X_i\right) + \epsilon\rho\right) \\
\leq& \exp\left(-\frac{\epsilon\rho d_n}{u_n}\left[\left(1 + \frac{\operatorname{Var}(X_1)}{\epsilon\rho u_n}\right)\ln\left(1 + \frac{\epsilon\rho u_n}{\operatorname{Var}(X_1)}\right) - 1\right]\right),
\end{aligned} \tag{3.24}$$

where the probability is measured with respect to the randomness in G, and where in step (a) we used the fact that

$$\mathbb{E}\left(\sum_{i=1}^{d_n} X_i\right) = \sum_{i=1}^{d_n} \mathbb{E}(X_i) = d_n \mathbb{E}(X_1) = d_n\left(\frac{1}{n}\sum_{i=1}^{n} \lambda_i\right) \leq \rho d_n. \tag{3.25}$$

It is not difficult to show that, if $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbf{\Lambda}_n(u_n)$, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n} \lambda_i \le \rho, \text{ and } \max_{1\le i\le n} \lambda_i \le u_n, \tag{3.26}$$

then the value of $\mathrm{Var}\,(X_1)$ is upper-bounded by the setting where

$$\lambda_i = \begin{cases} u_n, & \text{if } 1 \le i \le \frac{\rho n}{u_n} \\ 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

This implics that

$$\mathrm{Var}\,(X_1) \le \mathbb{E}(X_1^2) = \frac{\rho}{u_n} u_n^2 = \rho u_n. \tag{3.28}$$

Combining Eqs. (3.24) and (3.28), and the fact that the right-hand-side of (3.24) is non-decreasing in $\mathrm{Var}\,(X_1)$, we have that there exists $\theta > 0$, so that for all $\epsilon \in (0, \theta)$,

$$\begin{aligned} \mathbb{P}(E_1) &\le \exp\left(-\frac{\epsilon \rho d_n}{u_n}\left[\left(1+\frac{1}{\epsilon}\right)\ln(1+\epsilon) - 1\right]\right) \\ &\overset{(a)}{\le} \exp\left(-\frac{\rho}{3}\cdot\frac{\epsilon^2 d_n}{u_n}\right), \end{aligned} \tag{3.29}$$

where the step (a) follows from the fact that $\left[\left(1+\frac{1}{x}\right)\ln(1+x) - 1\right] \sim \frac{1}{2}x$ as $x \downarrow 0$.

Let $\epsilon = \frac{1}{2}\min\{\frac{1}{\rho} - 1, \theta\}$, and suppose that $u_n \le \frac{\rho\epsilon^2}{6} d_n \ln^{-1} n$ for all $n \in \mathbb{N}$. Combining

Eq. (3.29) with the union bound, we have that

$$\begin{aligned}\mathbb{P}(\lambda \notin \mathbf{R}(G)) \leq& \mathbb{P}\left(\bigcup_{k=1}^{n/d_n} E_k\right) \\ \leq& \sum_{k=1}^{n/d_n} \mathbb{P}(E_k) \\ \leq& \frac{n}{d_n} \exp\left(-\frac{\rho}{3} \cdot \frac{\epsilon^2 d_n}{u_n}\right) \\ \overset{(a)}{\leq}& \frac{n}{d_n} \cdot \frac{1}{n^2} \\ \lesssim& n^{-1}, \end{aligned} \tag{3.30}$$

where step (a) follows from the assumption that $u_n \leq \frac{\rho\epsilon^2}{6} d_n \ln^{-1} n$. Because Eq. (3.30) holds for all $\lambda \in \mathbf{\Lambda}_n(u_n)$, we have proven the upper bound, Eq. (3.11), by letting $c_2 = \rho\epsilon^2/6$.

Lower Bound (Eq. (3.12)**).** For this part of the proof, we will assume that $u_n \geq c_3 d_n \ln^{-1} n$, for some $c_3 > 0$. Because we are interested in showing the lower bound, without loss of generality, we may assume that $u_n \ll n$. Denote by μ_n a probability distribution over $\mathbf{\Lambda}_n(u_n)$. Let λ be a random vector drawn from the distribution μ_n, independent of the randomness in the Random Modular architecture, G. The following basic fact is useful.

Lemma 3.14. *Suppose there exist* $\{\mu_n : n \in \mathbb{N}\}$ *and* $\{a_n : n \in \mathbb{N}\}$, *so that*

$$\mathbb{P}_{\lambda,G}(\lambda \notin \mathbf{R}(G)) \geq a_n, \quad \forall n \in \mathbb{N}. \tag{3.31}$$

Then

$$\sup_{\tilde{\lambda} \in \mathbf{\Lambda}_n(u_n)} \mathbb{P}_G(\tilde{\lambda} \notin \mathbf{R}(G)) \geq a_n, \quad \forall n \in \mathbb{N}. \tag{3.32}$$

Proof. We have that

$$
\begin{aligned}
\sup_{\tilde{\lambda}\in\mathbf{\Lambda}_n(u_n)} \mathbb{P}_G(\tilde{\lambda}\notin\mathbf{R}(G)) \geq & \int_{\tilde{\lambda}\in\mathbf{\Lambda}_n(u_n)} \mathbb{P}_G(\tilde{\lambda}\notin\mathbf{R}(G))d\mu_n(\tilde{\lambda}) \\
=& \mathbb{P}_{\lambda,G}(\lambda\notin\mathbf{R}(G)) \\
\geq & a_n. \qquad (3.33)
\end{aligned}
$$

□

In light of Lemma 3.14, we will find sequences, $\{\mu_n : n \in \mathbb{N}\}$, and $\{a_n : n \in \mathbb{N}\}$, with $\lim_{n\to\infty} a_n = 1$, so that Eq. (3.31) holds.

Fix $n \in \mathbb{N}$. We first construct the distribution μ_n. Let $\lambda' = \{\lambda'_1, \lambda'_2, \ldots, \lambda'_n\}$ be a random vector where all coordinates are independent, with

$$
\lambda'_i = \begin{cases} u_n, & \text{w.p. } \frac{\rho}{u_n(1+\epsilon)}, \\ 0, & \text{otherwise,} \end{cases} \qquad (3.34)
$$

for all i. Let the event H be defined by

$$
H = \left\{ \sum_{i=1}^{n} \lambda'_i \leq \rho n \right\}, \qquad (3.35)
$$

and by $\overline{H}$ its complement. Let λ be the random vector given by

$$
\lambda = \mathbf{I}(H)\lambda' + \mathbf{I}(\overline{H})\mathbf{0}, \qquad (3.36)
$$

where $\mathbf{0}$ is the $n\times1$ all-zero vector. That is, λ takes on the value of λ' if H occurs, and is set to zero, otherwise. It is not difficult to verify that by construction, $\lambda \in \mathbf{\Lambda}_n(u_n)$ almost surely. We will let μ_n be the distribution associated with λ.

We next show that

$$\lim_{n\to 1} \mathbb{P}_{\lambda,G}(\lambda \notin \mathbf{R}(G)) = 1, \tag{3.37}$$

which, by Lemma 3.14, will have proven our claim. Define the event

$$E_k = \left\{ \sum_{i\in A_k} \lambda'_i > (1+\epsilon)\rho d_n \right\}, \quad k \in \{1, \dots, n/d_n\}. \tag{3.38}$$

Note that, whenever $\epsilon > \frac{1}{\rho} - 1$, the occurrence of E_k for any k implies that λ' must not be in $\mathbb{R}(G)$. Therefore, we have that

$$\mathbb{P}(\lambda' \notin \mathbf{R}(G)) \geq \mathbb{P}\left(\bigcup_{k=1}^{n/d_n} E_k \right). \tag{3.39}$$

Let $\{X_i\}_{i\in\mathbb{N}}$ be i.i.d. Bernoulli random variables with $\mathbb{P}(X_1) = \frac{\rho}{u_n(1+\epsilon)}$. By the definition of λ' (cf. Eq. (3.34)), we have that

$$\begin{aligned}
\mathbb{P}(E_1) =& \mathbb{P}\left(\sum_{i\in A_1} \lambda'_i > (1+\epsilon)^2 \rho d_n \right) \\
=& \mathbb{P}\left(\sum_{i=1}^{d_n} X_i > (1+\epsilon)^2 \rho \frac{d_n}{u_n} \right) \\
=& \mathbb{P}\left(\frac{1}{d_n} \sum_{i=1}^{d_n} X_i > (1+\epsilon)^2 \mathbb{E}(X_1) \right).
\end{aligned} \tag{3.40}$$

By Sanov's theorem (cf. Chapter 12, [25]), we have that

$$\begin{aligned}
\mathbb{P}(E_1) =& \mathbb{P}\left(\frac{1}{d_n} \sum_{i=1}^{d_n} X_i > (1+\epsilon)\mathbb{E}(X_1) \right) \\
\gtrsim& \frac{1}{(d_n+1)^2} \exp\left(-D_B\left(\frac{(1+\epsilon)^2 \rho}{u_n} \middle\| \frac{\rho}{u_n} \right) d_n \right),
\end{aligned} \tag{3.41}$$

where $D_B(p\|q)$ is the Kullback-Leibler divergence between two independent Bernoulli distributions with parameters p and q, respectively, with

$$D_B(p\|q) = p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}. \tag{3.42}$$

Fixing $r \in (0,1)$, and using the fact that $\ln(1+y) \sim y$ as $y \to 0$, we have that

$$D_B\left(x\| rx\right) \sim h_r x, \quad \text{as } x \to 0. \tag{3.43}$$

where $h_r = 1 - r + \ln\frac{1}{r} > 0$. Recall that $d_n \geq c_1 \ln n$, and $u_n \geq c_3 d_n \ln^{-1} n$. By Eq. (3.43), with $x = (1+\epsilon)^2\rho/u_n$ and $r = 1/(1+\epsilon)^2$, we can set c_3 to be sufficiently large so that

$$D_B\left(\frac{(1+\epsilon)\rho}{u_n}\middle\| \frac{\rho}{u_n}\right) d_n \leq 2h\frac{d_n}{u_n}, \tag{3.44}$$

for all $n \geq 10$, where $h = h_{1/(1+\epsilon)^2} > 0$. Combining Eqs. (3.41) and (3.44), we have that

$$\mathbb{P}(E_1) \gtrsim \frac{1}{(d_n+1)^2}\exp\left(-2h\frac{d_n}{u_n}\right) \overset{(a)}{\gtrsim} n^{-2h/c_3} d_n^{-2}, \tag{3.45}$$

where step (a) follows from the assumption that $u_n \geq c_3 d_n \ln^{-1} n$ and $d_n \leq n$. Fix

$\epsilon > \frac{1}{\rho} - 1$ and $c_3 = 40h$. We have that

$$\begin{aligned}
\mathbb{P}(\lambda' \notin \mathbf{R}(G)) \geq & \mathbb{P}\left(\bigcup_{k=1}^{n/d_n} E_k\right) \\
& \overset{(a)}{=} 1 - \Pi_{k=1}^{n/d_n} (1 - \mathbb{P}(E_k)) \\
& \overset{(b)}{\geq} 1 - \left(1 - d_n^{-3} n^{1-2h/c_3} d_n/n\right)^{n/d_n} \\
& \overset{(c)}{\geq} 1 - \left(1 - n^{0.05} d_n/n\right)^{n/d_n} \\
& \to 1, \quad \text{as } n \to \infty,
\end{aligned} \tag{3.46}$$

where step (a) is based on the independence among the events in $\{E_k : k = 1, \ldots, n/d_n\}$, which is in turn based on the independence among the λ_i's, step (b) follows from Eq. (3.45), and step (c) from the assumption that $d_n \leq n^{0.3}$ and $c_3 = 40h$.

We next show that the event H occurs with high probability, as $n \to \infty$.

$$\begin{aligned}
\mathbb{P}(H) = & \mathbb{P}\left(\sum_{i=1}^{n} \lambda_i' \leq \rho n\right) \\
= & \mathbb{P}\left(\sum_{i=1}^{n} X_i \leq \rho n / u_n\right) \\
= & 1 - \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > (1+\epsilon)\mathbb{E}(X_1)\right) \\
\overset{(a)}{\geq} & 1 - \exp\left(-\frac{\epsilon^2 \rho}{3(1+\epsilon)} \cdot \frac{n}{u_n}\right) \to 1, \quad \text{as } n \to \infty,
\end{aligned} \tag{3.47}$$

where the X_is are i.i.d. Bernoulli random variables with $\mathbb{E}(X_1) = \frac{\rho}{u_n(1+\epsilon)}$, and step (a) follows from the Chernoff bound, and the fact that $u_n \ll n$.

We are now ready to prove Eq. (3.37). We have that

$$\begin{aligned}\mathbb{P}_{\lambda,G}(\lambda \notin \mathbf{R}(G)) =&\mathbb{P}_{\lambda',G}(\mathbf{I}(H)\lambda' + \mathbf{I}(\overline{H})\mathbf{0} \notin \mathbf{R}(G))\\ =&\mathbb{P}_{\lambda',G}\left(H \cap \{\lambda' \notin \mathbf{R}(G)\}\right)\\ \geq&\mathbb{P}(H) + \mathbb{P}\left(\lambda' \notin \mathbf{R}(G)\right) - 1\\ \overset{(a)}{\to}&1, \quad \text{as } n \to \infty, \end{aligned}\tag{3.48}$$

where step (a) follows from Eqs. (3.46) and (3.47). Together with Lemma 3.14, this completes the proof of the lower bound, Eq. (3.12). □

3.4.4 Proof of Theorem 3.9

Proof. Denote by $Q_i(t)$ the number of jobs in queue i at time t, and by $Q_q(t)$ the total number of jobs in queue cluster q, i.e.,

$$Q_q(t) = \sum_{i \in q} Q_i(t). \tag{3.49}$$

It is not difficult to verify that the evolution of $Q_q(\cdot)$ is identical to the number of jobs in an $M/M/k$ queue, with $k = d_n$ and arrival rate $\kappa_q = \sum_{i \in q} \lambda_i$. Note that since $\lambda \in \mathbf{R}(g_n)$, we have that $\kappa_q < d_n$. Using the equation for the expected waiting time in queue for an $M/M/k$ queue, one can show that the average waiting time across jobs arriving to cluster q, W_q, satisfies

$$\mathbb{E}\left(W_q | \gamma\lambda\right) = \frac{1}{\sum_{i \in q} \lambda_i} \sum_{i \in q} \lambda_i \mathbb{E}\left(W_i\right) = \frac{C(d_n, \kappa_q)}{d_n - \kappa_q} \leq \frac{C(d_n, \gamma d_n)}{(1-\gamma) d_n} \overset{(a)}{\lesssim} \exp(-c \cdot d_n), \tag{3.50}$$

where $C(x,y)$ is the Erlang's C formula: $C(x,y) = \frac{(xy)^x}{x!}\cdot\frac{1}{1-y}\left(\frac{(xy)^x}{x!}\cdot\frac{1}{1-y}+\sum_{i=0}^{x-1}\frac{(xy)^i}{i!}\right)^{-1}$, and step (a) follows from the fact that for all $\gamma \in (0,1)$, there exists $c > 0$, so that $C(x,\gamma x) \lesssim \exp(-cx)$ as $x \to \infty$.

□

3.4.5 Proof of Theorem 3.11

Proof. The proof consists of two parts. We first show that for all $\lambda \in \mathbf{\Lambda}_n(u_n)$, there always exists a feasible flow over the graph g_n^e between the queue and server clusters. We then illustrate how the scheduling policy based on such a flow leads to a diminishing delay, as in Eq. (3.16).

By the max-flow min-cut theorem, to check the feasibility of λ, it suffices to verify that

$$\sum_{q\in H}\left(\sum_{i\in q}\lambda_i\right) \le \frac{1+\rho}{2}d_1(n)|\mathcal{N}(H)|, \quad \forall H \subset \mathcal{Q}, \tag{3.51}$$

which is equivalent to having

$$\sum_{q\in H}\kappa_q \le |\mathcal{N}(H)|, \quad \forall H \subset \mathcal{Q}, \tag{3.52}$$

where $\kappa_q = \frac{2}{1+\rho}\cdot\frac{1}{d_1(n)}\sum_{i\in q}\lambda_i$. In other words, it suffices to show that

$$\left(\kappa_1,\dots,\kappa_{\frac{n}{d_1(n)}}\right) \in \mathbf{R}(g_n^e). \tag{3.53}$$

To this end, note that

$$\sum_{q\in Q}\kappa_q \le \frac{2}{1+\rho}\cdot\frac{1}{d_1(n)}\sum_{i\in I}\lambda_i \le \frac{2\rho}{1+\rho}\cdot\frac{n}{d_1(n)} \tag{3.54}$$

$$\max_{q\in Q}\kappa_q \le \frac{2}{1+\rho}\cdot\frac{1}{d_1(n)}\left(d_1(n)\max_{i\in I}\lambda_i\right) \le \frac{2}{1+\rho}u_n \overset{(a)}{\le} \beta_n, \tag{3.55}$$

where step (a) follows from the assumption that $u_n \le \frac{1+\rho}{2}\beta_n$. With Eqs. (3.54) and (3.55) at hand, the validity of Eq. (3.53) follows by applying Lemma 3.4 to the expander g_n^e, with the demand vector $\left(\kappa_1,\dots,\kappa_{\frac{n}{d_1(n)}}\right)$. This proves the existence of a feasible flow $\{f_{q,s}\}_{(q,s)\in E^e}$.

We are now ready to analyze the delay associated with the scheduling policy. Fix $q \in Q$. Recall the definition of $p_{s,q}$ in Eq. (3.14), the rate at which the queue cluster q gets chosen by the servers is

$$\begin{aligned}
\mu_q &= \sum_{s\in\mathcal{N}(q)} d_1(n)p_{s,q}\\
&= \sum_{s\in\mathcal{N}(q)} d_1(n)\left(\frac{f_{q,s}}{\sum_{q'\in\mathcal{N}(s)} f_{q',s}}\cdot\frac{1+\rho}{2}+\frac{1}{\deg(s)}\cdot\frac{1-\rho}{2}\right)\\
&= \left(\sum_{s\in\mathcal{N}(q)}\frac{f_{q,s}}{\sum_{q'\in\mathcal{N}(s)} f_{q',s}}\right)\frac{1+\rho}{2}d_1(n)+\left(d_1(n)\sum_{s\in\mathcal{N}(q)}\frac{1}{\deg(s)}\cdot\frac{1-\rho}{2}\right)\\
&\overset{(a)}{\ge} \left(\sum_{i\in q}\lambda_i\right)+\left(\frac{1-\rho}{2}\cdot\frac{\deg(q)}{\deg(s)}\right)d_1(n)\\
&\overset{(b)}{\ge} \left(\sum_{i\in q}\lambda_i\right)+\left(\frac{1-\rho}{2}\cdot\frac{\beta_n}{d_2(n)}\right)d_1(n)\\
&= \left(\sum_{i\in q}\lambda_i\right)+\frac{1-\rho}{4}\left(\ln^{-1}\frac{1}{\rho}+1\right)^{-1}d_1(n),
\end{aligned} \tag{3.56}$$

where step (a) follows from the feasibility of the flow $\{f_{q,s}\}_{(q,s)\in E^e}$, and step (b) from the fact that g_n^e is an (α_n,β_n)-expander, and hence $\deg(q)\ge\beta_n$, and the fact that g_n^e

has maximum degree $d_2(n)$, and hence $\deg(s) \leq d_2(n)$.

Denote by $Q_i(t)$ the number of jobs in queue i at time t, and by $Q_q(t)$ the total number of jobs in cluster q, i.e.,

$$Q_q(t) = \sum_{i \in q} Q_i(t). \tag{3.57}$$

It is not difficult to verify that, under the scheduling policy considered, the evolution of $Q_q(\cdot)$ is identical to that of the *number of jobs in system* in an initially empty $M/M/1$ queue with arrival rate $\kappa_q = \sum_{i \in q} \lambda_i$, and service rate μ_q.[12] Using Eq. (3.56) and the equation for the expected time in system in an $M/M/1$ queue, we have that the average waiting time across all queues in cluster q, W_q, satisfies

$$\mathbb{E}\left(W_q|\lambda\right) = \frac{1}{\sum_{i \in q} \lambda_i} \sum_{i \in q} \lambda_i \mathbb{E}\left(W_i\right) = \frac{1}{\mu_q - \lambda_q} \leq \frac{c}{d_1(n)}, \tag{3.58}$$

where $c = \frac{4}{1-\rho}(\ln^{-1} \frac{1}{\rho} + 1)$. Since Eq. (3.58) holds for all $q \in \mathcal{Q}$ and $\lambda \in \mathbf{\Lambda}_n(u_n)$, we have completed the proof of Theorem 3.11. □

3.5 Summary and Future Research

The main message of this chapter is that a large capacity region and diminishing delay can be jointly achieved in a system where the level of processing flexibility of each server is small compared to the system size. We proposed several flexibility

[12] Note that in the proof of Theorem 3.9, the total queue length process in a cluster evolves as the queue length of an $M/M/k$ queue, with $k = d_n$, whereas in this proof, Q_q is compared to the total number of jobs in system for an $M/M/1$ queue. This is because in a Modular architecture, an arriving job immediately initiates service if there is server in the corresponding subnetwork that is currently available. In contrast, in the Expanded Modular architecture, incoming jobs always wait in the queue until they are fetched by a server.

architectures, along with associated scheduling policies that achieve these objectives to various degrees. At a high-level, the key features of the different architectures are summarized as follows (see Table 3.1 for a more detailed comparison).

1. The Random Graph architecture provides, with high probability, a capacity region that is essentially optimal, and diminishing delays for most arrival rate vectors therein. It remains an open problem whether diminishing delays can be achieved for *all* arrival rates in the capacity region.

2. With proper randomization, a Modular architecture is able to provide small delays for "many" arrival rates, by means of a simple greedy scheduling policy. However, for any given Modular architecture, there are always many inadmissible arrival rate vectors, even if the maximum arrival rate across the queues is of constant order.

3. The Expanded Modular architecture is capable of providing both a large capacity region, and diminishing delays for *all* arrival rate vectors therein. However, such robustness comes at a cost, as the the designer has to make a trade-off between the size of the capacity region and the speed at which delay diminishes. Furthermore, our scheduling policy relies on the knowledge of the arrival rate vector, λ.

Based on our results, the Random Graph architecture appears to have the best performance. It remains an open problem, however, whether even better performance can be achieved. In particular, can one find an "ideal" flexibility architecture that guarantees a large capacity region with $u_n = \Omega(d_n)$, and a diminishing delay for *all* arrival rate vectors therein? This is formalized in the following conjecture.

Conjecture 3.15 (Existence of "Ideal" Architectures). *Suppose that $d_n \gg 1$. There exists a constant $h > 0$, such that if*

$$u_n/d_n \leq h, \quad \textit{for all } n \geq 1, \tag{3.59}$$

then there exists a sequence of architectures, $\{g_n\}_{n\geq 1}$, and associated scheduling policies, under which

$$\mathbb{E}\left(W|\lambda\right) \leq c_1 \exp(-c_2 \cdot d_n), \quad \textit{for all } n \geq 1 \textit{ and } \lambda \in \Lambda_n(u_n), \tag{3.60}$$

where c_1 and c_2 are positive constants independent of n or λ.

A weaker conjecture, more in line with the delay scaling we proved for the Random Graph architecture (Theorem 3.6), would only require that $\mathbb{E}(W|\lambda) \leq c_1/d_n$, instead of the exponential dependence on d_n in Eq. (3.60).

The scaling regime considered in this chapter assumes that the traffic intensity is fixed as n increases, which fails to capture system performance in the heavy-traffic regime ($\rho \approx 1$). It would be interesting to consider a scaling regime in which ρ and n scale simultaneously (e.g., as in the celebrated Halfin-Whitt regime [40]), but it is unclear at this stage what exact formulations and analytical techniques are appropriate.

Chapter 4

The Random Graph Architecture

4.1 Virtual Queue and the Scheduling Policy

In this chapter, we provide a detailed construction of the virtual-queue-based scheduling policy used in the Random Graph architecture, which will then be used to prove Theorem 3.6 of Chapter 3. We begin by describing some high-level ideas behind our design.

Regularity vs. Discrepancies Setting aside computational issues, an efficient scheduling policy is difficult to design because future inputs are unpredictable and random: one does not know *a priori* which part of the network will become more loaded, and hence current resource allocation decisions must take into account all possibilities for future arrivals and job sizes, which is difficult to carry out or analyze.

However, as the size of the system, n, becomes large, certain *regularities* in the arrival processes begin to emerge. To see this, consider the case where $\lambda_{n,i} = \lambda < 1$ for all n and i, and assume that at time $t > 0$, all servers are busy serving some job. Now, during time interval $[t, t + \gamma_n)$, "roughly" $\lambda n \gamma_n$ new jobs will arrive, and

$n\gamma_n$ servers will become available. For this $[t, t+\gamma_n)$ interval, denote by Γ the set of queues that received any job, and by Δ the set of $n\gamma_n$ servers who became available. If $\lambda n\gamma_n \ll n$, these incoming jobs are likely to spread out across the queues, so that most queues receive at most one job. Assuming that this is indeed the case, we see that the connectivity graph g_n restricted to $\Gamma \cup \Delta$, $g_n|_{\Gamma\cup\Delta}$, is a subgraph sampled uniformly at random among all $(\lambda n\gamma_n \times n\gamma_n)$-sized subgraphs of g_n. When $n\gamma_n$ is sufficiently large, and g_n is *well connected* (as in an Erdös-Rényi random graph with appropriate edge probability), we may expect that, with high probability, $g_n|_{\Gamma\cup\Delta}$ admits a matching (Definition 4.2) that includes the entire Γ, in which case *all* $\lambda n\gamma_n$ jobs can start receiving service by the end of the interval.

Note that when n is sufficiently large, despite the randomness in the arrivals, the symmetry in the system makes delay performance at a short time scale *insensitive* to the exact locations of the arrivals. In other words, treated collectively, the structure of the set of arrivals and available servers in a small interval becomes less random and more "regular," as $n \to \infty$. Of course, for any finite n, the presence of randomness means that *discrepancies* (events that deviate from the expected regularity) do not completely disappear. For instance, the following two types of events will occur with small, but nonzero, probability.

1. Arrivals may be located in a poorly connected subset of g_n.

2. Arrivals may concentrate on a small number of queues.

One will need to take care of these outliers, and hope that any of their negative impacts on performance are insignificant.

Following this line of thought, our scheduling policy aims to use most of the resources to dynamically target the *regular* portion of the traffic (by assigning jobs to servers in batches), while ensuring that the impact of the *discrepancies* is well contained. In particular, we will use a two-mode *virtual queue* to serve these two objectives:

1. A *"collect" mode*, which targets *regularity* in arrival and service times.
2. A *"clear" mode*, which is invoked once discrepancies occur.

The queue is "virtual," as opposed to "physical," in the sense that it merely serves to conceptually simplify the description of the scheduling policy.

Good Graph Note that the operations with the virtual queue have to fully comply with the underlying connectivity graph, g_n, which is fixed over time. We informally describe here what key structural properties a "good" g_n should possess, while the more detailed definitions and performance implications will be addressed in subsequent sections, as a part of the queueing analysis. In particular, the set $\mathcal{H}_n$ (as in Theorem 3.6), which consists of good graphs, is the intersection of the following subsets of $\mathcal{G}_n$:

1. $\hat{\mathcal{G}}_n$ (Lemma 4.3): g_n admits a full matching. This property will be used in the virtual queue to handle *discrepancies.*
2. $\tilde{\mathcal{G}}_n$ (Lemma 4.11): with high probability, g_n admits a large set of "flows" over a randomly sampled sublinear-sized subgraph. This property will be used in the virtual queue to take advantage of *regularity.*
3. $\mathcal{L}_n$ (Section 4.5) : g_n has an average degree that is of the order d_n. This property is to comply with our degree constraint.

Input to the Scheduling Policy Besides n, the scheduling policy uses the following inputs:

1. ρ, the traffic intensity as defined in Condition 3.1 in Section 3.2.2,
2. ϵ, a constant in $(0, 1 - \rho)$,
3. b_n, a batch size function,
4. g_n, the interconnection topology.

Notice that the fluctuation parameter, u_n, is *not* an input to the scheduling policy.

4.2 Arrivals to the Virtual Queue

The arrivals to the virtual queue are arranged in *batches.* Roughly speaking, a batch is a set of jobs that are treated collectively as a single entity. We define a sequence of random times $\{T_B(k)\}_{k \in \mathbb{Z}_+}$, by letting $T_B(0) = 0$, and for all $k \geq 1$,

$$T_B(k) = \text{time of the } (k\rho b_n)\text{th arrival to the system},$$

where $b_n \in \mathbb{Z}_+$ is a design parameter that corresponds to the size of the batch, and will be referred to as the *batch parameter.* We will refer to the time period $(T_B(k-1), T_B(k)]$ as the k**th batch period**, which also corresponds to the interarrival times to the virtual queue, defined as follows.

Definition 4.1. (Arrival Times to the Virtual Queue) *The time of arrival of the kth batch to the virtual queue is $T_B(k)$, and the corresponding interarrival time is $A(k) \triangleq T_B(k+1) - T_B(k)$.*

Finally, we will use the $n \times 1$ vector, $M(k)$, to represent the content of the batch, i.e., the ρb_n jobs that arrive during the kth batch period. In particular,

$$M_i(k) = \# \text{ of jobs arriving to queue } i \text{ during the } k\text{th batch period.} \tag{4.1}$$

4.3 Mode Transitions and Service Rules

This section describes the actions of the physical servers. Before getting into the details, we first describe the general ideas. For each batch of arrivals, we will first "collect" a number of available servers, which is approximately equal to the size of the batch:

1. With high probability, *all jobs* in the batch can be simultaneously assigned to a unique server through g_n.

2. With small probability, some jobs in the batch are located in a poorly connected subset of g_n, so that they cannot be assigned to the available servers. In this case, all jobs in the batch will be served one by one according to a *fixed* server-to-queue mapping (a "clear" phase).

To implement the above queueing dynamics, we will specify the evolution of *modes* and *actions* of the *virtual queue* as well as the *physical servers*, which will be described in detail in the remainder of this section. Examples of some of the mode transitions are illustrated in Figure 4-1.

4.3.1 Modes and Actions of the Virtual Queue

For the purpose of this subsection, we shall assume that each of the physical servers is in one of two modes: STANDBY and BUSY; the mode evolution for the physical servers will be described in the next subsection.

Mode transitions and scheduling actions for the virtual queue take place at discrete times, which we will refer to as the **service epochs**: let $T_S(0) = 0$, and for all

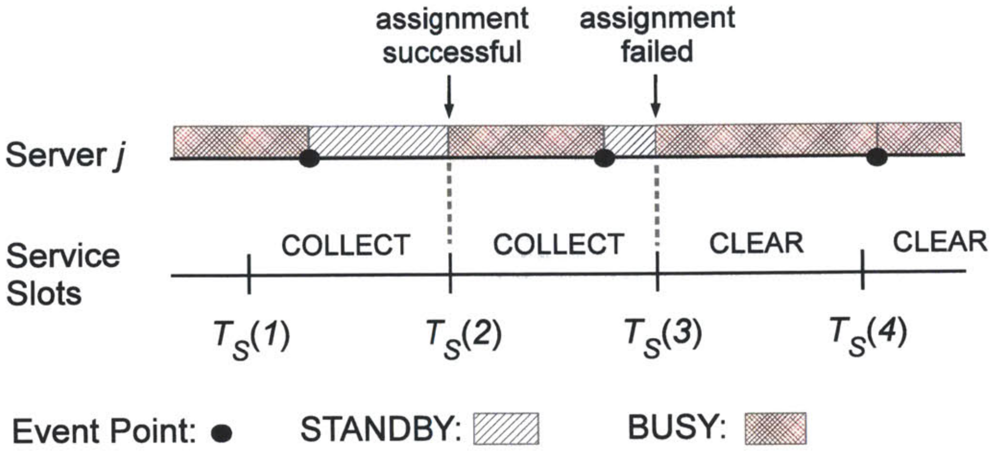

Figure 4-1: Examples of mode transitions at the physical server and service slots.

$k \geq 1$,

$$T_S(k) = k\frac{b_n}{n}(\rho + \epsilon), \tag{4.2}$$

where ϵ is a constant in $(0, 1 - \rho)$. We refer to the interval $(T_S(k-1), T_S(k)]$ as the kth **service slot**. To see how the length of the service slot was chosen, recall that the size of each batch is equal to ρb_n. The length of the service slot hence ensures that the expected number of servers that will become available during a single service slot is on the same order of, and strictly great than, the size of a batch.

In order to coordinate the actions of various physical servers, we will associate with each service slot one of the two modes: COLLECT and CLEAR. Accordingly, we will say that a service slot can be a COLLECT slot or a CLEAR slot. The first service slot is initialized to mode COLLECT. For all $k \geq 2$, the following takes place at the beginning of the kth service slot (cf. Figure 4-1).

1. Suppose the $(k-1)$th service slot is in mode COLLECT. Let $\Delta \subset J$ be the set of servers currently in STANDBY. Suppose there is at least one batch in the virtual queue, and let M' be the batch currently at the front of the queue.

(a) If there exists a assignment, F, from the set of all jobs in M' to the set of STANDBY servers, Δ, so that each job is assigned to a *unique* STANDBY server that is connected to its arriving queue, let all servers in Δ to which a job is assigned enter mode BUSY, and initiate the processing of the assigned job. If there remain some servers in Δ without a job (which will occur if $|M'| < |\Delta|$), let all these servers enter mode BUSY by initiating a vacation, with a length independently distributed according to Expo (1). This marks the departure of a batch from the virtual queue. We let the kth service slot remain in mode COLLECT.

(b) If no such assignment exists, we let the kth service slot be in mode CLEAR. All servers in mode STANDBY enter mode BUSY by initiating a vacation, with a length independently distributed according to Expo (1).

If there is no batch waiting in the virtual queue, let all servers in Δ enter mode BUSY by initiating a vacation, with a length independently distributed according to Expo (1). Let the kth service slot remain in mode COLLECT.

2. Suppose the $(k-1)$th service slot is in mode CLEAR (actions of physical servers during a CLEAR service slot will be described in Section 4.3.2).

   (a) If all jobs in the current batch have started receiving processing from one of the servers by the end of the $(k-1)$th service slot, we let the kth service slot be in mode COLLECT. This marks the departure of a batch from the virtual queue.

   (b) Otherwise, we let the kth service slot remain in mode CLEAR.

4.3.2 States and Actions of Physical Servers

We now describe the actions and mode evolution of the physical servers. We first introduce the notion of full matching over a bipartite graph, which will be used in the description of the scheduling rules for the physical servers.

Definition 4.2 (Full Matching in a Bipartite Graph). *Let $g = (E, I \cup J)$ be a bipartite graph, where $|I| = |J|$. We say that $L : I \to J$ is a full matching of g, if L is a bijection from I to J, and $(i, L(i)) \in E$ for all $i \in I$.*

As will become clear in the sequel, our scheduling policy will use a full matching L to ensure that every queue will receive at least a "minimum service rate" from some server. The following lemma modes that, with high probability, an Erdös-Rényi random bipartite graph with a sufficiently high edge probability admits a full matching. The proof consists of a simple argument using Hall's marriage theorem and a union bound (c.f. Lemma 2.1 in [14]).

Lemma 4.3. *Let $p(n) = d_n/n$. If $d_n \gg \ln n$, then there exists a sequence of sets $\{\hat{\mathcal{G}}_n\}_{n\geq 0}$, $\hat{\mathcal{G}}_n \subset \mathcal{G}_n$, such that $\lim_{n\to\infty} \mathbb{P}_{n,p(n)}(\hat{\mathcal{G}}_n) = 1$, g admits a full matching L for all $g \in \mathcal{G}_n$, $n \geq 0$.*

For the remainder of the subsection, we shall assume that the underlying connectivity graph, g, admits a full matching, L.

A physical server can be, at any time, in one of two modes: BUSY and STANDBY. The *end* of a BUSY mode will be referred to as an *event point*, and the time interval between two adjacent event points an *event period.* At each event point, a new decision is to be made as to which job the server will choose to serve during the next event period, or whether the server should serve any job at all. All servers will be initialized in a BUSY mode, with the time till the first event point distributed as Expo (1), independently across all servers.

Fix a full matching, L, in the connectivity graph, g. Suppose that at time t server $j \in J$ is at an event point, and let k^* be the index of the service slot (defined in Section 4.3.1) to which t belongs. Server j makes the following decisions (cf. Figure 4-1).

1. If the k^*th service slot is in mode COLLECT, server j enters mode STANDBY.

2. Otherwise (the k^*th service slot is in mode CLEAR), let M' be the batch at the front of the virtual queue, and $B' \subset I$ be the set of queues that still contain an unserved job from batch M'. Let $i^* = \min\{i : i \in B'\}$.

   (a) If $L(i^*) = j$, then server j starts processing a job in queue i^* that belongs to M', entering mode BUSY.

   (b) Otherwise, server j goes on a vacation of length Expo (1), entering mode BUSY.

   Note that the physical servers' actions during a CLEAR service slot are designed to serve all jobs in B' in a sequential fashion.

The above procedure describes all the mode transitions for a single server, except for one case: when in mode STANDBY, a server can be ordered by the virtual queue to start processing a job or initiate a vacation period. The rules that govern such transitions out of the STANDBY mode have been described as a part of the actions of the virtual queue in Section 4.3.1.

By now, we have described how the batches are formed (Section 4.2), and the actions of the virtual queue and physical servers (Sections 4.3.1 and 4.3.2). The scheduling policy is hence fully specified.

4.4 Dynamics of the Virtual Queue

In this section, we analyze the dynamics of the virtual queue, as well as the resulting delay experienced by the jobs, defined as follows:

Definition 4.4 (Delay in the Virtual Queue). *The delay for a batch in the virtual queue is the time elapsed from the batch's arrival to the virtual queue till the time when all jobs in the batch start receiving service from a physical server.*

The main idea behind the delay analysis is rather simple: we will treat the virtual queue as a $GI/GI/1$ queue, and use Kingman's bound [51] to derive an upper bound on the expected waiting time in queue. The combination of a batching policy with Kingman's bound is a fairly standard technique in queueing theory for deriving delay upper bounds (see, e.g., [83]). Our main effort will go into characterizing the various queueing primitives associated with the virtual queue (arrival rates, traffic intensity, and variances of inter-arrival and service times).

Starting with this section, we will focus on a specific batch size function of the form

$$b_n = K_n \frac{n \ln n}{d_n} = K_n \frac{n}{y_n}, \tag{4.3}$$

where

$$K_n = \max\left\{ \left(1 + \frac{\rho}{\epsilon}\right)^2 d_n/n \, , \, \frac{96}{1-\rho} \ln n \right\}, \tag{4.4}$$

and y_n is defined by

$$y_n = \frac{d_n}{\ln n}. \tag{4.5}$$

We shall also assume that $d_n \ll n$, and

$$d_n \geq \frac{4}{1-\rho} \ln^{2.1} n. \tag{4.6}$$

It is not difficult to verify, under these choices of d_n and K_n, that

$$b_n = K_n \frac{n \ln n}{d_n} \ll n, \tag{4.7}$$

that is, the batch size is *vanishingly small* compared to n. Finally, we assume that the arrival rate vector always belongs to the set $\Lambda_n(u_n)$ (Condition 3.1), and that

$$u_n \leq \frac{1-\rho}{8} d_n / \ln n. \tag{4.8}$$

4.4.1 Inter-arrival Time Statistics

We begin with a characterization of the inter-arrival-time distribution for the virtual queue.

Lemma 4.5. *The inter-arrival times of batches to the virtual queue, $\{A(k)\}_{k \geq 1}$, are i.i.d., with $\mathbb{E}(A(k)) = b_n/n$, and $\mathrm{Var}(A(k)) \lesssim b_n/n^2$.*

Proof. By definition, $A(k)$ is equal in distribution to the time until a Poisson process with rate ρn records ρb_n arrivals. Therefore, the $A(k)$'s are Erlang random variables (sum of ρb_n exponentials), with $\mathbb{E}(A(k)) = (\rho b_n)/(\rho n) = b_n/n$, and $\mathrm{Var}(A(k)) = \rho b_n \cdot \frac{1}{(\rho n)^2} \lesssim b_n/n^2$. □

4.4.2 A Breakdown of Service Times

We now turn our attention to the service times in the virtual queue, defined as follows.

Definition 4.6. (Service Times for Virtual Queue) *Consider the kth batch arriving at the virtual queue. Define the time of service initiation, E_k, to be beginning* *of the service slot during which the batch first reaches the front of the queue, and the time of departure, D_k, to be the end of the service slot during which the last job in the batch starts receiving service from one of the physical servers. The service time for the kth batch is defined to be*

$$S^M(k) = D_k - E_k.$$

The interval $[E_k, D_k)$ is referred to as the service period of the kth batch.

Note that the definition of service times in the virtual queue takes the actual time interval for which a batch stays at the front of the virtual queue, and rounds it up to the smallest set of *service slots* that contain it. The advantage of such a definition is that now the service times, $S^M(k)$, are i.i.d. Moreover, since this rounding procedure does not decrease the time a batch spends at the front of the virtual queue, one can show the resulting queueing waiting time serves as an upper bound for the actual waiting time of the batch. This is formalized in the following lemma, whose proof involves a simple coupling argument based on Lindley's recursion, which we omit.

Lemma 4.7. *Denote by $W_M(k)$ the waiting time for the kth batch in the virtual queue. Let $\tilde{W}(k)$ be the waiting time of the kth job arriving to a $GI/GI/1$ queue with arrival times $\{A(k)\}_{k\geq 1}$ and service times $\{S^M(k)\}_{k\geq 1}$. We have that*

$$\tilde{W}(k) \geq W_M(k), \quad \forall k \geq 1, \text{ almost surely.} \tag{4.9}$$

Based on our construction, the value of $S^M(k)$ is at least the length of one service slot in mode COLLECT. If a job-to-server assignment fails to exist by the end of the COLLECT slot, $S^M(k)$ will also include subsequent service slots in mode CLEAR, until all jobs in the batch have started to receive service from a physical server. We will therefore write

$$S^M(k) \overset{d}{=} S_{col} + X \cdot S_{cle}, \tag{4.10}$$

where S_{col} and S_{cle} correspond to the lengths of the service slots in mode COLLECT and CLEAR, respectively, and X is a Bernoulli random variable indicating the non-existence of a job-to-server assignment, with $\mathbb{P}(X = 1) = q(g_n)$, where $q(g_n)$ is defined as follows. Denoting by M' the batch at the front of the virtual queue, and Δ the set of STANDBY servers, at the end of the COLLECT slot, we let

$$q(g_n) = \mathbb{P}\{M' \text{ cannot be assigned to } \Delta \text{ over } g_n\}, \tag{4.11}$$

where the probability is taken over the randomness in M' and Δ, but is conditional on the underlying connectivity, g_n.

We now examine each quantity on the right-hand side of Eq. (4.10). The value of S_{col} is the simplest, as it is equal to the length of one service slot, and we have

$$S_{col} = T_S(1) = \frac{b_n}{n}(\rho + \epsilon). \tag{4.12}$$

We next look at S_{cle}, the total length of the subsequent service slots in mode CLEAR before the batch departs from the virtual queue. We shall define **CLEAR period** as the collection of successive CLEAR service slots associated with a batch. Recall from Section 4.3.2 that, during the CLEAR period, the time until the next job in the current batch starts to receive service from a physical server is exponentially distributed with mean 1. Because there are at most ρb_n jobs in a batch, conditional on $X = 1$, the length of a CLEAR period, S_{cle}, is no greater than the amount of time it takes for a Poisson process of rate 1 to record ρb_n arrivals, rounded to the end of the last service slot. Arguing similar to the proof of Lemma 4.5, we have the following lemma. Note that the distribution of S_{cle} does not depend on the structure of g_n, as long as g_n admits a full matching (cf. Lemma 4.3).

Lemma 4.8. $\mathbb{E}\left(S_{cle} \,|\, X = 1\right) \lesssim b_n$, *and* $\mathbb{E}\left(S_{cle}^2 \,|\, X = 1\right) \lesssim b_n^2$.

4.4.3 Probability of Assignment Success

We now examine the value of $1 - q(g_n)$, i.e., the probability that all jobs in a batch *can be* assigned to one of the STANDBY servers at the end of a COLLECT service slot. Recall the definition of flows in Definition 3.2. As a first step, for the convenience of notation and analysis, we will represent this job-to-server assignment as a binary *flow* over g_n, $\{f_e\}_{e \in E}$, where f_{ij} is equal to 1 if a job from queue i is assigned to server j, and 0, otherwise. Under this representation, assigning all jobs in the batch corresponds to finding a binary flow over g_n, which satisfies the "demand" induced by the batch.

Note that the feasibility condition in Definition 3.2 does not require the flow $\{f_e\}$ to be *binary* (i.e., $f_e \in \{0,1\}$ for all $e \in E$), a feature necessary for our purpose, since a job cannot be "split" by different servers. Fortunately, since the demand vector induced by a batch is always integral, it is well known that feasibility implies the existence of a feasible binary flow.

The following lemma is the main technical result of this section and will be used in the next subsection to bound the value of $q(g_n)$. It demonstrates that, with high probability, a random graph with degree d_n is able to accommodate a given flow whose maximum coordinate is approximately $d_n / \ln n$. The proof is given in Appendix A.1.4.

Lemma 4.9. *Fix* $\rho \in (0,1)$. *Let* $d_n \geq \frac{4}{1-\rho} \ln n$ *and* $u_n < \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n}$, *and fix some*

$\lambda \in \mathbf{\Lambda}_n(u_n)$. *Then, there exists $n' > 0$, independent of the choice of λ, so that*

$$\mathbb{P}_{n,d_n/n}\left(\{g_n : \lambda \in \mathbf{R}(g_n)\}\right) \geq 1 - \exp\left(-\frac{1-\rho}{4}d_n\right), \tag{4.13}$$

for all $n \geq n'$.

Flows on subgraphs We will now apply Lemma 4.9 to the randomly sampled subgraph of g_n, induced by the support of the batch and the set of STANDBY servers at the end of a COLLECT service slot. By doing so, we will establish the existence of a set of graphs, $\tilde{\mathcal{G}}_n \subset \mathcal{G}_n$, with the following two properties (Lemma 4.11):

1. The value of $q(g_n)$ is small, for every $g_n \in \tilde{\mathcal{G}}_n$. This property will help us upper bound the service time $S^M(k)$, using Eq. (4.10) and the moment bounds for S_{cle} developed earlier.
2. The set $\tilde{\mathcal{G}}_n$ has high probability under the Erdös-Rényi random graph model.

We start with some definitions. Fix $m \leq n$ and $\rho' < 1$. Let $\Xi^m(\rho')$ be the set of all demand vectors $\lambda^m = (\lambda_1^m, \ldots, \lambda_m^m)$ such that

$$\sum_{1 \leq i \leq m} \lambda_i^m \leq \rho' m, \tag{4.14}$$

$$\max_{1 \leq i \leq m} \lambda_i^m \leq \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{m}{n}. \tag{4.15}$$

The definition of Ξ is analogous to that of the capacity region, $\mathbf{\Lambda}$, but is intended to be used for a subgraph. Let $\mathcal{M}_{m,n}$ be the family of all $m \times m$ subsets of $I \cup J$, that is, $h \in \mathcal{M}_{m,n}$ if and only if $|h \cap I| = |h \cap J| = m$. Let $\psi_{m,n}$ be a probability measure on the product space $\Xi^m(\rho') \times \mathcal{M}_{m,n}$, and let $m(n)$ be such that $m(n) \to \infty$ as $n \to \infty$.

Define

$$l(g, \psi_{m(n),n}) = \psi_{m(n),n}\left(\left\{\left(\lambda^{m(n)}, h\right) \in \Xi^m \times \mathcal{M}_{m(n),n} : \lambda^{m(n)} \notin \mathbf{R}\left(g|_h\right)\right\}\right), \qquad (4.16)$$

where $\mathbf{R}(g|_h)$ is the set of feasible demand vectors for the subgraph $g|_h$ (Definition 3.2). We now define $\tilde{\mathcal{G}}_n\left(\psi_{m(n),n}\right)$ as the set of graphs in $\mathcal{G}_n$ for which the value of l is small. In particular,

$$\tilde{\mathcal{G}}_n\left(\psi_{m(n),n}\right) = \left\{g \in \mathcal{G}_n : l(g, \psi_{m(n),n}) \le n^{-3}\right\}. \qquad (4.17)$$

Informally, this is a set of graphs which, for the given measure $\psi_{m(n),n}$ on subgraphs and demand vectors, have a high probability that a random subgraph $g|_h$ will be able to admit the random demand vector $\lambda^{m(n)}$. Consistent with the general outline of the proof given in Section 4.1, we will show (a) that random graphs are highly likely to belong to $\tilde{\mathcal{G}}_n$ (Lemma 4.10) and (b) that graphs in $\tilde{\mathcal{G}}_n$ have favorable delay guarantees (Proposition 4.14).

Lemma 4.10. (Flow Feasibility on Random Subgraphs) *Suppose that* $d_n \ge \frac{4}{1-\rho}\ln n$*, and* $m(n) \in \left[\frac{16}{1-\rho'} \cdot \frac{n \ln n}{d_n}, n\right]$*. With* $p(n) = d_n/n$*, we have that, for any sequence* $\{\psi_{m(n),n} : n \ge 1\}$*,*

$$\lim_{n\to\infty} \mathbb{P}_{n,p(n)}\left(\tilde{\mathcal{G}}_n\left(\psi_{m(n),n}\right)\right) = 1. \qquad (4.18)$$

Remark on Lemma 4.10: Eq. (4.18) states that, with high probability, the Erdös-Rényi construction yields graphs in $\tilde{\mathcal{G}}_n$. This probabilistic statement is not to be confused with Eq. (4.17), which involves a *deterministic* property that holds for any $g \in \tilde{\mathcal{G}}_n$. In Eq. (4.17), the randomness lies only in the sampling of demand vectors and subgraphs (via $\psi_{m(n),n}$). This distinction is important for our analysis, because the interconnection topology, g_n, stays fixed over time, while the random subgraph,

$g_n|_{\Gamma\cup\Delta}$, is drawn independently for different batches.

Proof. **(Lemma 4.10)** Let $(\hat{\lambda}, H)$ be a random element of $\Xi^m(\rho') \times \mathcal{M}_{m,n}$, distributed according to $\psi_{m(n),n}(\cdot)$. Let G be an $(n, p(n))$ random bipartite graph over $I \cup J$, generated independently of H. Note that the distribution of G restricted to any $m(n)$-by-$m(n)$ subset of $I \cup J$ is that of an $(m(n), p(n))$ random bipartite graph.

We now invoke Lemma 4.9 on this random subgraph, which is of size $m(n)$, with average degree $d'_n = d_n\frac{m(n)}{n}$, and with an upper bound on the demand fluctuation $u'_n = \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{m(n)}{n}$. To verify that the ranges of values for d'_n and u'_n with respect to $m(n)$ satisfy the conditions in Lemma 4.9, note that

$$u'_n = \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{m(n)}{n} = \frac{1-\rho}{8} \cdot \frac{d'_n}{\ln n}. \tag{4.19}$$

and

$$d'_n = d_n \frac{m(n)}{n} \overset{(a)}{\geq} d_n \frac{4}{1-\rho'} \cdot \frac{\ln n}{d_n} \geq \frac{4}{1-\rho'} \ln n \overset{(b)}{\geq} \frac{4}{1-\rho'} \ln m(n), \tag{4.20}$$

where step (a) is based on the assumption that $m(n) \geq \frac{16}{1-\rho'} \cdot \frac{n \ln n}{d_n}$, and (b) on the fact that $m(n) \leq n$. Furthermore, the definition of $\Xi^m(\rho')$ (Eq. (4.14)) ensures that the total demand in $\hat{\lambda}$ is at most $\rho' m(n)$. We thus have

$$\begin{aligned}
\mathbb{E}\left(l(G, \psi_{m(n),n})\right) =& \mathbb{P}\left(\hat{\lambda} \notin \mathbf{R}(G|_H)\right) \\
\overset{(a)}{\leq}& \exp\left(-\frac{1-\rho'}{4} d'_n\right) \\
=& \exp\left(-\frac{1-\rho'}{4} d_n \frac{m(n)}{n}\right) \\
\overset{(b)}{\leq}& n^{-4},
\end{aligned} \tag{4.21}$$

where step (a) follows from Lemma 4.9 combined with Eqs. (4.19) and (4.20), and step (b) from the assumption that $m(n) \geq \frac{16}{1-\rho'} \cdot \frac{n \ln n}{d_n}$. Eq. (4.21) and Markov's inequality yield

$$\begin{aligned}\mathbb{P}_{n,p(n)}\big(\tilde{\mathcal{G}}_n\big) =& 1 - \mathbb{P}\left(l(G,\psi_{m(n),n}) > n^{-3}\right)\\ \geq& 1 - \frac{\mathbb{E}\left(l(G,\psi_{m(n),n})\right)}{n^{-3}}\\ \geq& 1 - \frac{1}{n},\end{aligned} \tag{4.22}$$

which converges to 1 as $n \to \infty$. □

Using Lemma 4.10, we can now obtain an upper bound on the value of $q(g_n)$.

Lemma 4.11. *Fix some $\lambda_n \in \Lambda_n(u_n)$. Let $p(n) = d_n/n$, and $b_n = K_n n/y_n$ (cf., Eq. (4.3)). There exists $\tilde{\mathcal{G}}_n \subset \mathcal{G}_n$, with*

$$\lim_{n\to\infty} \mathbb{P}_{n,p(n)}\left(\tilde{\mathcal{G}}_n\right) = 1, \tag{4.23}$$

such that if $g_n \in \tilde{\mathcal{G}}_n$, then

$$q(g_n) \lesssim n^{-3}.$$

Proof. Let $m(n) = \left(\rho + \frac{1}{2}\epsilon\right) b_n$. Denote by Γ the support of the batch, and by Δ the set of available servers who are in STANDBY at the end of the COLLECT service slot. Let $\tilde{\lambda}$ be the $|\Gamma| \times 1$ demand vector induced by the non-zero coordinates of batch $M(k)$,

$$\tilde{\lambda}_s = M_{i(s)}(k), \quad 1 \leq s \leq |\Gamma|, \tag{4.24}$$

where $i(s)$ is the sth non-zero coordinate of $M(k)$. Define the following events.

(i) $\mathcal{B}$: the event that $\max_{1\leq s\leq|\Gamma|} \tilde{\lambda}_s \leq \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n}$. This is the event where the variation among the coordinates of the demand vector $\tilde{\lambda}$ is not too large.

(ii) $\mathcal{C}$: the event that $|\Delta| \geq \left(\rho + \frac{1}{2}\epsilon\right) b_n \geq \frac{\rho+\frac{1}{2}\epsilon}{\rho}|\Gamma|$. This is the event where the set of available servers is not too small.

Let us fix a graph g_n. We can write

$$\begin{aligned} q(g_n) =& \mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right)\right) \\ \le& \mathbb{P}\left(\mathcal{B}\cap\mathcal{C}\right)\mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right)\middle|\mathcal{B}\cap\mathcal{C}\right) + \left(1 - \mathbb{P}\left(\mathcal{B}\cap\mathcal{C}\right)\right) \\ \le& \mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right)\middle|\mathcal{B}\cap\mathcal{C}\right) + \left(1 - \mathbb{P}\left(\mathcal{B}\right)\right) + \left(1 - \mathbb{P}(\mathcal{C})\right). \end{aligned} \tag{4.25}$$

The rest of the proof consists of the following two steps.

1. Conditioning on the occurrence of the events $\mathcal{B}$ and $\mathcal{C}$, we apply Lemma 4.10 to show that there exists a large set of graphs (w.r.t. $\mathbb{P}_{n,d(n)/n}(\cdot)$), $\tilde{\mathcal{G}}_n$, so that given any $g_n \in \tilde{\mathcal{G}}_n$, with high probability, the random demand vector $\tilde{\lambda}$ is feasible over the random subgraph $g_n|_{\Gamma\cup\Delta}$.

2. We then show that the event $\mathcal{B}\cap\mathcal{C}$ occurs with high probability.

Step 1. Let $\overline{\lambda}$ be the $m(n) \times 1$ vector that is an extension of $\tilde{\lambda}$, with

$$\overline{\lambda}_s = \begin{cases} \tilde{\lambda}_s, & 1 \le s \le |\Gamma|. \\ 0, & |\Gamma| < s \le m(n). \end{cases} \tag{4.26}$$

Note that by construction, the batch consists of ρb_n jobs, and hence $|\Gamma|$ is at most ρb_n, which is less than $m(n)$. By the definition of $\overline{\lambda}$, we have that, conditional on event $\mathcal{B}$,

$$\hat{\lambda} \in \mathbf{\Xi}^{m(n)}\left(\rho'\right), \tag{4.27}$$

with $\rho' = \frac{\rho}{\rho+\frac{1}{2}\epsilon}$. Similarly, conditional on event $\mathcal{C}$, the size of Δ is at least $m(n)$. Therefore, there exists a random variable H taking values in $\mathcal{M}_{m(n),n}$, so that

$$\mathbb{P}\left(\Gamma \subset H \text{ and } H \cap J \subset \Delta \,|\, \mathcal{B} \cap \mathcal{C}\right) = 1, \tag{4.28}$$

Let $\psi_{m(n),n}$ be the distribution of $(\overline{\lambda}, H)$ conditional on $(\mathcal{B} \cap \mathcal{C})$, over the set $\left(\Xi^{m(n)}(\rho'), \mathcal{M}_{m(n),n}\right)$. By Eq. (4.28) and the definition in Eq. (4.16), we have that

$$\mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right) \,|\mathcal{B} \cap \mathcal{C}\right) \le \mathbb{P}\left(\overline{\lambda} \notin \mathbf{R}\left(g_n|_H\right) \,|\mathcal{B} \cap \mathcal{C}\right) = l\left(g_n, \psi_{m(n),n}\right), \tag{4.29}$$

which, by the definition in Eq. (4.17), implies that

$$\mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right) \middle| \mathcal{B} \cap \mathcal{C}\right) \le l\left(g_n, \psi_{m(n),n}\right) \le n^{-3}, \tag{4.30}$$

for any $g_n \in \tilde{\mathcal{G}}_n\left(\psi_{m(n),n}\right)$. We now let the set $\tilde{\mathcal{G}}_n$ in the statement of Lemma 4.11 be the set $\tilde{\mathcal{G}}_n\left(\psi_{m(n),n}\right)$, for the measure $\psi_{m(n),n}$ defined earlier. According to Lemma 4.10, we have that

$$\mathbb{P}_{n,d_n/n}\left(\tilde{\mathcal{G}}_n\right) \to 1, \qquad \text{as } n \to \infty, \tag{4.31}$$

which establishes Eq. (4.18).

We now bound the value of $\mathbb{P}(\mathcal{C})$. We first note that $|\Gamma|$, the number of queues that receive at least one job from the batch, is no larger than the total number of jobs in a batch, ρb_n. Then, the inequality $(\rho + \frac{1}{2}\epsilon)b_n \ge \frac{\rho+\epsilon/2}{\rho}|\Gamma|$ is always true. It therefore suffices to analyze the probability that $|\Delta|$ is large. Recall that the length of each service slot is $\frac{b_n}{n}(\rho + \epsilon)$. This implies that for all $j \in J$,

$$\mathbb{P}\left(j \in \Delta\right) = 1 - \exp\left(-\frac{b_n}{n}(\rho + \epsilon)\right) \overset{(a)}{\sim} (\rho + \epsilon)\frac{b_n}{n}, \tag{4.32}$$

where $\left(1 - \exp\left(-\frac{b_n}{n}(\rho + \epsilon)\right)\right)$ is the probability of having at least one event point for a given server during a service slot, and step (a) follows from the Taylor series approximation of $\exp(x) \sim 1 + x$ as $x \downarrow 0$, and the fact that $b_n \ll n$ (cf. Eq. (4.7)). We then obtain

$$
\begin{aligned}
1 - \mathbb{P}(C) =& \mathbb{P}\left(\sum_{j \in J} \mathbb{I}(j \in \Delta) \le \left(\rho + \frac{1}{2}\epsilon\right) b_n \right) \\
\overset{(a)}{\sim}& \mathbb{P}\left(\sum_{j \in J} \mathbb{I}(j \in \Delta) \le \frac{\rho + \frac{1}{2}\epsilon}{\rho + \epsilon} \mathbb{E}\left(\sum_{j \in J} \mathbb{I}(j \in \Delta) \right) \right) \\
\overset{(b)}{\le}& \exp\left(-\frac{1}{2} b_n \left(\frac{\frac{1}{2}\epsilon}{\rho + \epsilon} \right)^2 \right) \\
=& \exp\left(-\frac{1}{2} \left(\frac{\frac{1}{2}\epsilon}{\rho + \epsilon} \right)^2 K_n \frac{n}{y_n} \right) \\
=& \exp\left(-\frac{1}{2} \left(\frac{\frac{1}{2}\epsilon}{\rho + \epsilon} \right)^2 K_n \frac{n}{d_n} \ln n \right) \\
\le& n^{-3},
\end{aligned}
\tag{4.33}
$$

whenever $K_n \ge 6\left(\frac{\rho+\epsilon}{\frac{1}{2}\epsilon}\right)^2 d_n/n = 24\left(1 + \frac{\rho}{\epsilon}\right)^2 d_n/n$ (cf. Eq. (4.4)). Step (a) follows from Eq. (4.32) and (b) from the Chernoff bound, $\mathbb{P}\left(X \le (1 - \delta)\mu\right) \le \exp\left(-\frac{1}{2}\delta^2 \mu\right)$, where X is a binomial random variable with $\mathbb{E}(X) = \mu$.

For the value of $\mathbb{P}(\mathcal{B})$, we have the following lemma, whose proof is given in Appendix A.1.5.

Lemma 4.12. *Suppose that*

1. $u_n \le \frac{1-\rho}{16} d_n / \ln n$, *and*

2. $b_n = K_n \frac{n \ln n}{d_n}$, *with* $K_n \ge \frac{24}{1-\rho}(\alpha + 1)\ln n$.

Let M_i be the number of jobs arriving to queue i within a batch. For any $\alpha > 0$, have that

$$
1 - \mathbb{P}\left(\max_{1 \le i \le I} M_i > \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n} \right) \lesssim n^{-\alpha}. \tag{4.34}
$$

In particular, setting $\alpha = 3$ in Lemma 4.12, we have that, because $K_n \geq \frac{96}{1-\rho}\ln n$ (cf. Eq. (4.4)),

$$1 - \mathbb{P}(\mathcal{B}) \lesssim n^{-3}. \tag{4.35}$$

Substituting Eqs. (4.30), (4.33), and (4.35) into Eq. (4.25), we obtain that

$$\begin{aligned} q(g_n) \leq & \mathbb{P}\left(\tilde{\lambda} \notin \mathbf{R}\left(g_n|_{\Gamma\cup\Delta}\right) \middle| \mathcal{B} \cap \mathcal{C}\right) + (1 - \mathbb{P}(\mathcal{B})) + (1 - \mathbb{P}(\mathcal{C})) \\ \lesssim & n^{-3} + n^{-3} + n^{-3} \\ \lesssim & n^{-3}. \end{aligned} \tag{4.36}$$

This completes the proof of Lemma 4.11. □

4.4.4 Service Time Statistics

We are now ready to bound the mean and variance of the service time distribution for the virtual queue, using Eqs. (4.10) and (4.12), and Lemmas 4.8 and 4.11.

Lemma 4.13. (Service Time Statistics for the Virtual Queue) *Fix some $\theta \in (0,1)$. Assume that $g_n \in \tilde{\mathcal{G}}_n$, and let $b_n = K_n n / y_n$. The service times of the batches, $S^M(k)$, are i.i.d., with*

$$\mathbb{E}\left(S^M(k)\right) \sim (\rho + \epsilon) \cdot \frac{K_n}{y_n}, \quad \text{and} \quad \mathrm{Var}\left(S^M(k)\right) \lesssim \frac{K_n^2}{y_n^2}.$$

Proof. Combining Eqs. (4.10) and (4.12) (for S_{col}), and Lemmas 4.8 (for S_{cle}) and

4.11 (for $q(g_n)$), we have

$$\begin{aligned}\mathbb{E}\left(S^M(k)\right) &\overset{(a)}{=} \mathbb{E}\left(S_{col}\right) + \mathbb{E}\left(X\right)\mathbb{E}\left(S_{cle} \middle| X = 1\right) \\ &= (\rho + \epsilon) \cdot \frac{K_n}{y_n} + q(g_n)\mathbb{E}\left(S_{cle} \middle| X = 1\right) \\ &\leq (\rho + \epsilon) \cdot \frac{K_n}{y_n} + Cn^{-3}\frac{n}{y_n} \\ &= (\rho + \epsilon) \cdot \frac{K_n}{y_n} + \frac{C}{n^2} \cdot \frac{1}{y_n}, \\ &\sim (\rho + \epsilon) \cdot \frac{K_n}{y_n}, \end{aligned} \tag{4.37}$$

where C is some positive constant. Similarly, using the fact that S_{col} is deterministic and that $\mathbb{E}(S_{cle}^2 \,|\, X = 1) \lesssim b_n^2$ (Lemma 4.8), we obtain that

$$\begin{aligned}\mathrm{Var}\left(S^M(k)\right) &= \mathrm{Var}\left(X \cdot S_{cle}\right) \\ &\leq \mathbb{E}\left(X S_{cle}^2\right) \\ &\leq q(g_n)\mathbb{E}\left(S_{cle}^2 \middle| X = 1\right) \\ &\lesssim n^{-3}b_n^2 \\ &\leq n^{-2}b_n^2 \\ &= \frac{K_n^2}{y_n^2}. \end{aligned} \tag{4.38}$$

This completes the proof. □

4.4.5 Delay Bound for the Waiting Time in the Virtual Queue

Let W_M be the steady-state **waiting time of a batch in the virtual queue**, as defined in Definition 4.4. Recall that $\hat{\mathcal{G}}_n$ is a set of graphs, where any $g_n \in \hat{\mathcal{G}}_n$ admits a full matching (cf. Lemma 4.3), and $\tilde{\mathcal{G}}_n$ is a set of graphs, where the probability of not having an assignment from the batch to the set of available servers at the end of a COLLECT service slot, $q_n(g_n)$, is small for any $g_n \in \tilde{\mathcal{G}}_n$ (cf. Lemma 4.11). The following is the main result of this subsection.

Proposition 4.14. (Delays in the Virtual Queue) *If* $g_n \in \hat{\mathcal{G}}_n \cap \tilde{\mathcal{G}}_n$, *then*

$$\mathbb{E}(W_M) \lesssim \frac{K_n}{y_n}. \tag{4.39}$$

Proof. Recall that by Lemma 4.7, the waiting times in the virtual queue are bounded above, almost surely, by the waiting times in a $GI/GI/1$ queue with interarrival times $\{A(k)\}_{k\geq 1}$ and service times $\{S^M(k)\}_{k\geq 1}$. It hence suffices for us to bound the steady-state expected waiting time of the latter. We use Kingman's bound [51], which states that the expected waiting time in steady state for a $GI/GI/1$ queue, W, is bounded by $\mathbb{E}(W) \leq \tilde{\lambda}\frac{\sigma_a^2+\sigma_s^2}{2(1-\tilde{\rho})}$, where $\tilde{\lambda}$ is the arrival rate, $\tilde{\rho}$ is the traffic intensity, and σ_a^2 and σ_s^2 are the variances for the interarrival times and service times, respectively. Using Lemmas 4.5 and 4.13, we have

$$\begin{aligned}
\tilde{\lambda} &= \frac{1}{\mathbb{E}(A(k))} \lesssim \frac{n}{b_n} = \frac{y_n}{K_n}, \\
\tilde{\rho} &= \frac{\mathbb{E}(S^M(k))}{\mathbb{E}(A(k))} \sim \frac{(\rho+\epsilon)\frac{K_n}{y_n}}{\frac{K_n}{y_n}} = \rho + \epsilon < 1, \\
\sigma_a^2 &= \mathrm{Var}(A(k)) \sim \frac{1}{\rho}\frac{b_n}{n^2} \lesssim \frac{K_n}{ny_n}, \\
\sigma_s^2 &= \mathrm{Var}\left(S^M(k)\right) \lesssim \frac{K_n^2}{y_n^2},
\end{aligned}$$

for all sufficiently large values of n. Using these inequalities in Kingman's bound, we obtain

$$\mathbb{E}(W_M) \lesssim \frac{y_n}{K_n}\left(\frac{K_n}{ny_n} + \frac{K_n^2}{y_n^2}\right) \lesssim \frac{K_n}{y_n}.$$

□

4.5 Proof of Theorem 3.6

Proof. The total queueing delay of a job is no more than the time to form a batch plus the waiting time in the virtual queue. If $g_n \in \hat{\mathcal{G}}_n \cap \tilde{\mathcal{G}}_n$, and $\lambda_n \in \mathbf{\Lambda}_n(u_n)$, then using Lemma 4.5 and Proposition 4.14, we obtain that, there exists $n_0 > 0$, so that

$$\begin{aligned}\mathbb{E}_\pi\left(W|g_n,\lambda_n\right) \leq & \mathbb{E}\left(A(1)\right) + \mathbb{E}\left(W^M\right) \\ \leq & C\frac{K_n}{y_n} \\ = & CK_n\frac{\ln n}{d_n} \\ \overset{(a)}{\leq} & K\max\left\{d_n/n, \ln n\right\}\frac{\ln n}{d_n} \\ \leq & K\frac{\ln^2 n}{d_n},\end{aligned} \tag{4.40}$$

for all $n \geq n_0$, where C and K are positive constants that do not depend on n, g_n, and λ_n. Step (a) follows from the fact that $K_n = \max\left\{24\left(1+\frac{\varrho}{\epsilon}\right)^2 d_n/n\,,\, \frac{96}{1-\rho}\ln n\right\}$ (cf. Eq. (4.4)). Furthermore, by the weak law of large numbers, there exist $\epsilon_n \downarrow 0$, such that

$$\lim_{n\to\infty}\mathbb{P}\left(1-\epsilon_n \leq \frac{\deg\left(G_n\right)}{d_n} \leq 1+\epsilon_n\right) = 1, \tag{4.41}$$

where G_n is a $(n, d_n/n)$ random graph. Let $\mathcal{L}_n = \left\{g_n \in \mathcal{G}_n : \frac{\deg(g_n)}{d_n} \in \left[1-\epsilon_n, 1+\epsilon_n\right]\right\}$. We have that $\mathbb{P}_{n,d_n/n}\left(\mathcal{L}_n\right) \to 1$ (Eq. (4.41)), $\mathbb{P}_{n,d_n/n}(\hat{\mathcal{G}}_n) \to 1$ (Lemma 4.3), and $\mathbb{P}_{n,d_n/n}(\tilde{\mathcal{G}}_n) \to 1$ (Lemma 4.11), as $n \to \infty$. Let $\mathcal{H}_n = \hat{\mathcal{G}}_n \cap \tilde{\mathcal{G}}_n \cap \mathcal{L}_n$. It follows that

$$\mathbb{P}_{n,d_n/n}\left(\mathcal{H}_n\right) \geq 1 - \delta_n, \tag{4.42}$$

for all n, for some $\delta_n \downarrow 0$. Note that the definitions of $\hat{\mathcal{G}}_n$, $\tilde{\mathcal{G}}_n$ and $\mathcal{L}_n$ do not involve the arrival rates λ_n, and hence δ_n does not depend on λ_n. Eqs. (4.40) and (4.42) complete the proof of Theorem 3.6. □

Chapter 5

Queueing with Future Information

Starting from this chapter, we shall shift our emphasis from the issue of *flexible architecture*, to that of *information.* Our inquiries have two main motivations. The first motivation stems from the simple fact that real-time information can be difficult to obtain. When the system size becomes large, the infrastructure needed to support complete information sharing among all components can quickly become prohibitively expensive. Therefore, in many large-scale flexible systems, it can be a practical imperative that we understand whether one can devise efficient policies with only limited information sharing, and still achieve performance that is competitive with a centralized policy with full information sharing. This line of reasoning will be explored in Chapter 7 in the form of designing optimal decentralized scheduling policies for partially flexible systems.

Besides the difficulty of information sharing, there lies a more "positive" motivation towards the other side of the "information availability" spectrum. In addition to asking what to do when information is *limited*, we would also like to know whether it is possible to harness performance gains when *more information* becomes available.

While there are many ways to define "more," in this report, we shall investigate increments of information along the axis of *time*, that is, more information about the *future*. This will be the focus of the current chapter, as well as the next.

It turns out that a moderate amount of *predictive information* can enable substantial performance improvements in flexible systems. We will demonstrate that, for a class of single-queue admission control problems, the decision maker can leverage *future information* about the arrivals and services to drastically reduce the heavy-traffic delay compared to that of an *optimal* online policy, even when the future information is restricted to a *finite* lookahead window starting from the current time frame.

The admission control model is, in fact, closely related to our understanding of flexible resource allocation systems, although this may not seem immediately obvious. We will show, in Chapter 7, that the single-queue admission control model is essentially equivalent to the problem faced by a local queue in a Partial Pooling architecture, as described in Section 2.2. Therefore, the benefits of future information also apply to that family of partially flexible systems.

5.1 Introduction

5.1.1 Variable, but Predictable

Two important ingredients often make the design and analysis of a queueing system difficult: the demand and the resources can be both *variable* and *unpredictable*. *Variability* refers to the fact that job arrivals or the availability of resources can be highly volatile and non-uniform across the time horizon. *Unpredictability* means that the exact type of non-uniformity is not known to the decision maker ahead of time, and she is obliged to make allocation decisions only based on the state of the system at the moment, together with some statistical estimates about the future.

While the world will always be volatile, in many cases, the amount of unpredictability about the future may be reduced thanks to *forecasting* technologies and the increasing availability of data. For instance,

1. Advance booking in the hotel and textile industries allows for accurate demand forecasts [31].

2. The availability of monitoring data enables traffic controllers to predict the traffic pattern around potential bottlenecks [75].

3. Advance scheduling for elective surgeries could inform care providers several weeks before the intended appointment [50].

In all of these examples, future demand remains *exogenous* and variable, yet the decision maker can obtain some information about their realizations, ahead of time.

Is there significant performance gain to be harnessed by "looking into the future"? In this chapter we provide a largely affirmative answer, in the context of a class of admission control problems.

5.1.2 Admission Control Viewed as Resource Allocation

We begin by informally describing our problem. Consider a single queue equipped with a server that runs at a rate of $1-p$ jobs per unit time, where p is a fixed constant in $(0,1)$, as depicted in Figure 5-1. The queue receives a stream of incoming jobs, arriving at rate $\lambda \in (0,1)$. If $\lambda > 1-p$, the arrival rate is greater than the server's processing rate, and some form of *admission control* is necessary in order to keep the system stable. In particular, upon its arrival to the system, a job will either

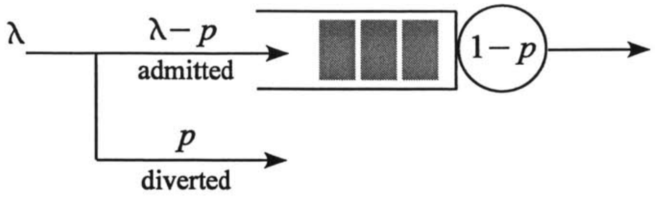

Figure 5-1: An illustration of the admission control problem, with a constraint on the a rate of diversion.

be *admitted* to the queue, or *diverted.* In the latter case, the job does not join the queue, and, from the perspective of the queue, disappears from the system entirely. The goal of the decision maker is to minimize the average delay experienced by the admitted jobs, while obeying the constraint that the average rate at which jobs are diverted *does not exceeded* p. [1]

While the admission control problem is described above as a stand-alone system, we can also interpret it as a form of interaction between two types of resources. In particular, one can think of our problem as one of *resource allocation*, where a decision maker tries to match incoming demands with either (1) a *slow local resource* that corresponds to the server, or (2) a *fast external resource* that can process any job diverted to it almost instantaneously. Both types of resources are *constrained*, in the sense that their capacities ($1-p$ and p, respectively) cannot change over time, due to physical or contractual restrictions. The processing time of a job at the fast resource is *negligible compared to that at the slow resource*, as long as the rate of diversion to the fast resource stays below p in the long run. Under this interpretation, minimizing the average delay across *all* jobs is equivalent to minimizing the average delay across just the *admitted* jobs, since the jobs diverted to the fast resource can be thought of

[1] Note that as $\lambda \to 1$, the rate of admitted jobs, $\lambda - p$, approaches the server's capacity $1-p$, and hence we will refer to the system's behavior when $\lambda \to 1$ as the *heavy-traffic regime*.

as being processed immediately and experiencing no delay at all.

For a more concrete example, consider a web service company that enters a long term contract with an external cloud computing provider for a fixed amount of computation resources (e.g., virtual machine instance time) over the contract period.[2] During the contract period, any incoming request can be either served by the in-house server (slow resource), or be diverted to the cloud (fast resource), and in the latter case, the job does not experience congestion delay since the scalability of the cloud allows for multiple VM instances to be running in parallel (and potentially on different physical machines). The decision maker's constraint is that the total amount of diverted jobs to the cloud must stay below the amount prescribed by the contract, which, in our case, translates into a maximum diversion rate over the contract period. Similar scenarios can also arise in other domains, where the slow versus fast resources could, for instance, take the form of:

1. an in-house manufacturing facility, versus an external contractor;
2. a slow toll booth on the freeway, versus a special lane that lets a car pass without paying the toll;
3. hospital bed resources within a single department, versus a cross-departmental central bed pool.

Connections to Partially Flexible Systems. What does the admission control problem have to do with our study of flexibility? As it turns out, an essentially

[2] *Example.* As of September 2012, Microsoft's Windows Azure cloud services offer a 6-month contract for $71.99 per month, where the client is entitled for up to 750 hours of virtual machine (VM) instance time each month, and any additional usage would be charged at a 25% higher rate. Due to the large scale of the Azure data warehouses, the speed of any single VM instance can be treated as roughly constant, and independent of the total number of instances that the client is running concurrently.

equivalent decision problem arises in the Partial Pooling system that we introduced earlier in Section 2.2, where the notion of "fast resource" manifests itself in the form of a *flexible server pool*. An illustration of the Partial Pooling system is duplicated in Figure 5-2 for convenience.

We now explain the connection informally. Recall that in a Partial Pooling system, a fraction p of a total of n units of processing resources is *fully flexible*, and takes the form of a central server pool that collectively operates at rate pn, while the remaining $1-p$ fraction is *inflexible* and dedicated to the corresponding local queues (Figure 5-2). Under this framework, the admission control problem studied in this chapter is essentially the problem faced by each one of the local queues, that is, deciding whether an incoming job should be queued locally, or "diverted" to a "central queue," and ultimately served by one of the flexible servers at the central server pool (Figure 5-3). When n is large, the central server pool operates at a significantly faster speed than the local inflexible servers. As a result, we can expect that, as long as each local queue diverts at a rate that is strictly less than p, the queueing delay at the central queue vanishes as $n \to \infty$, thus becoming the "fast resource" in our earlier interpretation. This connection to the Partial Pooling model will be explored in greater detail in Section 7.3.3, where the above intuition will be made rigorous.

5.1.3 Overview of Main Contributions

We preview our main results in this section. The formal statements will be given in Section 5.3.

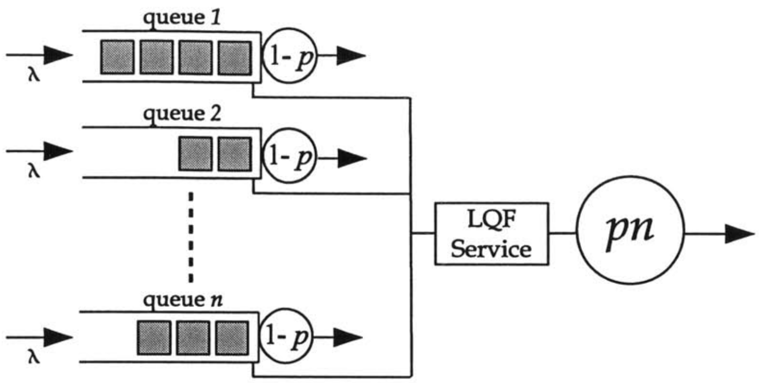

Figure 5-2: Illustration of the Partial Pooling model with flexible and inflexible resources, [84].

Summary of the Problem

We consider a continuous-time admission control problem, depicted in Figure 5-1. The problem is characterized by three parameters: λ, p, and w:

1. Jobs arrive to the system at a rate of λ jobs per unit time, with $\lambda \in (0,1)$. The server operates at a rate of $1-p$ jobs per unit time, with $p \in (0,1)$.
2. The decision maker determines whether each arriving job is admitted to the queue or diverted, with the goal of minimizing the time-average queue length[3], subject to the constraint that the time-average diversion rate does not exceed p jobs per unit time.
3. The decision maker has access to *information about the future*, which takes the form of a *lookahead window* of length $w \in \mathbb{R}_+$. In particular, at any time t, the times of arrivals and service availability within the interval $[t, t+w]$ are

[3] By Little's Law, the average queue length is essentially the same as average delay, up to a constant factor. See Section 5.2.4.

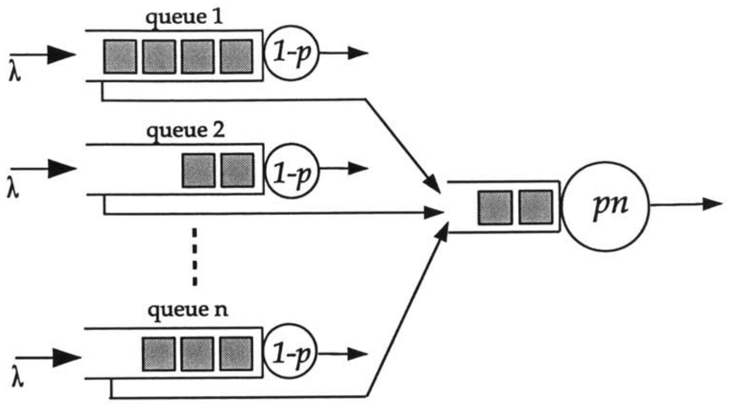

Figure 5-3: Resource pooling using a central queue.

revealed to the decision maker. We will consider the following cases for w.

(a) $w = 0$, the *online problem*, where no future information is available.

(b) $w = \infty$, the *offline problem*, where the entire future has been revealed.

(c) $0 < w < \infty$, where the future is revealed only over a finite lookahead window.

Throughout, we will fix $p \in (0, 1)$, and will be primarily interested in the system's behavior in the *heavy-traffic regime* of $\lambda \to 1$.

Overview of Main Results

Our main contribution is to demonstrate that the performance of a diversion policy is highly sensitive to the amount of future information available, measured by the value of w.

Fix $p \in (0, 1)$, and let the arrival and service processes be Poisson. For the online problem ($w = 0$), we show that the optimal time-average queue length, C_0^{opt}, approaches infinity in the heavy-traffic regime, at the rate

$$C_0^{opt} \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1.$$

In sharp contrast, the optimal average queue length among offline policies ($w = \infty$), C_∞^{opt}, converges to a *constant*,

$$C_\infty^{opt} \to \frac{1-p}{p}, \quad \text{as } \lambda \to 1,$$

and this limit is achieved by a so-called No-Job-Left-Behind policy. Figure 5-4 illustrates this difference in delay performance for a particular value of p.

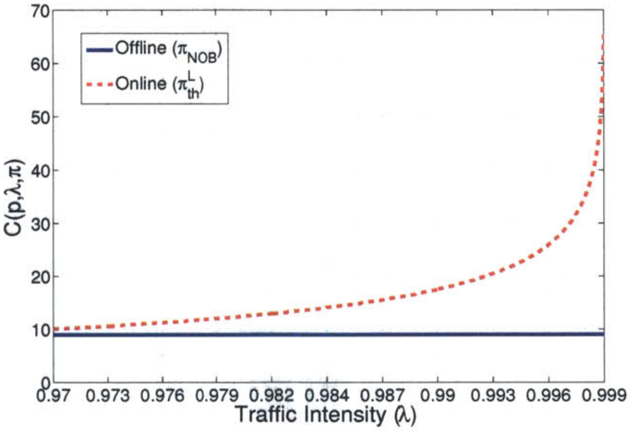

Figure 5-4: Comparison of heavy-traffic delay scaling between optimal online and offline policies, with $p = 0.1$ and $\lambda \to 1$. The value plotted is the resulting average queue length as a function of λ.

We then show that the No-Job-Left-Behind policy for the offline problem can be modified, so that the *same* optimal heavy-traffic limit of $\frac{1-p}{p}$ is achieved even with a *finite* lookahead window, w_λ, where

$$w_\lambda = \mathcal{O}\left(\log \frac{1}{1-\lambda}\right), \quad \text{as } \lambda \to 1.$$

This is of practical importance, because in any realistic application only a finite amount of future information can be obtained.

Finally, we provide a matching *lower bound on future information*, which states that it *necessary* to have

$$w_\lambda = \Omega\left(\log \frac{1}{1-\lambda}\right), \quad \text{as } \lambda \to 1 \tag{5.1}$$

in order to achieve *any* substantial improvement over the online policy. Combined with the upper bound on w_λ, this implies that system performance depends *critically* on the amount of future information available.

On the methodological end, we use a sample-path-based framework to analyze the performance of the offline and finite-lookahead policies, borrowing tools from renewal theory and the theory of random walks. We believe that our techniques can be substantially generalized to incorporate general arrival and service processes, diffusion approximations, as well as observation noise. See Section 5.8 for a more elaborate discussion.

5.1.4 Related Work

There is an extensive body of work devoted to various Markov (or *online*) admission control problems; the reader is referred to the survey [80], and references therein. Typically, the problem is formulated as a Markov decision problem (MDP), where the decision maker, by admitting or rejecting incoming jobs, seeks to maximize a long-term average objective consisting of rewards (e.g., throughput) minus costs (e.g., waiting time experienced by a customer). The case where the maximization is performed subject to a constraint on some average cost has also been studied, and it has been shown, for a family of reward and cost functions, that an optimal policy assumes a "threshold-like" form, where the decision maker diverts the next job only if the current queue length is greater than or equal to L, with possible randomization if at level $L-1$, and always admits the job if below $L-1$ (cf. [12]). Indeed, our problem, where one tries to minimize average queue length (delay) subject to a lower-bound on the throughput (i.e., a maximum diversion rate), can be shown to belong to this category, and the online heavy-traffic scaling result can be obtained with moderate effort within the MDP framework.

However, the resource allocation interpretation of our admission control problem as that of matching jobs with fast and slow resources, and, in particular, its connection to resource pooling in the many-server limit, seems to be largely unexplored. The difference in motivation perhaps explains why the optimal online heavy-traffic delay scaling of $\log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ that emerges by fixing p and taking $\lambda \to 1$ has not appeared in the literature, to the best our knowledge.

There is also an extensive literature on *competitive analysis*, which focuses on the *worst-case* performance of online algorithms compared to that of an optimal offline version (where one knows the entire input sequence). The reader is referred to [16] for a comprehensive survey, and the references therein on packing-type problems, such as load balancing and machine scheduling [8], and call admission and routing [7], which are more closely related to our problem. While our optimality result for the policy with a finite lookahead window is stated in terms of the *average* performance under stochastic inputs, we believe that the analysis can be extended to yield worst-case competitive ratios under certain input regularity conditions.

In sharp contrast to our understanding of online problems, much less is known for settings in which information about the future is taken into consideration. In [66], the author considers a variant of the flow control problem where the decision maker knows the job size of the arriving customer, as well as the arrival time and job size of the next customer, with the goal of maximizing certain discounted or average reward. A characterization of an optimal stationary policy is derived under a standard semi-Markov decision problem framework, which is possible because the lookahead is limited to the next arriving job. In [21], the authors consider a scheduling problem with one server and M parallel queues, motivated by applications in satellite systems where the link qualities between the server and the queues vary over time. The authors compare the throughput of several online policies with that of an offline policy that has access to all future instances of link qualities. However, the offline policy takes the form of a Viterbi-like dynamic program, which, while being throughput-optimal by definition, provides limited qualitative insight.

One challenge that arises as one tries to move beyond the online setting is that policies with lookahead typically do not admit a clean Markovian description, and hence common techniques for analyzing Markov decision problems do not easily apply. To circumvent this obstacle, we will first relax our problem to be fully offline, which turns out to be surprisingly amenable to analysis. We then use the insights from the optimal offline policy to construct an optimal policy with a finite lookahead window, in a rather straightforward manner.

The idea of exploiting future information or predictions to improve decision making has also been explored in other application domains. Advance reservations (a form of future information) have been studied in lossy networks [24, 55] and, more recently, in revenue management [54]. Using simulations, [50] demonstrates that the use of a one- and two-week advance scheduling window for elective surgeries can improve efficiency at the associated intensive care unit (ICU). The benefits of an advanced booking program for supply chains have been shown in [31], in the form of reduced demand uncertainties. While similar in spirit, the motivation and dynamics in these models are very different from ours.

Finally, our formulation in terms of slow and fast resources has been in part inspired by the literature on resource pooling systems, where one improves overall system performance by (partially) sharing individual resources. The connection of our problem to the Partial Pooling model in [84] is discussed in detail in Section 7.3.3. For the general topic of resource pooling, interested readers are referred to [11, 41, 59, 60] and the references therein.

5.1.5 Organization of the Chapter

The rest of the chapter is organized as follows. The mathematical model for our problem is described in Section 5.2. Section 5.3 contains the statements of our main results, and introduces the No-Job-Left-Behind policy (π_{NOB}), which will be a central object of study for this chapter. Section 5.4 presents two alternative descriptions of the No-Job-Left-Behind policy that have important structural, as well as algorithmic, implications. Sections 5.5 through 5.7 are devoted to the proofs for the results concerning the online, offline and finite-lookahead policies, respectively. The proof for the lower bound on future information requires a fairly different style of analysis, and will be treated exclusively in Chapter 6. Finally, Section 5.8 contains some concluding remarks and future directions.

5.2 Model and Setup

5.2.1 System Dynamics

An illustration of the system setup was given in Figure 5-1. The system consists of a single-server queue running in continuous time ($t \in \mathbb{R}_+$), with an unbounded buffer that stores all unprocessed jobs. The queue is assumed to be empty at $t = 0$.

Jobs arrive to the system according to a Poisson process with rate $\lambda \in (0,1)$, so that the intervals between two adjacent arrivals are independent and exponentially distributed with mean $\frac{1}{\lambda}$. We will denote by $\{A(t) : t \in \mathbb{R}_+\}$ the cumulative arrival process, where $A(t) \in \mathbb{Z}_+$ is the total number of arrivals to the system by time t.

The processing of jobs by the server is modeled by a Poisson process of rate $1-p$. When the service process makes a jump at time t, we say that a service token is generated. If the queue is not empty at time t, exactly one job "consumes" the service token and leaves the system immediately. Otherwise, the service token is wasted and has no impact on the future evolution of the system. [4] We will denote by $\{S(t) : t \in \mathbb{R}_+\}$ the cumulative token generation process, where $S(t) \in \mathbb{Z}_+$ is the total number of service tokens generated by time t.

When $\lambda > 1-p$, in order to maintain the stability of the queue, a decision maker has the option of "diverting" a job *at the moment of its arrival.* Once diverted, a job is removed from the system. Finally, the decision maker is allowed to divert up to a time-average rate of p.

[4] For our purpose, it is important to note a key assumption implicit in the service token formulation: the processing times are intrinsic to the server, and *independent* of the job being processed. For instance, the sequence of service times will not depend on the order in which the jobs in the queue are served, so long as the server remains busy throughout the period. This distinction is of little relevance for an $M/M/1$ queue, but can be important in our case, where the diversion decisions may depend on the future. See discussion in Section 5.8.

5.2.2 Initial Sample Path

Let $\{Q^0(t) : t \in \mathbb{R}_+\}$ be the continuous-time queue length process, where $Q^0(t) \in \mathbb{Z}_+$ is the queue length at time t if *no diversion* is applied at any time. We say that an *event* occurs at time t, if there is either an arrival, or a generation of service token, at time t. Let T_k, $k \in \mathbb{N}$, be the time of the kth event in the system. Denote by $\{Q^0[k] : k \in \mathbb{Z}_+\}$ the embedded discrete-time process of $\{Q^0(t)\}$, where $Q^0[k]$ is the length of the queue sampled immediately after the kth event, [5]

$$Q^0[k] = Q^0(T_k+), \quad k \in \mathbb{N}.$$

with the initial condition $Q^0[0] = 0$. It is well-known that Q^0 is a reflected random walk on $\mathbb{Z}_+$, such that for all $x_1, x_2 \in \mathbb{Z}_+$ and $k \in \mathbb{Z}_+$,

$$\mathbb{P}\left(Q^0[k+1] = x_2 \mid Q^0[k] = x_1\right) = \begin{cases} \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = -1, \\ 0, & \text{otherwise}, \end{cases} \tag{5.2}$$

if $x_1 > 0$, and

$$\mathbb{P}\left(Q^0[k+1] = x_2 \mid Q^0[k] = x_1\right) = \begin{cases} \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{5.3}$$

if $x_1 = 0$. Note that, when $\lambda > 1 - p$, the random walk Q^0 is transient.

[5] The notation $f(x+)$ denotes the right-limit of f at x : $f(x+) = \lim_{y \downarrow x} f(y)$. In this particular context, the values of $Q^0[k]$ are well defined, since the sample paths of Poisson processes are right-continuous-with-left-limits (RCLL) almost surely.

The process Q^0 contains *all relevant information* in the arrival and service processes, and will be the main object of study of this chapter. We will refer to Q^0 as the *initial sample path* throughout the chapter, to distinguish it from sample paths obtained after diversions have been made.

5.2.3 Diversion Policies

Since a diversion can only take place when there is an arrival, it suffices to define the locations of diversions with respect to the discrete-time process $\{Q^0[k] : k \in \mathbb{Z}_+\}$, and throughout, our analysis will focus on discrete-time queue length processes unless otherwise specified. Let $\Phi(Q)$ be the locations of all arrivals in a discrete-time queue length process Q, i.e.,

$$\Phi(Q) = \{k \in \mathbb{N} : Q[k] > Q[k-1]\},$$

and for any $M \subset \mathbb{Z}_+$, define the counting process $\{I(M,k) : k \in \mathbb{N}\}$ associated with M as[6]

$$I(M,k) = |\{1,\ldots,k\} \cap M|. \tag{5.4}$$

Definition 5.1 (Feasible Diversion Sequence). *The sequence $M = \{m_i\}$ is said to be a feasible diversion sequence with respect to a discrete-time queue length process, Q^0, if all of the following hold:*

1. *All elements in M are distinct, so that at most one diversion occurs at any slot.*

2. *$M \subset \Phi(Q^0)$, so that a diversion occurs only when there is an arrival.*

[6] $|X|$ denotes the cardinality of X.

3.

$$\limsup_{k\to\infty} \frac{1}{k} I(M,k) \leq \frac{p}{\lambda + (1-p)}, \quad a.s., \tag{5.5}$$

so that the time-average diversion rate is at most p.

In general, M is also allowed to be a finite set.

The denominator $\lambda + (1-p)$ in Eq. (5.5) is due to the fact that the total rate of events in the system is $\lambda + (1-p)$.[7] Analogously, the diversion rate in continuous time is defined by

$$r_d = (\lambda + 1 - p) \cdot \limsup_{k\to\infty} \frac{1}{k} I(M,k). \tag{5.6}$$

The impact of a diversion sequence to the evolution of the queue length process is formalized in the following definition.

Definition 5.2 (Diversion Maps). *Fix an initial queue length process $\{Q^0[k] : k \in \mathbb{N}\}$ and a corresponding feasible diversion sequence $M = \{m_i\}$.*

1. *The* **point-wise diversion map** $D_P(Q^0, m)$ *outputs the resulting process if a diversion is made to Q^0 in slot m, and only in that slot. Let $Q' = D_P(Q^0, m)$. Then,*

$$Q'[k] = \begin{cases} Q^0[k] - 1, & \text{if } k \geq m, \text{ and } Q^0[t] > 0, \forall t \in \{m, \ldots, k\}. \\ Q^0[k], & \text{otherwise,} \end{cases} \tag{5.7}$$

2. *The* **multi-point diversion map** $D(Q^0, M)$ *outputs the resulting process if all diversions in the set M are made to Q^0. Define Q^i recursively by Q^i =*

[7] This is equal to the total rate of jumps in $A(\cdot)$ and $S(\cdot)$.

$D_P\left(Q^{i-1}, m_i\right)$, $\forall i \in \mathbb{N}$. Then, $Q^{\infty} = D\left(Q^0, M\right)$ is defined as the pointwise limit

$$Q^{\infty}[k] = \lim_{i \to \min\{|M|, \infty\}} Q^i[k], \quad \forall k \in \mathbb{Z}_+. \tag{5.8}$$

The definition of the pointwise diversion map reflects the earlier assumption that the service time of a job only depends on the speed of the server at the moment and is independent of the job's identity (See Section 5.2). Note also that the value of $Q^{\infty}[k]$ depends only on the total number of diversions before k (Eq. (5.7)), which is at most k, and the limit in Eq. (5.8) is well-defined. Moreover, it is not difficult to see that the order in which diversions are made has no impact on the resulting sample path, as stated in the lemma below. The proof is omitted.

Lemma 5.3. *Fix an initial sample path Q^0, and let M and $\tilde{M}$ be two feasible diversion sequences that contain the same elements. Then $D\left(Q^0, M\right) = D\left(Q^0, \tilde{M}\right)$.*

We next define the notion of a diversion policy, which outputs a diversion sequence based on the (limited) knowledge of an initial sample path Q^0. Informally, a diversion policy is said to be w-lookahead if it makes its diversion decisions based on the knowledge of Q^0 up to w units of time into the future (in continuous time).

Definition 5.4 (w-Lookahead Diversion Policies). *Fix $w \in \mathbb{R}_+ \bigcup \{\infty\}$. Let $\mathcal{F}_t = \sigma\left(Q^0(s); s \le t\right)$ be the natural filtration induced by $\{Q^0(t) : t \in \mathbb{R}_+\}$, and $\mathcal{F}_{\infty} = \bigcup_{t \in \mathbb{Z}_+} \mathcal{F}_t$. A w-predictive diversion policy is a mapping, $\pi : {\mathbb{Z}_+}^{\mathbb{R}_+} \to \mathbb{N}^{\mathbb{N}}$, such that*

1. *The set $M = \pi\left(Q^0\right)$ is a feasible diversion sequence, a.s.;*

2. *The event $\{k \in M\}$ is $\mathcal{F}_{T_k + w}$ measurable, for all $k \in \mathbb{N}$.*

We will denote by Π_w the family of all w-lookahead diversion policies.

The parameter w in Definition 5.4 captures the amount of information that the diversion policy has about the future.

1. When $w = 0$, all diversion decisions are made solely based on the knowledge of the system until the current time frame. We will refer to Π_0 as *online policies.*

2. When $w = \infty$, the entire sample path of Q^0 is revealed to the decision maker ahead of time. We will refer to Π_∞ as *offline policies.*

3. We will refer to Π_w, where $0 < w < \infty$, as policies with a *lookahead window of size w.*

5.2.4 Performance Measure

Given a discrete-time queue length process Q and $k \in \mathbb{N}$, we denote by $S(Q,k) \in \mathbb{Z}_+$ the partial sum

$$S(Q,k) = \sum_{l=1}^{k} Q[l]. \tag{5.9}$$

Definition 5.5 (Average Post-diversion Queue Length). *Let Q^0 be an initial queue length process. Define $C(p,\lambda,\pi) \in \mathbb{R}_+$ as the expected time-average queue length after applying a diversion policy π:*

$$C(p,\lambda,\pi) = \mathbb{E}\left(\limsup_{k\to\infty} \frac{1}{k} S\left(Q_\pi^\infty, k\right)\right), \tag{5.10}$$

where $Q_\pi^\infty = D\left(Q^0, \pi\left(Q^0\right)\right)$, and the expectation is taken over all realizations of Q^0, and the randomness used by π internally, if any.

Remark: Delay versus Queue Length. By Little's Law, the long-term average waiting time of a typical customer in the queue is equal to the long-term average queue length divided by the arrival rate (independent of the service discipline of the server). Therefore, if our goal is to minimize the average waiting time of the jobs that remain after diversions, it suffices to use $C(p, \lambda, \pi)$ as a performance metric in order to judge the effectiveness of a diversion policy π. In particular, denote by $T_{all} \in \mathbb{R}_+$ the time-average queueing delay experienced by all jobs, where diverted jobs are assumed to have a delay of zero, then $\mathbb{E}(T_{all}) = \frac{1}{\lambda}C(p, \lambda, \pi)$, and hence the average queue length and delay coincide in the heavy-traffic regime, as $\lambda \to 1$. With an identical argument, it is easy to see that the average delay among *admitted* jobs, T_{adt}, satisfies $\mathbb{E}(T_{adt}) = \frac{1}{\lambda - r_d}C(p, \lambda, \pi)$, where r_d is the continuous-time diversion rate under π. Therefore, we may use the terms "delay" and "average queue length" interchangeably in the rest of the chapter, with the understanding that they represent essentially the same quantity up to a constant.

Finally, we define the notion of an optimal delay within a family of policies.

Definition 5.6 (Optimal Delay). *Fix $w \in \mathbb{R}_+$. We call $C^*_w(p, \lambda)$ the optimal delay in Π_w, where*

$$C^*_w(p, \lambda) = \inf_{\pi \in \Pi_w} C(p, \lambda, \pi). \tag{5.11}$$

5.3 Summary of Main Results

In this section, we state the main results of this chapter. The proofs will be presented in Sections 5.5 through 5.7.

5.3.1 Optimal Delay for Online Policies

Definition 5.7 (Threshold Policies). *We say that π^L_{th} is an L-threshold policy, if a job arriving at time t is diverted if and only if the queue length at time $t-$ is greater than or equal to L.*

The following theorem shows that the class of threshold policies achieves the optimal heavy-traffic delay scaling in Π_0.

Theorem 5.8 (Optimal Online Policies). *Fix $p \in (0,1)$, and let*

$$L(p,\lambda) = \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil.$$

Then,

1. $\pi_{th}^{L(p,\lambda)}$ *is feasible for all $\lambda \in (1-p,1)$.*
2. $\pi_{th}^{L(p,\lambda)}$ *is asymptotically optimal in Π_0 as $\lambda \to 1$:*

$$C\left(p,\lambda,\pi_{th}^{L(p,\lambda)}\right) \sim C_0^*(p,\lambda) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1.$$

Proof. See Section 5.5. □

5.3.2 Optimal Delay for Offline Policies

Given the sample path of a random walk Q, let $U(Q,k)$ the number of slots till Q reaches the level $Q[k]-1$ after slot k:

$$U(Q,k) = \min\{j \geq 1 : Q[k+j] = Q[k]-1\}. \tag{5.12}$$

Definition 5.9 (No-Job-Left-Behind Policy[8]). *Given an initial sample path Q^0, the No-Job-Left-Behind policy, denoted by π_{NOB}, diverts all arrivals in the set Ψ, where*

$$\Psi = \left\{k \in \Phi\left(Q^0\right) : U\left(Q^0,k\right) = \infty\right\}. \tag{5.13}$$

[8] The reason for choosing this name will be made in clear in Section 5.4.1, using the "stack" interpretation of this policy.

We will refer to the diversion sequence generated by π_{NOB} as $M^{\Psi} = \{m_i^{\Psi} : i \in \mathbb{N}\}$, where $M^{\Psi} = \Psi$.

In other words, π_{NOB} would divert a job arriving at time t if and only if the initial queue length process never returns to below the current level in the future, which also implies that

$$Q^0[k] \geq Q^0\left[m_i^{\Psi}\right], \quad \forall i \in \mathbb{N}, k \geq m_i^{\Psi}, \tag{5.14}$$

Examples of the π_{NOB} policy being applied to a particular sample path are given in Figures 5-5 and 5-6 (illustration), as well as in Figure 5-7 (simulation).

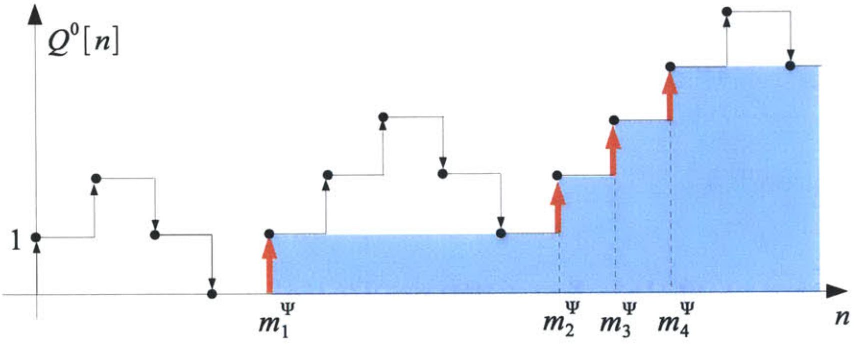

Figure 5-5: Illustration of applying π_{NOB} to an initial sample path, Q^0, where the diversions are marked by the bold arrows (in red).

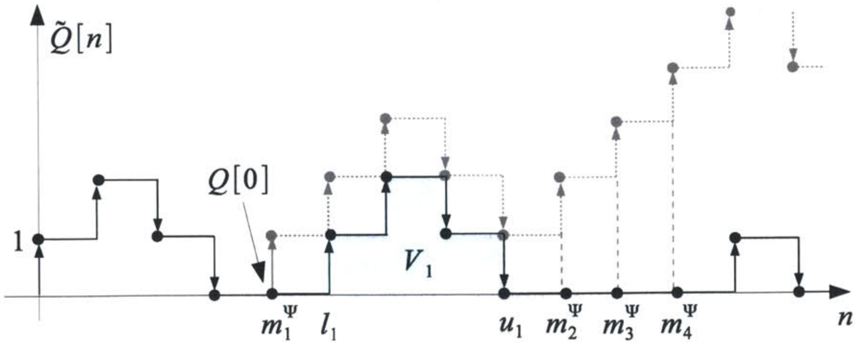

Figure 5-6: The solid lines depict the resulting sample path, $\tilde{Q} = D(Q^0, M^{\Psi})$, after applying π_{NOB} to Q^0.

It turns out that the delay performance of π_{NOB} is about as good as we can hope for in heavy traffic, as is formalized in the next theorem.

Theorem 5.10 (Optimal Offline Policies). *Fix* $p \in (0,1)$.

1. *The policy* π_{NOB} *is feasible for all* $\lambda \in (1-p, 1)$, *and* [9]

$$C(p, \lambda, \pi_{NOB}) = \frac{1-p}{\lambda - (1-p)}. \tag{5.15}$$

2. *The policy* π_{NOB} *is asymptotically optimal in* Π_∞ *as* $\lambda \to 1$*:*

$$\lim_{\lambda \to 1} C(p, \lambda, \pi_{NOB}) = \lim_{\lambda \to 1} C^*_\infty(p, \lambda) = \frac{1-p}{p}.$$

Proof. See Section 5.6. □

Remark 5.11. *Heavy-traffic "Delay Collapse."* It is perhaps surprising to observe that the heavy-traffic scaling essentially collapses under π_{NOB}: the average queue length converges to a finite value, $\frac{1-p}{p}$, as $\lambda \to 1$, which is in sharp contrast with the optimal scaling of $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ for the online policies, given by Theorem 5.8 (see Figure 5-4 for an illustration of this difference). A "stack" interpretation of the No-Job-Left-Behind policy (Section 5.4.1) will help us understand intuitively why such a drastic discrepancy exists between the online and offline heavy-traffic scaling behaviors.

Also, as a by-product of Theorem 5.10, observe that the heavy-traffic limit scales,

[9] It is easy to see that π_{NOB} is not a very efficient diversion policy for relative small values of λ. In fact, $C(p, \lambda, \pi_{NOB})$ is a *decreasing* function of λ. This problem can be fixed by injecting into the arrival process a Poisson process of "dummy jobs" of rate $1 - \lambda - \epsilon$, so that the total rate of arrival is $1 - \epsilon$, where $\epsilon \approx 0$. This reasoning implies that $(1-p)/p$ is a uniform upper-bound for $C^*_\infty(p, \lambda)$, for all $\lambda \in (0,1)$.

with p, as

$$\lim_{\lambda \to 1} C^*_\infty(p,\lambda) \sim \frac{1}{p}, \quad \text{as } p \to 0. \tag{5.16}$$

This is consistent with an intuitive notion of "flexibility": delay should degenerate as the system's ability to divert away jobs diminishes.

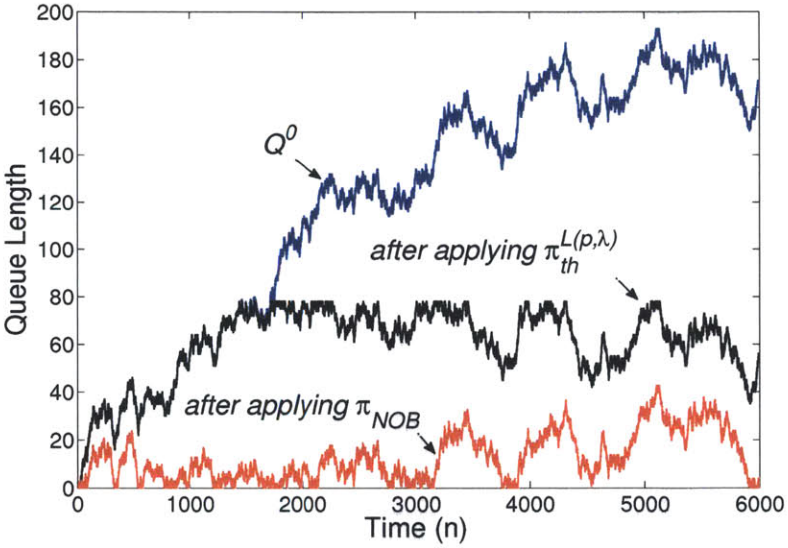

Figure 5-7: Example sample paths of Q^0 and those obtained after applying $\pi_{th}^{L(p,\lambda)}$ and π_{NOB} to Q^0, with $p = 0.05$ and $\lambda = 0.999$.

5.3.3 Policies with a Finite Lookahead Window

In practice, infinite prediction into the future is certainly too much to ask for. In this section, we show that a natural modification of π_{NOB} allows for the *same delay* to be achieved, using only a *finite* lookahead window, whose length, w_λ, increases to infinity as $\lambda \to 1$. [10]

[10] In a way, this is not entirely surprising, because the π_{NOB} policy leads to a diversion rate of $\lambda - (1-p)$, and there is an additional $p - [\lambda - (1-p)] = 1 - \lambda$ unused diversion rate that can be

Denote by $w \in \mathbb{R}_+$ the size of the lookahead window in continuous time, and $W(k) \in \mathbb{Z}_+$ the window size in the discrete-time embedded process Q^0, starting from slot k. Letting T_k be the time of the kth event in the system, then

$$W(k) = \max\{l \in \mathbb{Z}_+ : T_{k+l} \leq T_k + w\}. \tag{5.17}$$

For $x \in \mathbb{N}$, define the set of indices

$$U(Q,k,x) = \min\{j \in \{1,\dots,x\} : Q[k+j] = Q[k] - 1\}. \tag{5.18}$$

Definition 5.12. ***(w-No-Job-Left-Behind Policy)*** *Given an initial sample path Q^0 and $w > 0$, the w-No-Job-Left-Behind policy, denoted by π^w_{NOB}, diverts all arrivals in the set Ψ^w, where*

$$\Psi^w = \left\{k \in \Phi\left(Q^0\right) : U\left(Q^0, k, W(k)\right) = \infty\right\},$$

which corresponds to the set $\{j \in \{1,\dots,x\} : Q[k+j] = Q[k] - 1\}$ in Eq. (5.18) *being empty.*

It is easy to see that π^w_{NOB} is simply π_{NOB} applied within the confinement of a finite window: a job at t is diverted if and only if the initial queue length process does not return to below the current level *within the next w units of time*, assuming no further diversions are made. Since the window is finite, it is clear that $\Psi^w \supset \Psi$ for any $w < \infty$, and hence $C(p,\lambda,\pi^w_{NOB}) \leq C(p,\lambda,\pi_{NOB})$ for all $\lambda \in (1-p)$. The only issue now becomes that of feasibility: by making decisions only based on a finite lookahead window, we may end up deleting at a rate greater than p.

exploited.

The following theorem summarizes the above observations, and gives an upper bound on the appropriate window size, w, as a function of λ.[11]

Theorem 5.13. (Optimal Delay Scaling with Finite Lookahead) *Fix* $p \in (0,1)$. *There exists* $c_h > 0$, *such that if*

$$w_\lambda \geq c_h \ln \frac{1}{1-\lambda}, \quad \forall \lambda \in (1-p, 1), \tag{5.19}$$

then $\pi_{NOB}^{w_\lambda}$ *is feasible, and*

$$C\left(p, \lambda, \pi_{NOB}^{w_\lambda}\right) \leq C\left(p, \lambda, \pi_{NOB}\right) = \frac{1-p}{\lambda - (1-p)}, \tag{5.20}$$

Since $C_{w_\lambda}^*\left(p,\lambda\right) \geq C_\infty^*\left(p,\lambda\right)$ *and* $C_{w_\lambda}^*(p,\lambda) \leq C\left(p,\lambda,\pi_{NOB}^{w_\lambda}\right)$, *we also have that*

$$\lim_{\lambda \to 1} C_{w_\lambda}^*\left(p,\lambda\right) = \lim_{\lambda \to 1} C_\infty^*\left(p,\lambda\right) = \frac{1-p}{p}. \tag{5.21}$$

Proof. See Section 5.7.1. □

Theorem 6.1 sends a strong positive message, that is, a substantial delay improvement can still be harnessed even if the lookahead window is of finite length, as is the case in nearly all practical applications, as long as w scales faster than $c_h \ln \frac{1}{1-\lambda}$, as $\lambda \to 1$. On the other hand, in applications where future information is much more limited, can one still hope to leverage future information to achieve non-trivial performance gain over the online policies?

Unfortunately, the answer is, largely, "no." Our next result states that if the amount of future information is smaller even by a *constant factor*, then not only will the delay be infinite in the heavy-traffic regime, but the delay scaling will essentially

[11] Note that Theorem 5.13 implies Theorem 5.10 and is hence stronger.

be *no better* than that of an online policy. The proof of the theorem requires a fairly different set of tools than that employed to establish other results in this chapter, and will be given in Chapter 6.

Theorem 5.14 (Necessity of Future Information). *Fix* $p \in (0,1)$. *There exist* $c_l > 0$ *and* $\tilde{\lambda} \in (1-p,1)$, *so that if*

$$w_\lambda \leq c_l \ln \frac{1}{1-\lambda}, \quad \forall \lambda \in (\tilde{\lambda}, 1), \tag{5.22}$$

then

$$C^*_{w_\lambda}(p,\lambda) = \Theta\left(\ln \frac{1}{1-\lambda}\right), \quad \textit{as } \lambda \to 1. \tag{5.23}$$

Together, Theorem 5.13 and 5.14 suggest that the performance of the admission control problem *depends critically* on the amount of future information available, and in particular, on how the length of the lookahead window, w_λ, scales relative to the critical value of $\Theta\left(\ln \frac{1}{1-\lambda}\right)$.

Figure 5-8 provides a graphical summary of Theorems 5.8 through 5.14. Note that the constants in the scaling are not being differentiated.

Remarks on the Information Lower Bound. There are several interesting implications of Theorem 5.14. First, by virtue of being a lower bound for the case where the decision maker is given the *exact* realization of the future input over the lookahead horizon, Theorem 5.14 automatically extends to settings where predictions can be *noisy* or *corrupted*, as is typically the case in practical applications.

From an operational point of view, although Theorem 5.14 invalidates the usefulness of future information in certain regimes, it is nevertheless reassuring to know that a simple online policy could do almost as well as any sophisticated prediction-guided policy, even when the amount of predictive information available grows with the traffic intensity. Moreover, the theorem does not rule out the possibility of meaningful prediction-guided policies when future information is limited; it only implies that our search for such scenarios should aim at more moderate, *constant factor* performance improvements over online policies. In fact, numerical results in [93] on a similar admission control model suggest that sizable performance gains can still be achieved, even with limited and noisy predictive information.

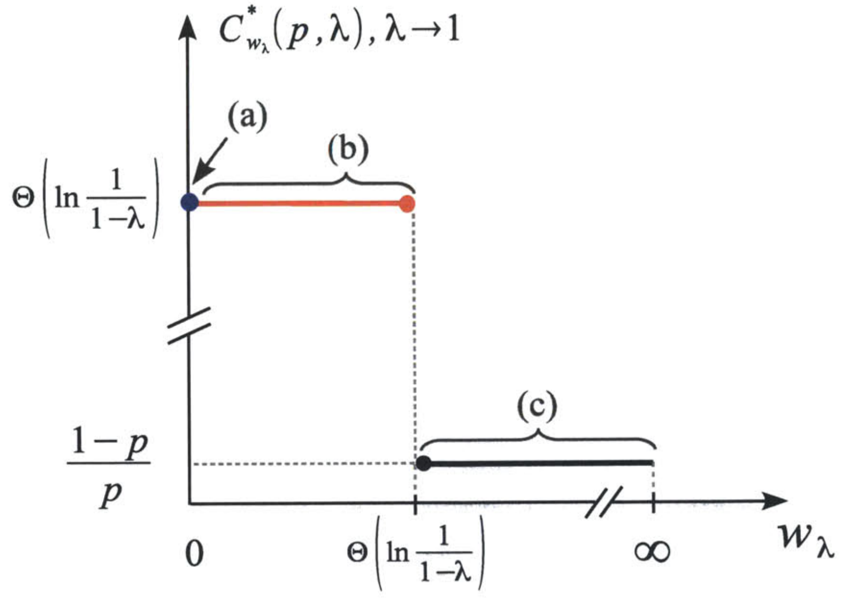

Figure 5-8: Optimal delay scaling in the heavy-traffic regime, as a function of the length of the lookahead window, w_λ. The blue (a), red (b), and black (c) segments correspond to the regimes established by Theorems 5.8, 5.14, and 5.13, respectively.

Delay-Information Duality Our results imply an interesting conservation law, or dual relationship, between delay and future information: from Eqs. (5.19) and (5.23), we see that the *sum* of the delay ($C^*_{w_\lambda}(p,\lambda)$) and information ($w_\lambda$) must be of order $\Omega\left(\ln \frac{1}{1-\lambda}\right)$, as $\lambda \to 1$. Put in another way, future information that is of order $\Theta\left(\log \frac{1}{1-\lambda}\right)$ is sufficient to achieve a finite delay limit, and one has to suffer $\Theta\left(\log \frac{1}{1-\lambda}\right)$ in delay, if there is only (just a bit) less future information (Figure 5-8).

Even though such conservation seems to suggest that there is no "free lunch" to be had, the ability to understand and make such trade-offs can still be useful, because depending on the application, future information may be significantly less costly than delay, or *vice versa*.

5.4 Interpretations of π_{NOB}

We present two equivalent ways of describing the No-Job-Left-Behind policy π_{NOB}. The *stack interpretation* helps us derive the asymptotic diversion rate of π_{NOB} in a simple manner, and illustrates the superiority of π_{NOB} over an online policy. Another description of π_{NOB} using time-reversal shows us that the set of diversions made by π_{NOB} can be calculated efficiently in linear time (with respect to the length of the time horizon).

5.4.1 Stack Interpretation

Suppose that the service discipline adopted by the server is that of last-in-first-out (LIFO), where it always fetches a task that has arrived the latest. In other words, the queue works as a *stack*. Suppose that we first simulate the stack without any diversion. It is easy to see that when the arrival rate λ is greater than the service rate $1-p$, there will be a growing set of jobs at the bottom of the stack that will *never* be processed. Label all such jobs as "left-behind." For example, Figure 5-5 shows the evolution of the queue over time, where all "left-behind" jobs are colored with a blue shade. One can then verify that the policy π_{NOB} given in Definition 5.9 is equivalent to deleting all jobs that are labeled "left-behind," hence the term "No-Job-Left-Behind." Figure 5-6 illustrates applying π_{NOB} to a sample path of Q^0, where the ith job to be diverted is precisely the ith job among all jobs that would have never been processed by the server under a LIFO policy.

One advantage of the stack interpretation is that it makes obvious the fact that the diversion rate induced by π_{NOB} is equal to $\lambda-(1-p)<p$, as stated in the following lemma.

Lemma 5.15. *For all $\lambda > 1-p$, the following statements hold.*

1. *With probability one, there exists $T < \infty$, such that every service token generated after time T is matched with some job. In other words, the server never idles after some finite time.*

2. *Let $Q = D(Q^0, M^{\Psi})$. We have*

$$\limsup_{k\to\infty} \frac{1}{k} I\left(M^{\Psi}, k\right) \le \frac{\lambda-(1-p)}{\lambda+1-p}, \quad a.s., \tag{5.24}$$

   which implies that π_{NOB} is feasible for all $p \in (0,1)$ and $\lambda \in (1-p, 1)$.

Proof. See Appendix B.1.1 □

"Anticipation" vs. "Reaction"

Some geometric intuition from the stack interpretation shows that the power of π_{NOB} essentially stems from being highly *anticipatory*. Looking at Figure 5-5, one sees that the jobs that are "left behind" at the bottom of the stack correspond to those who arrive during the intervals where the initial sample path Q^0 is taking a consecutive "upward hike." In other words, π_{NOB} begins to divert jobs when it anticipates that the arrivals are *just about to* get intense. Similarly, a job in the stack will be "served" if Q^0 curves down eventually in the future, which corresponds π_{NOB}'s ceasing to divert jobs as soon as it anticipates that the next few arrivals can be handled by the server alone. In sharp contrast is the nature of the optimal online policy, $\pi_{th}^{L(p,\lambda)}$, which is by definition "reactive" and begins to divert only when the current queue length has already reached a high level. The differences in the resulting sample paths are illustrated via simulations in Figure 5-7. For example, as Q^0 continues to increase during the first 1000 time slots, π_{NOB} begins deleting immediately after $t = 0$, while no diversion is made by $\pi_{th}^{L(p,\lambda)}$ during this period.

As a rough analogy, the offline policy starts to divert *before* the arrivals get busy, but the online policy can only divert *after* the burst in arrival traffic has been realized, by which point it is already "too late" to fully contain the delay. This explains, to a certain extent, why π_{NOB} is capable of achieving "delay collapse" in the heavy-traffic regime (i.e., a finite limit of delay as $\lambda \to 1$, Theorem 5.10), while the delay under even the best online policy diverges to infinity as $\lambda \to 1$ (Theorem 5.8).

5.4.2 A Linear-time Algorithm for π_{NOB}

While the offline diversion problem serves as a nice abstraction, it is impossible to actually store information about the *infinite* future in practice, even if such information is available. A natural finite-horizon version of the offline diversion problem can be posed as follows: given the values of Q^0 over the first N slots, where N finite, one would like to compute the set of diversions made by π_{NOB}:

$$M_N^{\Psi} = M^{\Psi} \cap \{1, \ldots, N\},$$

assuming that $Q^0[k] > Q^0[N]$ for all $k \geq N$. Note that this problem also arises in computing the sites of diversions for the π^w_{NOB} policy, where one would replace N with the length of the lookahead window, w.

We have the following algorithm, which identifies all slots on which a new "minimum" (denoted by the variable S) is achieved in Q^0, when viewed in the *reverse* order of time.

A Linear-time Algorithm for π_{NOB}

```
$S \leftarrow Q^0[N]$, and $M^\Psi_N \leftarrow \varnothing$
for $k = N$ down to 1 do
  if $Q^0[k] < S$ then
    $M^\Psi_N \leftarrow M^\Psi_N \cup \{k+1\}$
    $S \leftarrow Q^0[k]$
  else
    $M^\Psi_N \leftarrow M^\Psi_N$
  end if
end for
return $M^\Psi_N$
```

It is easy to see that the running time of the above algorithm scales linearly with the length of the time horizon, N. Note that this is not the only possible linear-time algorithm. In fact, one can verify that the simulation procedure used in describing the stack interpretation of π_{NOB} (Section 5.4), which keeps track of which jobs would eventually be served, is itself a linear-time algorithm. However, the time-reversed version given here is arguably more intuitive and simpler to describe.

5.5 Optimal Online Policies

Starting from this section and through Section 5.7, we present the proofs of the results stated in Section 5.3.

We begin with showing Theorem 5.8, by formulating the online problem as a Markov decision problem (MDP) with an average cost constraint, which then enables us to use existing results to characterize the form of optimal policies. Once the family of threshold policies has been shown to achieve the optimal delay scaling in Π_0 under heavy-traffic, the exact form of the scaling can be obtained in a fairly straightforward manner from the steady-state distribution of a truncated birth-death process.

5.5.1 A Markov Decision Problem Formulation

Since both the arrival and service processes are Poisson, we can formulate the problem of finding an optimal policy in Π_0 as a continuous-time Markov decision problem with an average-cost constraint, as follows. Let $\{Q(t) : t \in \mathbb{R}_+\}$ be the resulting continuous-time queue length process after applying some policy in Π_0 to Q^0. Let T_l be the lth upward jump in Q and τ_l the length of the lth inter-jump interval, $\tau_l = T_l - T_{l-1}$. The task of a diversion policy, $\pi \in \Pi_0$, amounts to choosing, for each inter-jump interval, a *diversion action*, $a_l \in [0,1]$, where the value of a_l corresponds to the probability that the next arrival during the current inter-jump interval will be diverted. Define R and K to be the *reward* and *cost* functions of an inter-jump interval, respectively,

$$R(Q_l, a_l, \tau_l) = -Q_l \cdot \tau_l, \tag{5.25}$$

$$K(Q_l, a_l, \tau_l) = \lambda(1 - a_l)\tau_l, \tag{5.26}$$

where $Q_l = Q(T_l)$. The corresponding MDP seeks to maximize the time-average reward[12]

$$\bar{R}_\pi = \liminf_{k\to\infty} \frac{\mathbb{E}_\pi\left(\sum_{l=1}^k R(Q_l, a_l, \tau_l)\right)}{\mathbb{E}_\pi\left(\sum_{l=1}^k \tau_l\right)} \tag{5.27}$$

while obeying the average-cost constraint

$$\bar{C}_\pi = \limsup_{k\to\infty} \frac{\mathbb{E}_\pi\left(\sum_{l=1}^k K(Q_l, a_l, \tau_l)\right)}{\mathbb{E}_\pi\left(\sum_{l=1}^k \tau_l\right)} \le p. \tag{5.28}$$

To see why this MDP solves our diversion problem, observe that $\bar{R}_\pi$ is the negative of the time-average queue length, and $\bar{C}_\pi$ is the time-average diversion rate.

It is well known that the type of constrained MDP described above admits an optimal policy that is stationary [5], which means that the action a_l depends solely on current state, Q_l, and is independent of the time index l. Therefore, it suffices to describe π using a sequence, $\{b_q : q \in \mathbb{Z}_+\}$, such that $a_l = b_q$ whenever $Q_l = q$. Moreover, when the state space is finite[13], stronger characterizations of the b_q's have been obtained for a family of reward and cost functions under certain regularity assumptions (Hypotheses 2.7, 3.1 and 4.1 in [12]), which are satisfied in our model (Eqs. (5.25) and (5.26)). Theorem 5.8 will be proved using the next known result (adapted from Theorem 4.4 in [12]):

Lemma 5.16. *Fix p and λ, and let the buffer size B be finite. There exists an*

[12] It is possible to show that in the online setting, the average cost and reward defined here are interchangable with those in Eqs. (5.10) and (5.5), respectively.

[13] This corresponds to a finite buffer size in our problem, where one can assume that the next arrival is automatically diverted when the buffer is full, independent of the value of a_l.

optimal stationary policy, $\{b_q^*\}$*, of the form*

$$b_q^* = \begin{cases} 1, & q < L^* - 1, \\ \xi, & q = L^* - 1, \\ 0, & q \geq L^*, \end{cases}$$

for some $L^* \in \mathbb{Z}_+$ *and* $\xi \in [0,1]$.

5.5.2 Proof of Theorem 5.8

Proof. **(Theorem 5.8)** In words, Lemma 5.16 states that the optimal policy admits a "quasi-threshold" form: it diverts the next arrival when $Q(t) \geq L^*$, admits when $Q(t) < L^* - 1$, and admits with probability ξ when $Q(t) = L^* - 1$. Suppose, for the moment, that the statements of Lemma 5.16 also hold when the buffer size is infinite, an assumption to be justified by the end of the proof. Denoting by π_p^* the stationary optimal policy associated with $\{b_q^*\}$, when the constraint on the average diversion rate is p (Eq. (5.28)). The evolution of $Q(t)$ under π_p^* is that of a birth-death process truncated at state L^*, with the transition rates given in Figure 5-9, and the time-average queue length is equal to the expected queue length in steady state. Using standard calculations involving the steady-state distribution of the induced Markov process, it is not difficult to verify that

$$C(p, \lambda, \pi_{th}^{L^*-1}) \leq C(p, \lambda, \pi_p^*) \leq C(p, \lambda, \pi_{th}^{L^*}), \tag{5.29}$$

where L^* is defined as in Lemma 5.16, and $C(p, \lambda, \pi)$ is the time-average queue length under policy π, defined in Eq. (5.10).

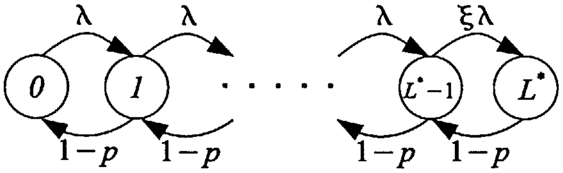

Figure 5-9: The truncated birth-death process induced by π_p^*.

Denote by $\{\mu_i^L : i \in \mathbb{N}\}$ the steady-state probability of the queue length being equal to i, under a threshold policy π_{th}^L. Assuming $\lambda \neq 1-p$, standard calculations using the balance equations yield

$$\mu_i^L = \left(\frac{\lambda}{1-p}\right)^i \cdot \left(\frac{1-\frac{\lambda}{1-p}}{1-\left(\frac{\lambda}{1-p}\right)^{L+1}}\right), \quad \forall 1 \leq i \leq L, \tag{5.30}$$

and $\mu_i^L = 0$ for all $i \geq L+1$. The time-average queue length is given by

$$\begin{aligned} C(p, \lambda, \pi_{th}^L) &= \sum_{i=1}^{L} i \cdot \mu_i^L \\ &= \frac{\theta}{(\theta-1)(\theta^{L+1}-1)} \cdot \left[1-\theta^L + L\theta^L(\theta-1)\right], \end{aligned} \tag{5.31}$$

where $\theta = \frac{\lambda}{1-p}$. Note that when $\lambda > 1-p$, μ_i^L is decreasing with respect to L for all $i \in \{0, 1, \ldots, L\}$ (Eq. (5.30)), which implies that the time-average queue length is monotonically increasing in L, i.e.,

$$\begin{aligned} C(p,\lambda,\pi_{th}^{L+1}) - C(p,\lambda,\pi_{th}^{L}) =& (L+1)\cdot\mu_{L+1}^{L+1} + \sum_{i=0}^{L} i\cdot(\mu_i^{L+1} - \mu_i^{L}) \\ \geq& (L+1)\cdot\mu_{L+1}^{L+1} + L\cdot\left(\sum_{i=0}^{L}\mu_i^{L+1} - \mu_i^{L}\right) \\ =& (L+1)\cdot\mu_{L+1}^{L+1} + L\cdot\left(1 - \mu_i^{L+1} - 1\right) \\ =& \mu_{L+1}^{L+1} \\ >& 0. \end{aligned} \tag{5.32}$$

It is also easy to see that, fixing p, since we have that $\theta > 1+\delta$ for all λ sufficiently close to 1, where $\delta > 0$ is a fixed constant, we have

$$C(p,\lambda,\pi_{th}^{L}) = \left(\frac{\theta^{L+1}}{\theta^{L+1}-1}\right)L - \frac{\theta}{\theta-1}\cdot\frac{\theta^{L}-1}{\theta^{L+1}-1} \sim L, \quad \text{as } L\to\infty. \tag{5.33}$$

Since diversions only occur when $Q(t)$ is in state L, from Eq. (5.30), the average rate of diversions in continuous time under π_{th}^{L} is given by,

$$r_d\left(p,\lambda,\pi_{th}^{L},\right) = \lambda\cdot\pi_L = \lambda\cdot\left(\frac{\lambda}{1-p}\right)^{L}\cdot\left(\frac{1-\frac{\lambda}{1-p}}{1-\left(\frac{\lambda}{1-p}\right)^{L+1}}\right). \tag{5.34}$$

Define

$$L(x,\lambda) = \min\left\{L\in\mathbb{Z}_{+} : r_d\left(p,\lambda,\pi_{th}^{L},\right)\leq x\right\}, \tag{5.35}$$

that is, $L(x,\lambda)$ is the smallest L for which π_{th}^{L} remains feasible, given a diversion rate constraint of x. Using Eqs. (5.34) and (5.35) to solve for $L(p,\lambda)$, we obtain, after some algebra,

$$L(p,\lambda) = \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1, \tag{5.36}$$

and, by combining Eq. (5.36) and Eq. (5.33) with $L = L(p,\lambda)$, we have

$$C(p,\lambda,\pi_{th}^{L(p,\lambda)}) \sim L(p,\lambda) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{5.37}$$

By Eqs. (5.32) and (5.35), we know that $\pi_{th}^{L(p,\lambda)}$ achieves the minimum average queue length among all feasible threshold policies. By Eq. (5.29), we must have that

$$C\left(p,\lambda,\pi_{th}^{L(p,\lambda)-1}\right) \leq C(p,\lambda,\pi_p^*) \leq C\left(p,\lambda,\pi_{th}^{L(p,\lambda)}\right), \tag{5.38}$$

Since Lemma 5.16 only applies when $B < \infty$, Eq. (5.38) holds whenever the buffer size, B, is greater than $L(p,\lambda)$ but finite. We next extend Eq. (5.38) to the case of $B = \infty$. Denote by ν_p^* a stationary optimal policy, when $B = \infty$ and the constraint on average diversion rate is equal to p (Eq. (5.28)). The upper bound on $C(p,\lambda,\pi_p^*)$ in Eq. (5.38) automatically holds for $C(p,\lambda,\nu_p^*)$, since $C(p,\lambda,\pi_{th}^{L(p,\lambda)})$ is still feasible when $B = \infty$. It remains to show a lower bound of the form

$$C(p,\lambda,\nu_p^*) \geq C\left(p,\lambda,\pi_{th}^{L(p,\lambda)-2}\right) \tag{5.39}$$

when $B = \infty$, which, together with the upper bound, will have implied that the scaling of $C(p,\lambda,\pi_{th}^{L(p,\lambda)})$ (Eq. (5.37)) carries over to ν_p^*,

$$C\left(p,\lambda,\nu_p^*\right) \sim C(p,\lambda,\pi_{th}^{L(p,\lambda)}) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1, \tag{5.40}$$

thus proving Theorem 5.8.

To show Eq. (5.39), we will use a straightforward truncation argument that relates the performance of an optimal policy under $B = \infty$ to the case of $B < \infty$. Denote by $\{b_q^*\}$ the diversion probabilities of a stationary optimal policy, ν_p^*, and by $\{b_q^*(B')\}$ the diversion probabilities for a truncated version, $\nu_p^*(B')$, with

$$b_q^*(B') = \mathbb{I}\left(q \leq B'\right) \cdot b_q^*,$$

for all $q \geq 0$. Since ν_p^* is optimal and yields the minimum average queue length, it is without loss of generality to assume that the Markov process for $Q(t)$ induced by ν_p^* is positive recurrent. Denoting by $\{\mu_i^*\}$ and $\{\mu_i^*(B')\}$ the steady-state probability of queue length being equal to i under ν_p^* and $\nu_p^*(B')$, respectively, it follows from the positive recurrence of $Q(t)$ under ν_p and some algebra, that

$$\lim_{B' \to \infty} \mu_i^*(B') = \mu_i^*, \tag{5.41}$$

for all $i \in \mathbb{Z}_+$, and

$$\lim_{B' \to \infty} C\left(p, \lambda, \nu_p^*(B')\right) = C\left(p, \lambda, \nu_p^*\right). \tag{5.42}$$

By Eq.(5.41) and the fact that $b_i^*(B') = b_i^*$ for all $0 \leq i \leq B'$, we have that[14]

$$\lim_{B' \to \infty} r_d\left(p, \lambda, \nu_p^*(B')\right) = \lim_{B' \to \infty} \lambda \sum_{i=0}^{\infty} \mu_i^*(B') \cdot (1 - b_i^*(B')) = r_d\left(p, \lambda, \nu_p^*\right) \leq p. \tag{5.43}$$

It is not difficult to verify, from the definition of $L(p, \lambda)$ (Eq. (5.35)), that

$$\lim_{\delta \to 0} L(p + \delta, \lambda) \geq L(p, \lambda) - 1,$$

[14] Note that in general, $r_d\left(p, \lambda, \nu_p^*(B')\right)$ could be greater than p, for any finite B'.

for all p, λ. For all $\delta > 0$, choose B' to be sufficiently large, so that

$$C\left(p, \lambda, \nu_p^*(B')\right) \le C\left(p, \lambda, \nu_p^*\right) + \delta, \tag{5.44}$$

$$L\left(\lambda, r_d\left(p, \lambda, \nu_p^*(B')\right)\right) \ge L(p, \lambda) - 1, \tag{5.45}$$

Let $p' = r_d\left(p, \lambda, \nu_p^*(B')\right)$. Since $b_i^*(B') = 0$ for all $i \ge B' + 1$, by Eq. (5.45) we have

$$C\left(p, \lambda, \nu_p^*(B')\right) \ge C\left(p, \lambda, \pi_{p'}^*\right), \tag{5.46}$$

where π_p^* is the optimal stationary policy given in Lemma 5.16 under any the finite buffer size $B > B'$. We have

$$\begin{aligned} C\left(p, \lambda, \nu_p^*\right) + \delta &\overset{(a)}{\ge} C\left(p, \lambda, \nu_p^*(B')\right) \\ &\overset{(b)}{\ge} C\left(p, \lambda, \pi_{p'}^*\right) \\ &\overset{(c)}{\ge} C\left(p, \lambda, \pi_{th}^{L(p',\lambda)-1}\right) \\ &\overset{(d)}{\ge} C\left(p, \lambda, \pi_{th}^{L(p,\lambda)-2}\right), \end{aligned} \tag{5.47}$$

where the inequalities (a) through (d) follow from Eqs. (5.44), (5.46), (5.38), and (5.45), respectively. Since Eq. (5.47) holds for all $\delta > 0$, we have proven Eq. (5.39). This completes the proof of Theorem 5.8. □

5.6 Optimal Offline Policies

We prove Theorem 5.10 in this section, which is completed in two parts. In the first part (Section 5.6.2), we give a full characterization of the sample path that results under the policy π_{NOB} (Proposition 5.18), which turns out to be a *positive recurrent* random walk. This allows us to obtain the steady-state distribution of the queue length under π_{NOB} in closed-form. From this, the expected queue length, which is equal to the time-average queue length, $C(p, \lambda, \pi_{NOB})$, can be easily derived and is shown to be $\frac{1-p}{\lambda-(1-p)}$. Several side results we obtain along this path will also be used in subsequent sections.

The second part of the proof (Section 5.6.3) focuses on showing the heavy-traffic optimality of π_{NOB} among the class of all feasible offline policies, namely, that $\lim_{\lambda\to 1} C(p, \lambda, \pi_{NOB}) = \lim_{\lambda\to 1} C^*_\infty(p, \lambda)$, which, together with the first part, proves Theorem 5.10 (Section 5.6.4). The optimality result is proved using a sample-path-based analysis, by relating the resulting queue length sample path of π_{NOB} to that of a greedy diversion rule, which has an optimal diversion performance over a *finite* time horizon, $\{1, \ldots, N\}$, given any initial sample path. We then show that the discrepancy between π_{NOB} and the greedy policy, in terms of the resulting time-average queue length after diversion, diminishes almost surely as $N \to \infty$ and $\lambda \to 1$ (with the two limits taken in this order). This establishes the heavy-traffic optimality of π_{NOB}.

5.6.1 Additional Notation

Define $\tilde{Q}$ as the resulting queue length process after applying π_{NOB}

$$\tilde{Q} = D\left(Q^0, M^\Psi\right).$$

and Q as the shifted version of $\tilde{Q}$, so that Q starts from the first diversion in $\tilde{Q}$,

$$Q[k] = \tilde{Q}[k + m_1^\Psi], \quad k \in \mathbb{Z}_+. \tag{5.48}$$

We say that $B = \{d, \ldots, u\} \subset \mathbb{N}$ is a **busy period** of Q, if

$$Q[d-1] = Q[u] = 0, \text{ and } Q[k] > 0 \text{ for all } k \in \{d, \ldots, u-1\}. \tag{5.49}$$

We may write $B_j = \{d_j, \ldots, u_j\}$ to mean the jth busy period of Q. An example of a busy period is illustrated in Figure 5-6.

Finally, we will refer to the set of slots between two adjacent diversions in Q (note the offset of m_1),

$$E_i = \left\{m_i^{\Psi} - m_1^{\Psi}, m_i^{\Psi} + 1 - m_1^{\Psi}, \ldots, m_{i+1}^{\Psi} - 1 - m_1^{\Psi}\right\}, \tag{5.50}$$

as the ith **diversion epoch**.

5.6.2 Performance of the No-Job-Left-Behind Policy

For simplicity of notation, throughout this section, we will denote by $M = \{m_i : i \in \mathbb{N}\}$ the diversion sequence generated by applying π_{NOB} to Q^0, when there is no ambiguity (as opposed to using M^{Ψ} and m_i^{Ψ}). The following lemma summarizes some important properties of Q which will be used repeatedly.

Lemma 5.17. *Suppose $1 > \lambda > 1 - p > 0$. The following hold with probability one.*

1. *For all $k \in \mathbb{N}$, we have $Q[k] = Q^0[k + m_1] - I(M, k + m_1)$.*

2. *Fix some $k \in \mathbb{N}$. We have $k = m_i - m_1$ for some i, if and only if*

$$Q[k] = Q[k-1] = 0, \tag{5.51}$$

with the convention that $Q[-1] = 0$. In other words, the appearance of two consecutive zeros in Q is equivalent to having a diversion on the second zero.

3. $Q[k] \in \mathbb{Z}_+$ *for all* $k \in \mathbb{Z}_+$.

Proof. See Appendix B.1.2 □

The next proposition is the main result of this subsection. It specifies the probability law that governs the evolution of Q.

Proposition 5.18. $\{Q[k] : k \in \mathbb{Z}_+\}$ *is a random walk on* $\mathbb{Z}_+$*, with* $Q[0] = 0$*, and, for all* $k \in \mathbb{N}$ *and* $x_1, x_2 \in \mathbb{Z}_+$*,*

$$\mathbb{P}\left(Q[k+1] = x_2 \mid Q[k] = x_2\right) = \begin{cases} \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$*, and*

$$\mathbb{P}\left(Q[k+1] = x_2 \mid Q[k] = x_1\right) = \begin{cases} \frac{1-p}{\lambda+1-p}, & x_2 - x_1 = 1, \\ \frac{\lambda}{\lambda+1-p}, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$.

Proof. For a sequence $\{X[k] : k \in \mathbb{N}\}$ and $s, t \in \mathbb{N}$, $s \leq t$, we will use the short-hand

$$X_s^t = \{X[s], \ldots, X[t]\}.$$

Fix $k \in N$, and a sequence $(q_1, \dots, q_k) \subset \mathbb{Z}_+{}^k$. We have

$$\begin{aligned}
&\mathbb{P}\left(Q[k] = q[k] \,\middle|\, Q_1^{k-1} = q_1^{k-1}\right) \\
=&\sum_{l=1}^{k} \sum_{\substack{t_1,\dots,t_l,\\ t_l \le k-1+t_1}} \mathbb{P}\left(Q[k] = q[k] \,\middle|\, Q_1^{k-1} = q_1^{k-1}, m_1^l = t_1^l, m_{l+1} \ge k + t_1\right) \\
&\cdot \mathbb{P}\left(m_1^l = t_1^l, m_{l+1} \ge k + t_1 \,\middle|\, Q_1^{k-1} = q_1^{k-1}\right) \qquad (5.52)
\end{aligned}$$

Restricting to the values of t_is and $q[i]$s for which the summand is non-zero, the first factor in the summand can be written as

$$\begin{aligned}
&\mathbb{P}\left(Q[k] = q[k] \,\middle|\, Q_1^{k-1} = q_1^{k-1}, m_1^l = t_1^l, m_{l+1} \ge k + t_1\right) \\
=&\mathbb{P}\left(\tilde{Q}[k + m_1] = q[k] \,\middle|\, \tilde{Q}_{m_1+1}^{m_1+k-1} = q_1^{k-1}, m_1^l = t_1^l, m_{l+1} \ge k + t_1\right) \\
\overset{(a)}{=}&\mathbb{P}\Big(Q^0[k + t_1] = q[k] + l \,\Big|\, Q^0[s + t_1] = q[s] + I\left(\{t_i\}_{i=1}^l, s + t_1\right), \forall 1 \le s \le k - 1, \\
&\qquad\qquad \text{and } \min_{r \ge k+t_1} Q^0[r] \ge l\Big) \\
\overset{(b)}{=}&\mathbb{P}\left(Q^0[k + t_1] = q[k] + l \,\middle|\, Q^0[k - 1 + t_1] = q[k-1] + l, \text{ and } \min_{r \ge k+t_1} Q^0[r] \ge l\right), \qquad (5.53)
\end{aligned}$$

where $\tilde{Q}$ was defined in Eq. (5.6.1). Step (a) follows from Lemma 5.17 and the fact that $t_l \le k - 1 + t_1$, and (b) from the Markov property of Q^0 and the fact that the events $\{\min_{r \ge k+t_1} Q^0[r] \ge l\}$, $\{Q^0[k + t_1] = q[k] + l\}$, and their intersection, depend only on the values of $\{Q^0[s] : s \ge k + t_1\}$, and are hence independent of $\{Q^0[s] : 1 \le s \le k - 2 + t_1\}$ conditional on the value of $Q^0[t_1 + k - 1]$.

Since the process Q lives in $\mathbb{Z}_+$ (Lemma 5.17), it suffices to consider the case of

$q[k] = q[k-1] + 1$, and show that

$$\begin{aligned}&\mathbb{P}\Big(Q^0[k+t_1] = q[k-1] + 1 + l \,\Big|\, Q^0[k-1+t_1] = q[k-1] + l,\\ &\qquad\qquad\qquad\qquad \text{and } \min_{r \ge k+t_1} Q^0[r] \ge l\Big)\\ &= \frac{1-p}{\lambda + 1 - p},\end{aligned} \tag{5.54}$$

for all $q[k-1] \in \mathbb{Z}_+$. Since $Q[m_i - m_1] = Q[m_i - 1 - m_1] = 0$ for all i (Lemma 5.17), the fact that $q[k] = q[k-1] + 1 > 0$ implies that

$$k < m_{l+1} - 1 + m_1. \tag{5.55}$$

Moreover, since $Q^0[m_{l+1} - 1] = l$ and $k < m_{l+1} - 1 + m_1$, we have that

$$q[k] > 0 \text{ implies } Q^0[t] = l, \text{ for some } t \ge k + 1 + m_1. \tag{5.56}$$

We consider two cases, depending on the value of $q[k-1]$.

Case 1: $q[k-1] > 0$. Using the same argument that led to Eq. (5.56), we have that

$$q[k-1] > 0 \text{ implies } Q^0[t] = l, \text{ for some } t \ge k + m_1. \tag{5.57}$$

It is important to note that, despite the similarity in conclusions, Eqs. (5.56) and

(5.57) are different in their assumptions (i.e., $q[k]$ versus $q[k-1]$). We have

$$\begin{aligned}
&\mathbb{P}\left(Q^0[k+t_1] = q[k-1]+1+l \,\middle|\, Q^0[k-1+t_1] = q[k-1]+l,\right.\\
&\qquad\qquad\qquad\qquad\qquad\qquad \left.\text{and } \min_{r\geq k+t_1} Q^0[r] \geq l\right)\\
&\overset{(a)}{=}\mathbb{P}\left(Q^0[k+t_1] = q[k-1]+1+l \,\middle|\, Q^0[k-1+t_1] = q[k-1]+l,\right.\\
&\qquad\qquad\qquad\qquad\qquad\qquad \left.\text{and } \min_{r\geq k+t_1} Q^0[r] = l\right)\\
&\overset{(b)}{=}\mathbb{P}\left(Q^0[2] = q[k-1]+1 \,\middle|\, Q^0[1] = q[k-1], \text{ and } \min_{r\geq 2} Q^0[r] = 0\right)\\
&\overset{(c)}{=}\frac{1-p}{\lambda+1-p},
\end{aligned} \tag{5.58}$$

where (a) follows from Eq. (5.57), (b) from the stationary and space-homogeneity of the Markov chain Q^0, and (c) from the following well-known property of a transient random walk conditional to returning to zero.

Lemma 5.19. *Let $\{X[k] : k \in \mathbb{N}\}$ be a random walk on $\mathbb{Z}_+$, such that for all $x_1, x_2 \in \mathbb{Z}_+$ and $k \in \mathbb{N}$,*

$$\mathbb{P}(X[k+1] = x_2 \mid X[k] = x_2) = \begin{cases} q, & x_2 - x_1 = 1, \\ 1-q, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$, and

$$\mathbb{P}(X[k+1] = x_2 \mid X[k] = x_1) = \begin{cases} q, & x_2 - x_1 = 1, \\ 1-q, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$*, where* $q \in \left(\frac{1}{2}, 1\right)$*. Then for all* $x_1, x_2 \in \mathbb{Z}_+$ *and* $k \in \mathbb{N}$*,*

$$\mathbb{P}\left(X[k+1] = x_2 \middle| X[k] = x_1, \min_{r \geq k+1} X[r] = 0\right) = \begin{cases} 1-q, & x_2 - x_1 = 1, \\ q, & x_2 - x_1 = -1, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 > 0$*, and*

$$\mathbb{P}\left(X[k+1] = x_2 \middle| X[k] = x_1, \min_{r \geq k+1} X[r] = 0\right) = \begin{cases} 1-q, & x_2 - x_1 = 1, \\ q, & x_2 - x_1 = 0, \\ 0, & \textit{otherwise}, \end{cases}$$

if $x_1 = 0$*. In other words, conditional on the eventual return to* 0 *and until that happens, a transient random walk obeys the same probability law as a random walk with the reversed one-step transition probabilities.*

Proof. See Appendix B.1.3. □

Case 2: $q[k-1]=0$. We have

$$\begin{aligned}
&\mathbb{P}\Big(Q^0[k+t_1]=q[k-1]+1+l\,\Big|\,Q^0[k-1+t_1]=q[k-1]+l,\\
&\qquad\qquad\qquad\qquad\text{and }\min_{r\geq k+t_1}Q^0[r]\geq l\Big)\\
&\overset{(a)}{=}\mathbb{P}\Big(Q^0[k+t_1]=1+l,\text{ and }\min_{r>k+t_1}Q^0[r]=l\,\Big|\,Q^0[k-1+t_1]=l,\\
&\qquad\qquad\qquad\qquad\text{and }\min_{r\geq k+t_1}Q^0[r]\geq l\Big)\\
&\overset{(b)}{=}\mathbb{P}\Big(Q^0[2]=2,\text{ and }\min_{r>2}Q^0[r]=1\,\Big|\,Q^0[1]=1,\text{ and }\min_{r\geq 2}Q^0[r]\geq 1\Big),\\
&\overset{\Delta}{=}x,
\end{aligned}\tag{5.59}$$

where (a) follows from Eq. (5.56) (note its difference with Eq. (5.57)), and (b) from the stationarity and space-homogeneity of Q^0, and the assumption that $l\geq 1$ (Eq. (5.52)).

Since Eqs. (5.58) and (5.59) hold for all $x_1,l\in\mathbb{Z}_+$ and $k\geq m_1+1$, by Eq. (5.52), we have that

$$\mathbb{P}\left(Q[k]=q[k]\,\middle|\,Q_1^{k-1}=q_1^{k-1}\right)=\begin{cases}\frac{1-p}{\lambda+1-p}, & q[k]-q[k-1]=1,\\ \frac{\lambda}{\lambda+1-p}, & q[k]-q[k-1]=-1,\\ 0, & \text{otherwise,}\end{cases}\tag{5.60}$$

if $q[k-1]>0$, and

$$\mathbb{P}\left(Q[k]=q[k]\,\middle|\,Q_1^{k-1}=q_1^{k-1}\right)=\begin{cases}x, & q[k]-q[k-1]=1,\\ 1-x, & q[k]-q[k-1]=0,\\ 0, & \text{otherwise,}\end{cases}\tag{5.61}$$

if $q[k-1]=0$, where x represents the value of the probability in Eq. (5.59). Clearly, $Q[0]=Q^0[m_1]=0$. We next show that x is indeed equal to $\frac{1-p}{\lambda+1-p}$, which will have proven Proposition 5.18.

One can in principle obtain the value of x by directly computing the probability in line (b) of Eq. (5.59), which can be quite difficult to do. Instead, we will use an indirect approach that turns out to be computationally much simpler: we will relate x to the rate of diversion of π_{NOB} using renewal theory, and then solve for x. As a by-product of this approach, we will also get a better understanding of an important regenerative structure of π_{NOB} (Eq. (5.67)), which will be useful for the analysis in subsequent sections.

By Eqs. (5.60) and (5.61), Q is a positive recurrent Markov chain, and $Q[k]$ converges to a well defined steady-state distribution, $Q[\infty]$, as $k \to \infty$. Letting $\pi_i = \mathbb{P}(Q[\infty]=i)$, it is easy to verify via the balance equations that

$$\pi_i = \pi_0 \frac{x(\lambda+1-p)}{\lambda} \cdot \left(\frac{1-p}{\lambda}\right)^{i-1}, \quad \forall i \geq 1, \tag{5.62}$$

and since $\sum_{i\geq 0} \pi_i = 1$, we obtain

$$\pi_0 = \frac{1}{1+x\cdot\frac{\lambda+1-p}{\lambda-(1-p)}}. \tag{5.63}$$

Since the chain Q is also irreducible, the limiting fraction of time that Q spends in state 0 is therefore equal to π_0:

$$\lim_{k\to\infty} \frac{1}{k}\sum_{t=1}^{k} \mathbb{I}\left(Q[t]=0\right) = \pi_0 = \frac{1}{1+x\cdot\frac{\lambda+1-p}{\lambda-(1-p)}}. \tag{5.64}$$

Next, we would like to know many of these visits to state 0 correspond to a diversion. Recall the notion of a busy period and diversion epoch, defined in Eqs. (5.49) and (5.50), respectively. By Lemma 5.17, k corresponds to a diversion if any only if $Q[k] = Q[k-1] = 0$. Consider a diversion in slot m_i. If $Q[m_i + 1] = 0$, then $m_i + 1$ also corresponds to a diversion, i.e., $m_i + 1 = m_{i+1}$. If instead $Q[m_i + 1] = 1$, which happens with probability x, the fact that $Q[m_{i+1} - 1] = 0$ implies that there exists at least one busy period, $\{d, \ldots, u\}$, between m_i and m_{i+1}, with $d = m_i$ and $u \leq m_{i+1} - 1$. At the end of this period, a new busy period starts with probability x, and so on. In summary, a diversion epoch E_i consists of the slot $m_i - m_1$, plus N_i busy periods, where the N_i are i.i.d, with[15]

$$N_1 \overset{d}{=} \text{Geo}(1-x) - 1, \tag{5.65}$$

and hence

$$|E_i| = 1 + \sum_{j=1}^{N_i} B_{i,j}, \tag{5.66}$$

where $\{B_{i,j} : i, j \in \mathbb{N}\}$ are i.i.d random variables, and $B_{i,j}$ corresponds to the length of the jth busy period in the ith epoch.

Define $W[t] = (Q[t], Q[t+1])$, $t \in \mathbb{Z}_+$. Since Q is Markov, $W[t]$ is also a Markov chain, taking values in ${\mathbb{Z}_+}^2$. Since a diversion occurs in slot t if and only if $Q[t] = Q[t-1] = 0$ (Lemma 5.17), $|E_i|$ corresponds to excursion times between two adjacent visits of W to the the state $(0,0)$, and hence are i.i.d. Using the Elementary Renewal Theorem, we have

$$\lim_{k\to\infty} \frac{1}{k} I(M,k) = \frac{1}{\mathbb{E}(|E_1|)}, \quad a.s., \tag{5.67}$$

by viewing each visit of W to $(0,0)$ as a renewal event and using the fact that exactly one diversion occurs within a diversion epoch. Denoting by R_i the number of visits to

[15] Geo(p) denotes a geometric random variable with mean $\frac{1}{p}$.

state 0 within E_i, we have that $R_i = 1+N_i$. Treating R_i as the reward associated with the renewal interval E_i, we have, by the time-average of a renewal reward process (c.f., Theorem 6, Chapter 3, [33]), that

$$\lim_{k\to\infty}\frac{1}{k}\sum_{t=1}^{k}\mathbb{I}\left(Q[t]=0\right)=\frac{\mathbb{E}\left(R_1\right)}{\mathbb{E}\left(|E_1|\right)}=\frac{\mathbb{E}\left(N_1\right)+1}{\mathbb{E}\left(|E_1|\right)},\quad a.s., \tag{5.68}$$

by treating each visit of Q to $(0,0)$ as a renewal event. From Eqs. (5.67) and (5.68), we have

$$\frac{\lim_{k\to\infty}\frac{1}{k}I\left(M,k\right)}{\lim_{k\to\infty}\frac{1}{k}\sum_{t=1}^{k}\mathbb{I}\left(Q[t]=0\right)}=\frac{1}{\mathbb{E}(N_1)}=1-x. \tag{5.69}$$

Combing Eqs. (5.24), (5.64) and (5.69), and the fact that $\mathbb{E}(N_1)=\mathbb{E}(\text{Geo}(1-x))-1=\frac{1}{1-x}-1$, we have

$$\frac{\lambda-(1-p)}{\lambda+1-p}\cdot\left[1+x\cdot\frac{\lambda+1-p}{\lambda-(1-p)}\right]=1-x, \tag{5.70}$$

which yields

$$x=\frac{1-p}{\lambda+1-p}. \tag{5.71}$$

This completes the proof of Proposition 5.18. □

We summarize some of the key consequences of Proposition 5.18 below, most of which are easy to derive using renewal theory and well-known properties of positive-recurrent random walks.

Proposition 5.20. *Suppose that* $1>\lambda>1-p>0$*, and denote by* $Q[\infty]$ *the steady-state distribution of* Q*.*

1. *For all* $i\in\mathbb{Z}_+$,

$$\mathbb{P}\left(Q[\infty]=i\right)=\left(1-\frac{1-p}{\lambda}\right)\cdot\left(\frac{1-p}{\lambda}\right)^i. \tag{5.72}$$

2. Almost surely, we have that

$$\lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} Q[i] = \mathbb{E}\left(Q\left[\infty\right]\right) = \frac{1-p}{\lambda-(1-p)}. \tag{5.73}$$

3. Let $E_i = \{m_i^\Psi, m_i^\Psi + 1, \ldots, m_{i+1}^\Psi - 1, m_{i+1}^\Psi\}$. *Then the* $|E_i|$ *are i.i.d, with*

$$\mathbb{E}\left(|E_1|\right) = \frac{1}{\lim_{k\to\infty} \frac{1}{k} I\left(M^\Psi, k\right)} = \frac{\lambda+1-p}{\lambda-(1-p)}, \tag{5.74}$$

and there exists $a, b > 0$ *such that for all* $x \in \mathbb{R}_+$

$$\mathbb{P}\left(|E_1| \geq x\right) \leq a \cdot \exp\left(-b \cdot x\right). \tag{5.75}$$

4. Almost surely, we have that

$$m_i^\Psi \sim \frac{1}{\mathbb{E}\left(|E_1|\right)} \cdot i = \frac{\lambda-(1-p)}{\lambda+1-p} \cdot i, \tag{5.76}$$

as $i \to \infty$.

Proof. Claim 1 follows from the well-known steady-state distribution of a positive recurrent reflected random walk, or equivalently, the fact that $Q[\infty]$ has the same distribution as the steady-state number of jobs in an $M/M/1$ queue with traffic intensity $\rho = \frac{1-p}{\lambda}$. For Claim 2, since Q is an irreducible Markov chain that is positive recurrent, it follows that its time-average coincides with $\mathbb{E}\left(Q[\infty]\right)$ almost surely.

The fact that the E_is are i.i.d was shown in the discussion preceding Eq. (5.67) in the proof of Proposition 5.18. The value of $\mathbb{E}\left(|E_1|\right)$ follows by combining Eqs. (5.24) and (5.67).

Let $B_{i,j}$ be the length of the jth busy period (defined in Eq. (5.49)) in E_i. By definition, $B_{1,1}$ is distributed as the time till the random walk Q reaches state 0, starting from state 1. We have

$$\mathbb{P}\left(B_{1,1} \geq x\right) \leq \mathbb{P}\left(\sum_{j=1}^{\lfloor x \rfloor} X_j \leq -1\right),$$

where the X_j's are i.i.d, with $\mathbb{P}\left(X_1 = 1\right) = \frac{1-p}{\lambda+1-p}$ and $\mathbb{P}\left(X_1 = -1\right) = \frac{\lambda}{\lambda+1-p}$, which, by the Chernoff bound, implies an exponential tail bound for $\mathbb{P}\left(B_{1,1} \geq x\right)$, and in particular,

$$\lim_{\theta \downarrow 0} G_{B_{1,1}}(\theta) = 1, \tag{5.77}$$

By Eq. (5.66), the moment generating function for $|E_1|$ is given by

$$\begin{aligned}
G_{|E_1|}(\epsilon) =& \mathbb{E}\left(\exp\left(\epsilon \cdot |E_1|\right)\right) \\
=& \mathbb{E}\left(\exp\left(\epsilon \cdot \left(1 + \sum_{j=1}^{N_1} B_{1,j}\right)\right)\right) \\
\overset{(a)}{=}& \mathbb{E}\left(e^{\epsilon}\right) \cdot \mathbb{E}\left(\exp\left(N_1 \cdot G_{B_{1,1}}(\epsilon)\right)\right) \\
=& \mathbb{E}\left(e^{\epsilon}\right) \cdot G_{N_1}\left(\ln\left(G_{B_{1,1}}(\epsilon)\right)\right),
\end{aligned} \tag{5.78}$$

where (a) follows from the fact that $\{N_1\} \cup \{B_{1,j} : j \in \mathbb{N}\}$ are mutually independent, and $G_{N_1}(x) = \mathbb{E}\left(\exp\left(x \cdot N_1\right)\right)$. Since $N_1 \overset{d}{=} \text{Geo}(1-x) - 1$, $\lim_{x \downarrow 0} G_{N_1}(x) = 1$, and by Eq. (5.77), we have that $\lim_{\epsilon \downarrow 0} G_{|E_1|}(\epsilon) = 1$, which implies Eq. (5.75).

Finally, Eq. (5.76) follows from the third claim and the Elementary Renewal Theorem. □

5.6.3 Optimality of the No-Job-Left-Behind Policy in Heavy Traffic

This section is devoted to proving the optimality of π_{NOB} as $\lambda \to 1$, stated in the second claim of Theorem 5.10, which we isolate here in the form of the following proposition.

Proposition 5.21. *Fix* $p \in (0,1)$. *We have that*

$$\lim_{\lambda \to 1} C\left(p, \lambda, \pi_{NOB}\right) = \lim_{\lambda \to 1} C^{*}_{\infty}\left(p, \lambda\right).$$

The proof is given at the end of this section, and we do so by showing the following:

1. Over a finite horizon N and given a fixed number of diversions to be made, a greedy diversion rule is optimal in minimizing the post-diversion area under Q over $\{1, \ldots, N\}$.

2. Any point of diversion chosen by π_{NOB} will also be chosen by the greedy policy, as $N \to \infty$.

3. The fraction of points chosen by the greedy policy but not by π_{NOB} diminishes as $\lambda \to 1$, and hence the delay produced by π_{NOB} is the best possible, as $\lambda \to 1$.

Fix $N \in \mathbb{N}$. Let $S\left(Q, N\right)$ be the partial sum $S\left(Q, N\right) = \sum_{k=1}^{N} Q\left[k\right]$. For any sample path Q, denote by $\Delta\left(Q, k\right)$ the marginal decrease of area under Q over the horizon $\{1, \ldots, N\}$ by applying a diversion at slot k, i.e.,

$$\Delta_P\left(Q, N, k\right) = S\left(Q, N\right) - S\left(D_P\left(Q, k\right), N\right),$$

and, analogously,

$$\Delta(Q,N,M') = S(Q,N) - S(D(Q,M'),N),$$

where M' is a diversion sequence.

We next define the notion of a greedy diversion rule, which constructs a diversion sequence by recursively adding the slot that leads to the maximum marginal decrease in $S(Q,N)$.

Definition 5.22. *(Greedy Diversion Rule) Fix an initial sample path Q^0, and $K, N \in \mathbb{N}$. The* ***greedy diversion rule*** *is a mapping, $G(Q^0,N,K)$, which outputs a finite diversion sequence $M^G = \{m_i^G : 1 \le i \le K\}$, given by*

$$\begin{aligned} m_1^G &\in \arg\max_{m \in \Phi(Q^0,N)} \Delta_P\left(Q^0,N,m\right), \\ m_l^G &\in \arg\max_{m \in \Phi(Q^{l-1},N)} \Delta_P\left(Q_{M^G}^{l-1},N,m\right), \quad 2 \le l \le K, \end{aligned}$$

where $\Phi(Q,N) = \Phi(Q) \cap \{1,\dots,N\}$ is the set of all locations in Q in the first N slots that can be diverted, and $Q_{M^G}^l = D(Q^0,\{m_i^G : 1 \le i \le l\})$. Note that we will allow $m_l^G = \infty$ if there is no more entry to divert (i.e., $\Phi(Q^{l-1}) \cap \{1,\dots,N\} = \varnothing$).

We now state a key lemma that will be used in proving Theorem 5.10. It shows that over a finite horizon and for a finite number of diversions, the greedy diversion rule yields the maximum reduction in the area under the sample path.

Lemma 5.23. ***(Dominance of Greedy Policy)*** *Fix an initial sample path Q^0, horizon $N \in \mathbb{N}$, and number of diversions $K \in \mathbb{N}$. Let M' be any diversion sequence with $I(M',N) = K$. Then,*

$$S\left(D\left(Q^0,M'\right),N\right) \ge S\left(D\left(Q^0,M^G\right),N\right),$$

where $M^G = G(Q^0, N, K)$ is the diversion sequence generated by the greedy policy.

Proof. By Lemma 5.3, it suffices to show that, for any sample path $\{Q[k] \in \mathbb{Z}_+ : k \in \mathbb{N}\}$ with $|Q[k+1] - Q[k]| = 1$ if $Q[k] > 0$ and $|Q[k+1] - Q[k]| \in \{0,1\}$ if $Q[k] = 0$, we have

$$S(D(Q, M'), N) \geq \Delta_P(Q, N, m_1^G) + \min_{\substack{|\tilde{M}|=l-1, \\ \tilde{M} \subset \Phi(D(Q, m_1^G), N)}} S\left(D\left(Q_{M^G}^1, \tilde{M}\right), N\right). \quad (5.79)$$

By induction, this would imply that we should use the greedy rule at every step of diversion up to K. The following lemma states a simple monotonicity property. The proof is elementary, and is omitted.

Lemma 5.24. ***(Monotonicity in Diversions)*** *Let Q and Q' be two sample paths such that*

$$Q[k] \leq Q'[k], \quad \forall k \in \{1, \ldots, N\}.$$

Then, for any $K \geq 1$,

$$\min_{\substack{|M|=K, \\ M \subset \Phi(Q,N)}} S(D(Q, M), N) \leq \min_{\substack{|M|=K, \\ M \subset \Phi(Q',N)}} S(D(Q', M), N). \quad (5.80)$$

and, for any finite diversion sequence $M' \subset \Phi(Q, N)$,

$$\Delta(Q, N, M') \geq \Delta(Q', N, M'). \quad (5.81)$$

Recall the definition of a busy period in Eq. (5.49). Let $J(Q, N)$ be the total number of busy periods in $\{Q[k] : 1 \leq k \leq N\}$, with the additional convention $Q[N+1] \triangleq 0$ so that the last busy period always ends on N. Let $B_j = \{d_j, \ldots, u_j\}$ be the jth busy period. It can be verified that a diversion in location k leads to a decrease in the value of $S(Q,N)$ that is no more than the width of the busy period to which k belongs (c.f., Figure 5-6). Therefore, by definition, a greedy policy always seeks to divert at each step the first arriving job during a longest busy period in the current sample path, and hence

$$\Delta\left(Q,N,G(Q,N,1)\right) = \max_{1\le j\le J(Q,N)} |B_j|. \tag{5.82}$$

Let

$$\mathcal{J}^*(Q,N) = \arg \max_{1\le j\le J(Q,N)} |B_j|.$$

We consider the following cases, depending on whether M' chooses to divert any job in the busy periods in $\mathcal{J}^*(Q,N)$.

Case 1: $M' \cap \left(\cup_{j\in\mathcal{J}^*(Q,N)} B_j\right) \neq \varnothing$. If $d_{j^*} \in M'$ for some $j^* \in \mathcal{J}^*$, by Eq. (5.82), we can set m_1^G to d_{j^*}. Since $m_1^G \in M'$ and the order of diversions does not impact the final resulting delay (Lemma 5.3), we have that Eq. (5.79) holds, and we are done. Otherwise, choose $m^* \in M' \cap B_{j^*}$ for some $j^* \in \mathcal{J}^*$, and we have $m^* > d_{j^*}$. Let

$$Q' = D_P\left(Q,m^*\right), \text{ and } \hat{Q} = D_P\left(Q,d_{j^*}\right).$$

Since $Q[k] > 0, \forall k \in \{d_{j^*},\ldots,u_{j^*}-1\}$, we have $\hat{Q}[k] = Q[k] - 1 \le Q'[k]$, $\forall k \in \{d_{j^*},\ldots,u_{j^*}-1\}$, and $Q'[k] = Q[k] = \hat{Q}[k]$, $\forall k \notin \{d_{j^*},\ldots,u_{j^*}-1\}$, which implies that

$$\hat{Q}[k] \le Q'[k], \quad \forall k \in \{1,\ldots,N\}. \tag{5.83}$$

Eq. (5.79) holds by combining Eq. (5.83) and Eq. (5.80) in Lemma 5.24, with $K = l-1$.

Case 2: $M' \cap \left(\cup_{j\in\mathcal{J}^*(Q,N)} B_j\right) = \varnothing$. Let m^* be any element in M', and $Q' = D_P\left(Q,m^*\right)$. Clearly, $Q[k] \ge Q'[k]$ for all $k \in \{1,\ldots,N\}$, and by Eq. (5.81) in

Lemma 5.24, we have that[16]

$$\Delta\left(Q,N,M'\backslash\{m^*\}\right) \geq \Delta\left(D_P\left(Q,m^*\right),N,M'\backslash\{m^*\}\right). \tag{5.84}$$

Since $M' \cap \left(\cup_{j\in\mathcal{J}^*(Q,N)} B_j\right) = \varnothing$, we have that

$$\Delta_P\left(D\left(Q,M'\backslash\{m^*\}\right),N,m_1^G\right) = \max_{1\leq j\leq J(Q,N)} |B_j| > \Delta_P\left(Q,N,m^*\right). \tag{5.85}$$

Let $\hat{M} = m_1^G \cup \left(M'\backslash\{m^*\}\right)$, we have that

$$\begin{aligned}
& S\left(D\left(Q,\hat{M}\right),N\right) \\
= \; & S\left(Q,N\right) - \Delta\left(Q,N,M'\backslash\{m^*\}\right) - \Delta_P\left(D\left(Q,M'\backslash\{m^*\}\right),N,m_1^G\right) \\
\overset{(a)}{\leq} \; & S\left(Q,N\right) - \Delta\left(D_P\left(Q,m^*\right),N,M'\backslash\{m^*\}\right) - \Delta_P\left(D\left(Q,M'\backslash\{m^*\}\right),N,m_1^G\right) \\
\overset{(b)}{<} \; & S\left(Q,N\right) - \Delta\left(D_P\left(Q,m^*\right),N,M'\backslash\{m^*\}\right) - \Delta_P\left(Q,N,m^*\right) \\
= \; & S\left(D\left(Q,M'\right),N\right),
\end{aligned}$$

where (a) and (b) follow from Eqs. (5.84) and (5.85), respectively. This shows that Eq. (5.79) holds (and in this case the inequality there is strict).

Cases 1 and 2 together complete the proof of Lemma 5.23.

□

We are now ready to prove Proposition 5.21.

Proof. **(Proposition 5.21)** Lemma 5.23 shows that, for any fixed number of diversions over a finite horizon N, the greedy diversion policy (Definition 5.22) yields the smallest area under the resulting sample path, Q, over $\{1,\ldots,N\}$. The main idea

[16] For finite sets A and B, $A\backslash B = \{a \in A : a \notin B\}$.

of proof is to show that the area under Q after applying π_{NOB} is asymptotically the same as that of the greedy policy, as $N \to \infty$ and $\lambda \to 1$ (in this particular order of limits). In some sense, this means that the jobs in M^{Ψ} account for almost all of the delays in the system, as $\lambda \to 1$. The following technical lemma is useful.

Lemma 5.25. *For a finite set $S \subset \mathbb{R}$, and $l \in \mathbb{N}$, define*

$$f(S,l) = \frac{\textit{sum of the } l \textit{ largest elements in } S}{|S|}.$$

Let $\{X_i : 1 \le i \le k\}$ be i.i.d random variables taking values in $\mathbb{Z}_+$, where $\mathbb{E}(X_1) < \infty$. Then for any sequence of random variables $\{H_k : k \in \mathbb{N}\}$, with $H_k \lesssim \alpha k$ a.s. as $k \to \infty$ for some $\alpha \in (0,1)$, we have

$$\limsup_{k\to\infty} f(\{X_i : 1 \le i \le k\}, H_k) \le \mathbb{E}\left(X_1 \cdot \mathbb{I}\left(X_1 \ge \overline{F}_{X_1}^{-1}(\alpha)\right)\right), \quad a.s., \tag{5.86}$$

where $\overline{F}_{X_1}^{-1}(y) = \min\{x \in \mathbb{N} : \mathbb{P}(X_1 \ge x) < y\}$.

Proof. See Appendix B.1.4. □

Fix an initial sample path Q^0. We will denote by $M^{\Psi} = \{m_i^{\Psi} : i \in \mathbb{N}\}$ the diversion sequence generated by π_{NOB} on Q^0. Define

$$d(k) = k - \max_{1 \le i \le I(M^{\Psi},k)} |E_i| \tag{5.87}$$

where E_i is the ith diversion epoch of M^{Ψ}, defined in Eq. (5.50). Since $Q^0[k] \ge Q^0[m_i]$ for all $i \in \mathbb{N}$, it is easy to check that

$$\Delta_P\left(D\left(Q^0, \{m_j^{\Psi} : 1 \le j \le i-1\}\right), k, m_i^{\Psi}\right) = k - m_i^{\Psi} + 1,$$

for all $i \in \mathbb{N}$. The function l was defined so that the first $I(M^{\Psi}, l(k))$ diversions made by a greedy rule over the horizon $\{1, \ldots, k\}$ are exactly $\{1, \ldots, l(k)\} \cap M^{\Psi}$. More formally, we have the following lemma.

Lemma 5.26. *Fix $k \in \mathbb{N}$, and let $M^G = G(Q^0, k, I(M^{\Psi}, d(k)))$. Then $m_i^G = m_i^{\Psi}$, for all $i \in \{1, \ldots, I(M^{\Psi}, l(k))\}$.*

Fix $K \in \mathbb{N}$, and an arbitrary feasible diversion sequence, $\tilde{M}$, generated by a policy in Π_∞. We can write

$$\begin{aligned} I\left(\tilde{M}, m_K^{\Psi}\right) =& I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right) + \left(I\left(M^{\Psi}, m_K^{\Psi}\right) - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right)\right) \\ & + \left(I\left(\tilde{M}, m_K^{\Psi}\right) - I\left(M^{\Psi}, m_K^{\Psi}\right)\right) \\ =& I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right) + \left(K - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right)\right) \\ & + \left(I\left(\tilde{M}, m_K^{\Psi}\right) - I\left(M^{\Psi}, m_K^{\Psi}\right)\right) \\ =& I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right) + h(K), \end{aligned} \tag{5.88}$$

where

$$h(K) = \left(K - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right)\right) + \left(I\left(\tilde{M}, m_K^{\Psi}\right) - I\left(M^{\Psi}, m_K^{\Psi}\right)\right). \tag{5.89}$$

We have the following characterization of h.

Lemma 5.27. *$h(K) \lesssim \frac{1-\lambda}{\lambda-(1-p)} \cdot K$, as $K \to \infty$, a.s.*

Proof. See Appendix B.1.5 □

Let

$$M^{G,k} = G\left(Q^0, k, I\left(\tilde{M}, k\right)\right), \tag{5.90}$$

where the greedy diversion map G was defined in Definition 5.22. By Lemma 5.26 and the definition of $M^{G,k}$, we have that

$$M^{\Psi} \cap \left\{1,\ldots,d\left(m_K^{\Psi}\right)\right\} \subset M^{G,m_K^{\Psi}}. \tag{5.91}$$

Therefore, we can write

$$M^{G,m_K^{\Psi}} = \left(M^{\Psi} \cap \left\{1,\ldots,d\left(m_K^{\Psi}\right)\right\}\right) \cup \overline{M}_K^G, \tag{5.92}$$

where $\overline{M}_K^G \triangleq M^{G,m_K^{\Psi}} \backslash \left(M^{\Psi} \cap \{1,\ldots,d\left(m_K^{\Psi}\right)\}\right)$. Since $\left|M^{G,m_K^{\Psi}}\right| = I\left(\tilde{M},m_K^{\Psi}\right)$ by definition, by Eq. (5.88),

$$\left|\overline{M}_K^G\right| = h(K). \tag{5.93}$$

We have

$$\begin{aligned} & S\left(D\left(Q^0,M^{\Psi}\right),m_K^{\Psi}\right) - S\left(D\left(Q^0,\tilde{M}\right),m_K^{\Psi}\right) \\ & \overset{(a)}{\leq} S\left(D\left(Q^0,M^{\Psi}\right),m_K^{\Psi}\right) - S\left(D\left(Q^0,M^{G,m_K^{\Psi}}\right),m_K^{\Psi}\right) \\ & \overset{(b)}{=} \Delta\left(D\left(Q^0,M^{\Psi}\right),m_K^{\Psi},\overline{M}_K^G\right), \end{aligned} \tag{5.94}$$

where (a) is based on the dominance of the greedy policy over any finite horizon (Lemma 5.23), and (b) follows from Eq. (5.92).

Finally, we claim that there exists $g(x) : \mathbb{R} \to \mathbb{R}_+$, with $g(x) \to 0$ as $x \to 1$, such that

$$\limsup_{K\to\infty} \frac{\Delta\left(D\left(Q^0,M^{\Psi}\right),m_K^{\Psi},\overline{M}_K^G\right)}{m_K^{\Psi}} \leq g(\lambda), \quad a.s. \tag{5.95}$$

Eqs. (5.94) and (5.95) combined imply that

$$\begin{aligned} C\left(p,\lambda,\pi_{NOB}\right) =& \limsup_{K\to\infty} \frac{S\left(D\left(Q^0,M^\Psi\right),m_K^\Psi\right)}{m_K^\Psi} \\ \leq& g(\lambda) + \limsup_{K\to\infty} \frac{S\left(D\left(Q^0,\tilde{M}\right),m_K^\Psi\right)}{m_K^\Psi}, \\ =& g(\lambda) + \limsup_{k\to\infty} \frac{S\left(D\left(Q^0,\tilde{M}\right),k\right)}{k}, \quad a.s., \end{aligned} \tag{5.96}$$

which shows that

$$C\left(p,\lambda,\pi_{NOB}\right) \leq g(\lambda) + \inf_{\pi\in\Pi_\infty} C\left(p,\lambda,\pi\right).$$

Since $g(\lambda) \to 0$ as $\lambda \to 1$, this proves Proposition 5.21.

To show Eq. (5.95), denote by Q the sample path after applying π_{NOB},

$$Q = D\left(Q^0, M^\Psi\right),$$

and by V_i the area under Q within E_i,

$$V_i = \sum_{k=m_i^\Psi}^{m_{i+1}^\Psi - 1} Q[k].$$

An example of V_i is illustrated as the area of the shaded region in Figure 5-6. By Proposition 5.18, Q is a Markov chain and so is the process $W[k] = (Q[k], Q[k+1])$. By Lemma 5.17, E_i corresponds to the indices between two adjacent returns of the chain W to state $(0,0)$. Since the ith return of a Markov chain to a particular state is a stopping time, it can be shown, using the strong Markov property of W, that the segments of Q, $\{Q[k] : k \in E_i\}$, are mutually independent and identically distributed among different values of i. Therefore, the V_i's are i.i.d. Furthermore,

$$\mathbb{E}\left(V_1\right) \overset{(a)}{\leq} \mathbb{E}\left(|E_1|^2\right) \overset{(b)}{<} \infty, \tag{5.97}$$

where (a) follows from the fact that $|Q[k+1]-Q[k]| \leq 1$ for all k, and hence $V_i \leq |E_i|^2$ for any sample path of Q^0, and (b) from the exponential tail bound on $\mathbb{P}(|E_1| \geq x)$, given in Eq. (5.75).

Since the value of Q on the two ends of E_i, m_i^Ψ and $m_{i+1}^\Psi - 1$, are both zero, each additional diversion within E_i cannot produce a marginal decrease of area under Q of more than V_i (c.f., Figure 5-6). Therefore, the value of $\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right)$ can be no greater than the sum of the $h(K)$ largest V_i's over the horizon $k \in \{1, \ldots, m_K^\Psi\}$. We have

$$\begin{aligned}
&\limsup_{K\to\infty} \frac{\Delta\left(D\left(Q^0, M^\Psi\right), m_K^\Psi, \overline{M}_K^G\right)}{m_K^\Psi} \\
&= \limsup_{K\to\infty} f\left(\{V_i : 1 \leq i \leq K\}, h(K)\right) \cdot \frac{K}{m_K^\Psi} \\
&\overset{(a)}{=} \limsup_{K\to\infty} f\left(\{V_i : 1 \leq i \leq K\}, h(K)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)} \\
&\overset{(b)}{=} \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda-(1-p)}\right)\right)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)}
\end{aligned} \tag{5.98}$$

where (a) follows from Eq. (5.76), and (b) from Lemmas 5.25 and 5.27. Since $\mathbb{E}\left(V_1\right) < \infty$, and $\overline{F}_{V_1}^{-1}(x) \to \infty$ as $x \to 0$, it follows that

$$\mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda-(1-p)}\right)\right)\right) \to 0,$$

as $\lambda \to 1$. Eq. (5.95) is proved by setting $g(\lambda) = \mathbb{E}\left(V_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{V_1}^{-1}\left(\frac{1-\lambda}{\lambda-(1-p)}\right)\right)\right) \cdot \frac{\lambda+1-p}{\lambda-(1-q)}$. This completes the proof of Proposition 5.21. □

Why not use Greedy?

The proof of Proposition 5.21 relies on a sample-path-wise coupling to the performance of a greedy diversion rule. It is then only natural to ask: since the time horizon is indeed finite in all practical applications, why don't we simply use the greedy rule as the preferred offline policy, as opposed to π_{NOB}?

There are at least two reasons for focusing on π_{NOB} instead of the greedy rule. First, the structure of the greedy rule is highly global, in the sense that each diversion decision uses information of the entire sample path over the horizon. As a result, the greedy rule tells us little on how to design a good policy with a *fixed* lookahead window (e.g., Theorem 5.13). In contrast, the performance analysis of π_{NOB} in Section 5.6.2 reveals a highly *regenerative* structure: the diversions made by π_{NOB} essentially depend only on the dynamics of Q^0 in the same diversion epoch (the E_i's), and what happens beyond the current epoch becomes irrelevant. This is the key intuition that led to our construction of the finite-lookahead policy in Theorem 5.13. A second (and perhaps minor) reason is that of computational complexity. By a small sacrifice in performance, π_{NOB} can be efficiently implemented using a linear-time algorithm (Section 5.4.2), while it is easy to see that a naive implementation of the greedy rule would require super-linear complexity with respect to the length of the horizon.

5.6.4 Proof of Theorem 5.10

Proof. **(Theorem 5.10)** The fact that π_{NOB} is feasible follows from Eq. (5.24) in Lemma 5.15, i.e.

$$\limsup_{k\to\infty} \frac{1}{k} I\left(M^{\Psi}, k\right) \le \frac{\lambda - (1-p)}{\lambda + 1 - p} < \frac{p}{\lambda + 1 - p}, \quad \text{a.s.}$$

Let $\{\tilde{Q}[k] : k \in \mathbb{Z}_+\}$ be the resulting sample path after applying π_{NOB} to the initial sample path $\{Q^0[k] : k \in \mathbb{Z}_+\}$, and let

$$Q[k] = \tilde{Q}\left[k + m_1^\Psi\right], \quad \forall k \in \mathbb{N},$$

where m_1^Ψ is the index of the first diversion made by π_{NOB}. Since $\lambda > 1 - p$, the random walk Q^0 is transient, and hence $m_1^\Psi < \infty$ almost surely. We have that, almost surely,

$$\begin{aligned} C(p, \lambda, \pi_{NOB}) &= \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} \tilde{Q}[i] \\ &= \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{m_1^\Psi} \tilde{Q}[i] + \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} Q[i] \\ &= \frac{1-p}{\lambda - (1-p)}, \end{aligned} \tag{5.99}$$

where the last equality follows from Eq. (5.73) in Proposition 5.20, and the fact that $m_1 < \infty$ almost surely. Letting $\lambda \to 1$ in Eq. (5.99) yields the finite limit of delay under heavy traffic:

$$\lim_{\lambda\to 1} C(p, \lambda, \pi_{NOB}) = \lim_{\lambda\to 1} \frac{1-p}{\lambda - (1-p)} = \frac{1-p}{p}.$$

Finally, the delay optimality of π_{NOB} in heavy traffic was proved in Proposition 5.21, i.e., that

$$\lim_{\lambda\to 1} C(p, \lambda, \pi_{NOB}) = \lim_{\lambda\to 1} C_\infty^*(p, \lambda).$$

This completes the proof of Theorem 5.10. □

5.7 Policies with a Finite Lookahead

5.7.1 Proof of Theorem 5.13

Proof. **(Theorem 5.13)** As pointed out in the discussion preceding Theorem 5.13, for any initial sample path and $w < \infty$, an arrival that is diverted under the π_{NOB} policy will also be diverted under π^{w}_{NOB}. Therefore, the delay guarantee for π_{NOB} (Theorem 5.10) carries over to $\pi^{w_\lambda}_{NOB}$, and for the rest of the proof, we will be focusing on showing that $\pi^{w_\lambda}_{NOB}$ is feasible under an appropriate scaling of w_λ. We begin by stating an exponential tail bound on the distribution of the discrete-time predictive window, $W(\lambda, k)$, defined in Eq. (5.17),

$$W(\lambda, k) = \max\left\{l \in \mathbb{Z}_+ : T_{k+l} \leq T_k + w_\lambda\right\}.$$

It is easy to see that $\{W(\lambda, m^{\Psi}_i) : i \in \mathbb{N}\}$ are i.i.d, with $W(\lambda, m^{\Psi}_1)$ distributed as a Poisson random variable with mean $(\lambda + 1 - p)w_\lambda$. Since

$$\mathbb{P}\left(W\left(\lambda, m^{\Psi}_1\right) \geq x\right) \leq \mathbb{P}\left(\sum_{l=1}^{\lfloor w_\lambda \rfloor} X_l\right),$$

where the X_l are i.i.d Poisson random variables with mean $\lambda + (1-p)$, applying the Chernoff bound, we have that, there exist $c, d > 0$ such that

$$\mathbb{P}\left(W\left(\lambda, m^{\Psi}_1\right) \geq \frac{\lambda + 1 - p}{2} \cdot w_\lambda\right) \leq c \cdot \exp(-d \cdot w_\lambda), \tag{5.100}$$

for all $w_\lambda > 0$.

We now analyze the diversion rate resulted by the $\pi^{w_\lambda}_{NOB}$ policy. For the purpose of analysis (as opposed to practical efficiency), we will consider a new diversion policy, denoted by σ^{w_λ}, which can be viewed as a relaxation of $\pi^{w_\lambda}_{NOB}$.

Definition 5.28. *Fix $w \in \mathbb{R}_+$. The diversion policy σ^w is defined so that for each diversion epoch E_i, $i \in \mathbb{N}$,*

1. *if $|E_i| \le W(\lambda, m_i^\Psi)$, then only the first arrival of this epoch, namely, the arrival in slot m_i^Ψ, is diverted;*

2. *otherwise, all arrivals within this epoch are diverted.*

It is easy to verify that σ^w can be implemented with w units of look-ahead, and the set of diversions made by σ^{w_λ} is a strict superset of $\pi^{w_\lambda}_{NOB}$ almost surely. Hence, the feasibility of σ^{w_λ} will imply that of $\pi^{w_\lambda}_{NOB}$.

Denote by D_i the number of diversions made by σ^{w_λ} in the epoch E_i. By the construction of the policy, the D_i are i.i.d, and depend only on the length of E_i and the number of arrivals within. We have[17]

$$\begin{aligned}
\mathbb{E}(D_1) \le& 1 + \mathbb{E}\left[|E_i| \cdot \mathbb{I}\left(|E_i| \ge W\left(\lambda, m_i^\Psi\right)\right)\right] \\
\le& 1 + \mathbb{E}\left[|E_i| \cdot \mathbb{I}\left(|E_i| \ge \frac{\lambda+1-p}{2} \cdot w_\lambda\right)\right] \\
&+ \mathbb{E}(|E_i|) \cdot \mathbb{P}\left(W\left(\lambda, m_i^\Psi\right) \le \frac{\lambda+1-p}{2} \cdot w_\lambda\right) \\
\le& 1 + \left(\sum_{l=\frac{\lambda+1-p}{2}\cdot w_\lambda}^{\infty} l \cdot a \cdot \exp(-b \cdot l)\right) + \frac{\lambda}{\lambda-(1-p)} \cdot c \cdot \exp(-d \cdot w_\lambda) \\
\overset{(a)}{\le}& 1 + h \cdot w_\lambda \cdot \exp(-l \cdot w_\lambda),
\end{aligned} \tag{5.101}$$

for some $h, l > 0$, where step (a) follows from the fact that $\sum_{l=k}^{\infty} l \cdot \exp(-b \cdot l) = \mathcal{O}(k \cdot \exp(-b \cdot k))$ as $k \to \infty$.

[17] For simplicity of notation, we assume that $\frac{\lambda+1-p}{2} \cdot w_\lambda$ is always an integer. This does not change the scaling behavior of w_λ.

Since the D_i are i.i.d, using basic renewal theory, it is not difficult to show that the average rate of diversion in discrete time under the policy σ^{w_λ} is equal to $\frac{\mathbb{E}(D_1)}{\mathbb{E}(E_1)}$. In order for the policy to be feasible, one must have that

$$\frac{\mathbb{E}(D_1)}{\mathbb{E}(E_1)} = \frac{\mathbb{E}(D_1)}{\lambda} \leq \frac{p}{\lambda + 1 - p}. \tag{5.102}$$

By Eqs. (5.101) and (5.102), we want to ensure that

$$\frac{p\lambda}{\lambda - (1-p)} \geq 1 + h \cdot w_\lambda \cdot \exp(-l \cdot w_\lambda),$$

which yields, after taking the logarithm on both sides,

$$w_\lambda \geq \frac{1}{b} \log\left(\frac{1}{1-\lambda}\right) + \frac{1}{b} \log\left(\frac{[\lambda - (1-p)] \cdot h \cdot w_\lambda}{1-p}\right). \tag{5.103}$$

It is not difficult to verify that for all $p \in (0,1)$ there exists a constant $c_h > 0$ such that the above inequality holds for all $\lambda \in (1-p, 1)$, by letting $w_\lambda = c_h \log(\frac{1}{1-\lambda})$. This proves the feasibility of σ^{w_λ}, which implies that $\pi_{NOB}^{w_\lambda}$ is also feasible. This completes the proof of Theorem 5.13. □

5.8 Summary and Future Research

The main objective of this chapter was to study the impact of future information on the performance of a class of admission control problems, with a constraint on the time-average rate of diversion. Our model is motivated as a study of a dynamic resource allocation problem between slow (congestion-prone) and fast (congestion-free) processing resources. It could also be viewed as the decision problem faced by a local queue for flexible sysems in the Partial Pooling family (cf. Section 2.2), where a fraction p of the system's resources are fully flexible, while the remaining resources are dedicated. Our main results show that the availability of future information can dramatically reduce the delay experienced by admitted jobs: the delay converges to a finite constant even as the traffic load approaches the system capacity ("heavy-traffic delay collapse") *if and only if* the decision maker is allowed a sufficiently large lookahead window (Theorems 5.13 and 5.14).

There are several interesting directions for future exploration. We believe that our results can be generalized to cases where the arrival and service processes are non-Poisson. We note that the π_{NOB} policy is indeed feasible for a wide range of non-Poisson arrival and service processes (e.g., renewal processes), as long as they satisfy a form of strong law of large numbers, with appropriate time-average rates (Lemma 5.15). It seems more challenging to generalize the results on the optimality of π_{NOB} and the performance guarantees. However, it may be possible to establish a generalization of the delay optimality result using limiting theorems (e.g., diffusion approximations). For instance, with sufficiently well-behaved arrival and service processes, we expect that one can establish a result similar to Proposition 5.18 by characterizing the queue length process that results from π_{NOB} as a reflected Brownian motion in $\mathbb{R}_+$, in the limit of $\lambda \to 1$ and $p \to 0$, with appropriate scaling.

Another interesting variation of our problem is the setting where each job comes with a prescribed *size*, or *workload*, and the decision maker is able to observe both the arrival times and workloads of jobs up to a finite lookahead window. It is conceivable that many analogous results can be established for this setting, by studying the associated workload (as opposed to queue length) process, while the analysis may be less clean due to the lack of a simple random-walk-based description of the system dynamics. Moreover, the *server* could potentially exploit additional information of the jobs' workloads in making scheduling decisions, and it is unclear what the performance and fairness implications are for the design of admission control policies.

There are other issues that need to be addressed if our offline policies (or policies with a finite lookahead) are to be applied in practice. A most important question relates to the impact of *observation noise* to performance, since in reality the future seen in the lookahead window cannot be expected to match the actual realization exactly. We conjecture, based on the analysis of π_{NOB}, that the performance of both π_{NOB}, and its finite-lookahead version, is robust to small noises or perturbations (e.g., if the actual sample path is at most ϵ away from the predicted one). The policy's robustness under prediction noise have been demonstrated via simulation in [93], while it remains to thoroughly verify and quantify the extent of the impact of noise, either empirically or through theory. Also, it is unclear what the best practices should be when the lookahead window is very small relative to the traffic intensity λ (i.e., $w \ll \log \frac{1}{1-\lambda}$). In this regime, our lower bound (Theorem 5.14) does not fully preclude the possibility of finding effective prediction-guided policies that improve upon an optimal online policy. Our work [93] explores, in the context of admission control for Emergency Departments, some of these issues, such as the impact of prediction noise and that of a short lookahead window, but the picture is far from complete.

Chapter 6

Necessity of Future Information

This chapter is devoted to the proof of the future information lower bound of Theorem 5.14, introduced in Chapter 5, which is re-stated below.

Theorem 6.1 (Necessity of Future Information, rep. of Theorem 5.14). *Fix $p \in (0,1)$. There exist $c_l > 0$ and $\tilde{\lambda} \in (1-p, 1)$, so that if*

$$w_\lambda \leq c_l \ln \frac{1}{1-\lambda}, \quad \forall \lambda \in (\tilde{\lambda}, 1), \tag{6.1}$$

then

$$C^*_{w_\lambda}(p,\lambda) = \Theta\left(\ln \frac{1}{1-\lambda}\right), \quad \textit{as } \lambda \to 1. \tag{6.2}$$

Despite having identical modeling assumptions, the proof techniques used for Theorem 6.1 are quite different from those employed to establish the achievability results of Theorem 5.10 and 5.13 in Chapter 5. This is due to the fact that, in order to establish a lower bound, instead of analyzing *one* policy (e.g., π^w_{NOB}), we will now need tools to characterize the performance of *all* feasible policies. The core of our arguments hinges upon a relationship between *diversion decisions* and *future* *idling* of the server, over a certain subset of input sample paths. This relationship is then used in conjunction with the excursion probabilities of a transient random walk to demonstrate that the system manager *must* maintain a relatively large queue length, when the amount of future information is limited. We believe that this line of arguments is fairly robust to changes in modeling assumptions, and can be generalized to prove information lower bounds for other dynamic resource allocation problems.

Organization The remainder of the chapter is organized as follows. In Section 6.1, we contrast our proof techniques with methods from the literature on Markov decision processes. Section 6.2 reviews the modeling assumptions, and introduces the necessary mathematical formalism. The proof of Theorem 6.1 is given in Section 6.3, with an outline of the proof ideas provided at the beginning of the section. We conclude the chapter in Section 6.4 and examine potential directions for future research.

6.1 Related Research

Theorem 6.1 can be viewed as a generalization of the *Markov* optimal admission control problem that has been studied in the literature [80], and it is interesting to highlight some of the differences in analytical approaches. Optimal policies in the Markov setting ($w_\lambda = 0$) are known to often admit a *threshold* (or control-limit) form, such as the one analyzed in Theorem 5.8, where a diversion is made only if the current queue length reaches a fixed threshold. To prove the optimality of these policies, one would typically analyze the Bellman equations of the corresponding Markov decision process (MDP) in order to establish a set of *monotonicity* properties in the policy space, e.g., that the cost-to-go function associated with a threshold policy would be dominated by those associated with policies that divert with non-zero probabilities when the queue is small (c.f. [94]). Successive application of such monotonicity properties can then narrow the policy space down to only those with a threshold form.

Unfortunately, such arguments employed in the Markov setting do not seem to carry over easily when the lookahead window is taken into account. While our setting can still be casted as an MDP by incorporating events during the lookahead window into the state, the structure of the state space is now considerably more complex (and increasingly so as we let $w_\lambda \to \infty$), and it is no longer clear if any monotonicity property continues to hold. Our proof techniques circumvent this additional complexity by focusing on the "macroscopic" sample-path characteristics of the system, instead of the more refined details of the Bellman equations. As a trade-off, our analysis is more coarse by nature, and it provides neither a characterization of the multiplicative *constant* in the delay scaling, nor a concrete diversion policy that achieves the information lower bound (which, fortunately, has been given in Chapter 5).

Our work is also similar in spirit to the techniques of information relaxation and path-wise optimization for MDPs [19, 27, 69]. In this case, one considers a relaxed version of the original MDP, where the decision maker has access to realizations of the future input sample paths. This relaxed problem is often simpler to solve and simulate than the original stochastic optimization problem, and hence can be used, for instance, as a performance benchmark for evaluating heuristic policies. Our work is different from this literature in several aspects. Most notably, we focus on rigorously understanding the stochastic dynamics involved in the relaxed problem with future information, and how performance scales with respect to the length of the lookahead window, as opposed to using the relaxed problem to approximate the performance of an optimal online policy, which is well understood in our setting.

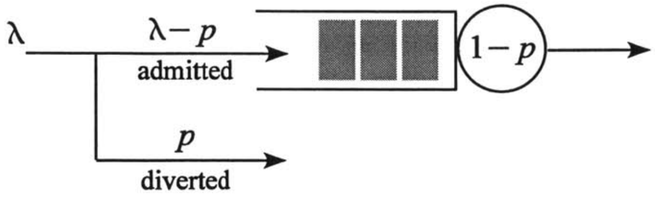

Figure 6-1: An illustration of the queueing admission control problem (rep. of Figure 5-8).

6.2 Model and Notation

We now present the mathematical formalism and modeling assumptions that will be used throughout the remainder of the chapter. We first review the admission control model, described in Section 5.2 of the preceding chapter. We shall, however, introduce some new notation to facilitate the analysis in this chapter. For instance, we will no longer work with the initial queue length, Q^0, and instead focus on a equivalent representation, S, which captures arrival-service discrepancies over finite intervals. A illustration of the model is reproduced in Figure 6-1 for convenience

System Dynamics. The system runs in continuous time, indexed by $t \in \mathbb{R}_+$. There is a *queue* with infinite waiting room, whose length at time t is denoted by $Q(t)$. The input to the system consists of two independent Poisson processes:

1. $\mathcal{A}$, with rate λ, which corresponds to the *arrival* of jobs;

2. $\mathcal{S}$, with rate $1 - p$, which corresponds to the generation of *service tokens.*

When an *event* occurs in $\mathcal{A}$ at time t, we say that a job has arrived to the system, and the value of $Q(t)$ is incremented by 1, if the job is "admitted" (see below for the description of admission policies). Similarly, when an event occurs in the process $\mathcal{S}$ at time t, we say that a service token is generated, and the value of $Q(t)$ is decremented by 1, if $Q(t) > 0$, and remains at 0, otherwise.[1]

Denote by $\mathcal{A} \cup \mathcal{S}$ the point process that consists of the union of the events in $\mathcal{A}$ and $\mathcal{S}$. For our purposes, it is more convenient to work with the sequence $\{(Z_m, R_m) : m \in \mathbb{N}\}$, where

$$Z_m = \text{time of the } m\text{th event in } \mathcal{A} \cup \mathcal{S}, \tag{6.3}$$

and R_m encodes the type of the mth event, with

$$R_m = \begin{cases} 1, & \text{if the } m\text{th event is in } \mathcal{A} \text{ (arrival)}, \\ -1, & \text{if the } m\text{th event is in } \mathcal{S} \text{ (service token)}. \end{cases} \tag{6.4}$$

We will let $\{\mathcal{N}(t) : t \in \mathbb{R}_+\}$ be the counting process associated with $\{Z_m\}$, with

$$\mathcal{N}(t) = \sup\{m \in \mathbb{Z}_+ : Z_m \le t\}, \tag{6.5}$$

and denote by $S(s,t)$ the *difference between the numbers of arrival and services tokens* in the interval $[s,t)$,

$$S(s,t) = \sum_{\mathcal{N}(s) \le m \le \mathcal{N}(t)-1} R_m. \tag{6.6}$$

Note that when $\lambda \ne 1-p$ the process $\{S(0,t) : t \in \mathbb{R}_+\}$ is a transient random walk, with

$$\mathbb{E}(S(0,t)) = [\lambda - (1-p)]t. \tag{6.7}$$

[1] When the queue is non-empty, the generation of a token can be interpreted as the completion of a previous job, upon which the server is ready to fetch the next job. The reader is referred to Section 2.1, for more details on the relationship between the service token model and the more conventional assumption of exponentially distributed job sizes.

Future Information. The notion of future information is captured by a *lookahead window*. At any time t, the system manager has access to the *realization* of all events in $\mathcal{A} \cup \mathcal{S}$ in the interval $[t, t + w_\lambda)$. Throughout the chapter, we will denote by w_λ the length of the lookahead window, under arrival rate λ.

Admission Policies. Upon arrival, each job is either *admitted*, in which case it joins the queue, or *diverted*, in which case it disappears from the system immediately. The role of a diversion policy, π, is to output a sequence of *diversion decisions* for all events, represented by the sequence of indicator variables, $\{H(m) : m \in \mathbb{N}\}$, where

$$H(m) = \mathbb{I}\{R_m = 1, \text{ and } \pi \text{ chooses to divert at time } Z_m\}. \tag{6.8}$$

Given the form of future information, we will require the diversion policy to be w_λ-causal, so that the decision made at time t depends only on the events in the time interval $[0, t + w_\lambda)$. A diversion policy is said to be *feasible*, if the resulting time-average rate of diversion is at most p, i.e.,

$$\limsup_{M \to \infty} \frac{\lambda + 1 - p}{M} \mathbb{E}\left(\sum_{m=1}^{M} H(m) \right) \le p. \tag{6.9}$$

where the constant $\lambda + 1 - p$ corresponds to the total rate of events in $\mathcal{A} \cup \mathcal{S}$. The objective of the decision maker is to choose a feasible policy, π, so as to *minimize* the time-average queue length, defined by[2]

$$C(p, \lambda, \pi) = \limsup_{M \to \infty} \mathbb{E}\left(\frac{1}{M} \sum_{m=1}^{M} Q(Z_m+) \right). \tag{6.10}$$

[2] Throughout, $f(x+)$ represents the limit $\lim_{y \downarrow x} f(y)$.

6.3 Proof of Theorem 6.1

The remainder of the chapter is devoted to the proof of Theorem 6.1. We begin with a high-level summary of the main steps involved. First, we argue that there exists a stationary optimal policy, which makes decisions only based on the current queue length and the content of the lookahead window (Section 6.3.1). This stationarity will allow us to simplify the analysis by focusing on the policy's actions over a finite time horizon.

We will prove Theorem 6.1 by contradiction, where we start by assuming that a small average queue length is indeed achievable under an optimal stationary policy, even with a short lookahead window, and later refute this assumption. Our main arguments are based on the identification of a set of *base sample paths* (Section 6.3.2), with the property that *any* feasible policy must perform poorly over these sample paths, should the length of the lookahead window be too small. The stationarity property described earlier will then allow us to extend this and show the policy's failure over the infinite time horizon. It is worth noting that the base sample paths are not *typical*, in the sense that their occurrences possess only vanishingly small probability, as $\lambda \to 1$. This is because the failures of a policy under a small lookahead window are not caused by the average behavior of the inputs, but rather by some rare excursions of the random walk $S(0, \cdot)$. Though occurring with small probability, these excursions are in some sense unforeseeable under a small lookahead window, and their existence forces an optimal policy to be overly restrained in diverting jobs and hence yield a large average queue length.

To carry out the arguments using the base sample paths, we will exploit a key relationship between *diversions* and *server idling*. In particular, we will demonstrate that, without sufficient lookahead, if a constant fraction of the arrivals are diverted during a specific portion of a base sample path, it will inevitably result in excessive idling of the server not far away in the future, even as $\lambda \to 1$. However, such server idling cannot occur in the heavy-traffic limit, since the server must be fully utilized in order to ensure system stability. This reasoning then implies that any policy that makes such diversions must be *infeasible*, or conversely, that any feasible policy must divert very few arrivals over these segments of the base sample paths (Proposition 6.7). However, such conservatism comes at a cost, in that it leads to long episodes during which the queue length stays at a high level (Proposition 6.9). We then argue that the frequent appearances of such "bad" episodes will result in a large average queue length in steady-state, which contradicts our initial assumption and hence completes the proof of Theorem 6.1.

6.3.1 Preliminaries

Without loss of generality, we will consider only the case where the length of the lookahead window diverges to infinity in the heavy-traffic regime, i.e.,

$$w_\lambda \to \infty, \quad \text{as } \lambda \to 1. \tag{6.11}$$

To see why this is justified, note that because we can always achieve the same average queue length with a longer lookahead window, the optimal average queue length $C^*_{w_\lambda}(p,\lambda)$ must be monototically non-increasing in w_λ. Therefore, any lower bound we obtain on $C^*_{w_\lambda}(p,\lambda)$ under the assumption of Eq. (6.11) also applies to the case where $w_\lambda = \mathcal{O}(1)$. For simplicity of notation, we will drop the dependency on W_λ, and denote by q_λ the optimal average queue length,

$$q_\lambda = C^*_{w_\lambda}(p,\lambda), \quad \forall \lambda \in (0,1). \tag{6.12}$$

Main Assumption. We will assume the validity of the following property throughout the remainder of the proof, which states that it is indeed possible to achieve a small delay whenever w_λ is of order $\Omega\left(\ln\frac{1}{1-\lambda}\right)$. Note that invalidating this assumption will immediately prove the lower bound on $C^*_{w_\lambda}(p,\lambda)$ in Theorem 6.1.

Property 6.2. *Fix* $p \in (0,1)$. *Suppose that* $w_\lambda \succeq \ln\frac{1}{1-\lambda}$, *as* $\lambda \to 1$. *Then,*

$$q_\lambda \ll \ln\frac{1}{1-\lambda}, \quad \textit{as } \lambda \to 1. \tag{6.13}$$

We shall also assume that $w_\lambda \succeq \ln\frac{1}{1-\lambda}$, as $\lambda \to 1$. Under this regime of w_λ, Property 6.2 implies that

$$q_\lambda \ll w_\lambda, \quad \text{as } \lambda \to 1. \tag{6.14}$$

State Representation and Stationary Policies

We now cast our problem as a Markov decision process, and argue that there always exists an *stationary* optimal policy that depends only on the state, which consists of the current queue length and the events during the lookahead window.

Since all diversion decisions are associated with events in $\mathcal{A} \cup \mathcal{S}$, it suffices to specify the nature of future information for the event times, $\{Z_m : m \in \mathbb{N}\}$. At $t = Z_m$, the *content* of the lookahead window is defined to be the vector $F(m) = (F_k(m) : k \in \mathbb{Z}_+)$, where

$$F_k(m) = (Z_{m+k} - Z_m\,,\, R_{m+k}), \quad 0 \le k \le \mathcal{N}(Z_m + w_\lambda) - 1. \tag{6.15}$$

In other words, $F_k(m)$ specifies the time of the kth future event starting from the current time, Z_m, along with its type for all events within the lookahead window of length w_λ. For future events beyond the lookahead window which we have no access to, we simply set the value of $F_k(m)$ to zero:

$$F_k(m) = (0, 0), \quad k \geq \mathcal{N}(Z_m + w_\lambda). \tag{6.16}$$

Note that according to this definition, all entries of $F(m)$ lie within the compact interval of $[-1, w_\lambda]$.

Recall that $Q(t)$ is the queue length at time t. Consider the sequence $\{X(m) : m \in \mathbb{N}\}$, where

$$X(m) = (Q(Z_m -), F(m)). \tag{6.17}$$

From this point on, we will refer to $\{X(m) : m \in \mathbb{N}\}$ as the *states* of our system.

Stationary Policies. We say that a policy π is stationary, if its diversion decision at time Z_m depends only on the state, $X(m)$, or formally, that

$$\mathbb{P}\left(H(m) = 1 \,|\, X(m)\right) = \mathbb{P}\left(H(m) = 1 \,\Big|\, \{(Z_k, R_k)\}_{k=1}^{\mathcal{N}(Z_m + w_\lambda)}\right), \quad \text{a.s.} \tag{6.18}$$

Since the arrivals and service tokens are generated according to Poisson processes, the future evolution of the system starting from $t = Z_m$ conditional on the current state X_m and diversion decision is independent of the past, and our problem can be treated as a Markov decision process (MDP), with

1. The state space of $(\mathbb{Z}_+) \times ([-1, w_\lambda]^{\mathbb{N}})$, endowed with the metric of $\|x - y\| = \sum_{i=1}^{\infty} 2^{-i} |x_i - y_i|$.

2. The action space of $[0, 1]$, where an action specifies the probability of diversion, $\mathbb{P}(H_m = 1)$.

3. The stochastic kernel associated with the arrival and service token processes, as well as the queueing and diversion dynamics.

4. The objective of minimizing an average penalty, given in Eq. (6.10), subject to the average cost constraint, given by Eq. (6.9).

MDPs of this kind have been studied in the literature, and it is known that there exist optimal policies that are *stationary* (c.f. [35, 43]), whose actions depend on the state $X(m)$ only. Therefore, without loss of generality, we will focus on the family of stationary policies for the remainder of the proof of Theorem 6.1.

Given a stationary policy, π, the resulting state sequence $\{X(m) : m \in \mathbb{N}\}$ is a time-homogeneous Markov chain, and admits a unique steady-state distribution. We assume that $Q(0)$ and $X(0)$ are initialized in their respective steady-state distributions. In this case, it is not difficult to show that $\{Q(t) : t \in \mathbb{R}_+\}$ is stationary and ergodic, so that the time-average queue length is equal to the expected queue length in steady-state, i.e.,

$$\mathbb{E}\left(Q(t)\right) = \mathbb{E}\left(Q(0)\right) = C(p, \lambda, \pi), \quad t \in \mathbb{R}_+. \tag{6.19}$$

and, similarly, the sequence of diversion decision $\{H_m : m \in \mathbb{N}\}$ is stationary and ergodic, with

$$\mathbb{E}(H_m) = \mathbb{E}(H_1) = \limsup_{M \to \infty} \frac{\mathbb{E}\left(\sum_{m=1}^{M} H(m)\right)}{M}, \quad \forall m \in \mathbb{N}. \tag{6.20}$$

Define the process $\{L(t) : t \in \mathbb{R}_+\}$, where

$$L(t) = \mathbb{I}\left\{Q(t) \le 2q_\lambda\right\}, \quad t \in \mathbb{R}_+. \tag{6.21}$$

The following lemma follows from the stationarity of $Q(\cdot)$, and applying Markov's inequality to $Q(0)$.

Lemma 6.3. *Fix $p \in (0,1)$. For all $\lambda \in (1-p, 0)$, we have that*

$$\mathbb{E}(L(t)) = \mathbb{P}(Q(0) \le 2q_\lambda) \ge \frac{1}{2}, \quad \forall t \in \mathbb{R}_+, \tag{6.22}$$

under any optimal stationary policy.

In the remainder of the proof, we will show that there exists $c_l > 0$ such that if $w_\lambda \le c_l \ln \frac{1}{1-\lambda}$, then Eq. (6.22) cannot be true under any sequence of optimal stationary policies, unless $Q^*(\lambda, w_\lambda) \gtrsim \ln \frac{1}{1-\lambda}$. This would invalidate Assumption 6.2, hence proving the lower bound on $C^*_{w_\lambda}(p,\lambda)$ in Theorem 6.1.

6.3.2 Base Sample Paths

We now describe a set of base sample paths which will serve as the basis of our subsequent analysis. In later sections, we will show that, roughly speaking, the non-negligible chance of occurrence of such sample paths will "force" any feasible policy to be overly conservative in diverting jobs, should w_λ be too small.

Let $B \in \mathbb{R}_+$ be a quantity whose value will be specified in the sequel. We define the following time markers, whose positions relative to each other are illustrated in Figure 6-2.

$$\begin{aligned} U_1 &= w_\lambda, \\ U_2 &= U_1 + B = w_\lambda + B, \\ U_3 &= U_2 + w_\lambda = 2w_\lambda + B, \end{aligned}$$

The set of base sample paths is defined as the intersection of the events $\mathcal{E}_1$ through $\mathcal{E}_5$, described as follows. Let ϵ, ζ and ϕ be positive constants.

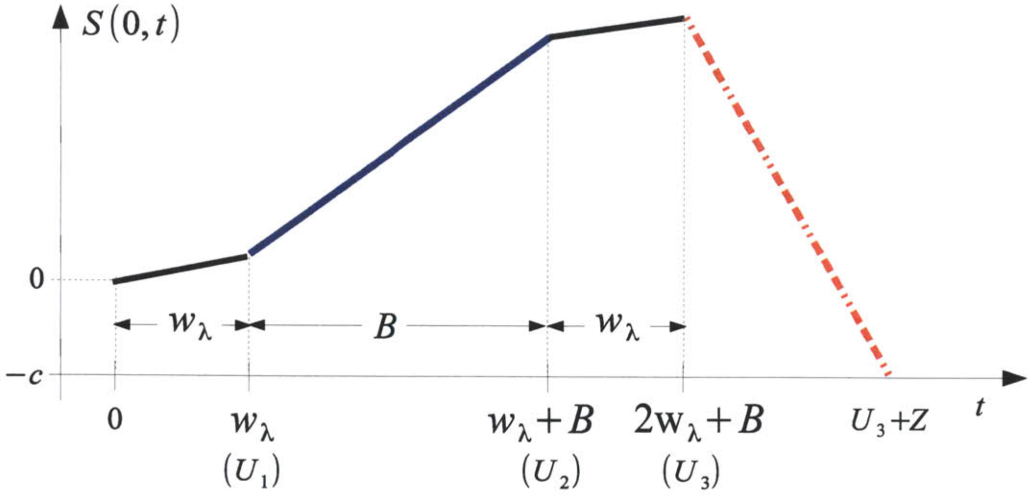

Figure 6-2: This figure illustrates the macroscopic behavior of the base sample paths. The blue segment represents a period of sustained upward drift of $S(0,\cdot)$, and the red segment that of a downward drift. The two black segments, each with length equal to that of the lookahead window, serve as a "buffer," ensuring that the actions of the diversion policy before the segment are *independent* from the evolution of $S(0,\cdot)$ afterwards.

1. Event $\mathcal{E}_1$, parameterized by ϵ and ζ, requires that the sample path of $S(0,\cdot)$ stays close to its expected behavior during the interval $[U_1,U_2)$:

$$\mathcal{E}_1 = \{|S(U_1,t) - [\lambda - (1-p)]t| \le \epsilon t + \zeta, \text{ for all } t \in [U_1,U_2)\}, \tag{6.23}$$

   When ϵ is small, this implies that $S(0,\cdot)$ undergoes a consistent *upward* drift during $[U_1,U_2)$. Event $\mathcal{E}_1$ is illustrated by the blue line segment in Figure 6-2.

2. Event $\mathcal{E}_2$ requires that the queue length at $t=0$ is not too large compared to the optimal average queue length,

$$\mathcal{E}_2 = \{Q(0) \le 6q_\lambda\}. \tag{6.24}$$

3. The events $\mathcal{E}_3$ and $\mathcal{E}_4$ put some restrictions on the amount of upward excursion of $S(0,\cdot)$ during the intervals $[0,U_1)$ and $[U_2,U_3)$, respectively,

$$\mathcal{E}_3 = \{S(0,U_1) \leq 2w_\lambda\}, \tag{6.25}$$

$$\mathcal{E}_4 = \{S(U_2,U_3) \leq 2w_\lambda\}, \tag{6.26}$$

The main purpose of $\mathcal{E}_3$ and $\mathcal{E}_4$ is to serve as "buffers" to induce a certain independence property, which will be useful for subsequent analysis: since the lengths of $[0,U_1)$ and $[U_2,U_3)$ are both equal to that of the lookahead window, the actions of the diversion policy before each interval are *independent* from the evolution of $S(0,\cdot)$ after it. The two events are illustrated by the black line segments in Figure 6-2.

4. Finally, the event $\mathcal{E}_5$ requires that $S(0,\cdot)$ will undergo a substantial *downward* excursion soon after U_3, as is illustrated by the red dotted line segment in Figure 6-2. Let Z be the stopping time

$$Z = \inf\{z \in \mathbb{R}_+ : S(U_3,U_3+z) < -[6q_\lambda + [\lambda - (1-p) - \epsilon]B + \zeta + 4w_\lambda]\}, \tag{6.27}$$

and $\mathcal{E}_5$ is defined by putting an upper bound on Z:

$$\mathcal{E}_5 = \{Z \leq \phi w_\lambda\}. \tag{6.28}$$

The right-hand-side of the inequality in the definition of Z was chosen so that, conditional on the joint occurrence of $\mathcal{E}_1$ through $\mathcal{E}_4$, a downward excursion in $S(0,\cdot)$ of such magnitude is guaranteed to *deplete* the queue by time U_3+Z. As will become clearer in the next section, this depletion will help us connect diversions to future idling of the server.

Note that the events $\mathcal{E}_1$, $\mathcal{E}_3$, $\mathcal{E}_4$, and $\mathcal{E}_5$ concern the input sample path $S(0,\cdot)$ only, and are independent of the diversion policies, while $\mathcal{E}_2$ also depends on the choice of diversion policy.

Having described the events that together characterize the base sample paths, we next illustrate some of their statistical properties. The first lemma shows that the events $\mathcal{E}_1$ through $\mathcal{E}_4$ can occur with fairly high probabilities. The proof is given in Appendix C.1.1.

Lemma 6.4. *1. Fix $\epsilon > 0$. For all $\theta \in (0,1)$, there exists $\zeta > 0$, so that for all $\lambda > 1 - \frac{1}{2}p$,*

$$\inf_{B \geq 0} \mathbb{P}(\mathcal{E}_1) \geq \theta. \tag{6.29}$$

2. Under optimal stationary policies, $\mathbb{P}(\mathcal{E}_2) = \mathbb{P}(Q(0) \leq 6q_\lambda) \geq \frac{5}{6}$, for all $\lambda \in (1-p, 1)$.

3. $\lim_{\lambda \to 1} \mathbb{P}(\mathcal{E}_3) = \lim_{\lambda \to 1} \mathbb{P}(\mathcal{E}_4) = 1$.

The next lemma shows that the event $\mathcal{E}_5$ occurs with a small yet non-negligible probability. The proof is given in Appendix C.1.2.

Lemma 6.5. *Fix $k, \phi, \zeta > 0$, and $\epsilon \in (0, \min\{\zeta, \lambda - (1-p)\})$. Suppose that $B = kw_\lambda$, and $q_\lambda \ll w_\lambda$, as $\lambda \to 1$. There exists $\gamma > 0$, such that*

$$\mathbb{P}(\mathcal{E}_5) \gtrsim \exp(-\gamma w_\lambda), \quad \text{as } \lambda \to 1. \tag{6.30}$$

Finally, the following independence properties among the events will be useful. The proof is given in Appendix C.1.3.

Lemma 6.6. *For a feasible diversion policy, the following hold.*

1. *The events $\mathcal{E}_1, \mathcal{E}_3, \mathcal{E}_4$ and $\mathcal{E}_5$ are independent.*

2. *The event $\mathcal{E}_2$ is independent from $\mathcal{E}_1, \mathcal{E}_4$, and $\mathcal{E}_5$, but not necessarily from $\mathcal{E}_3$.*

3. *Denote by Y the number of diversions in the interval $[U_1, U_2)$, i.e.,*

$$Y = \sum_{\mathcal{N}(U_1) \le m \le \mathcal{N}(U_2)-1} H(m). \tag{6.31}$$

Then, Y is independent of $\mathcal{E}_5$.

6.3.3 From Diversions to Server Idling

The goal of this subsection is to show that, if w_λ is small, then the number of diversions made during the the interval $[U_1, U_2)$, i.e., the random variable Y (Eq. (6.35)), must also be appropriately small, under any optimal stationary policy. To achieve this, we will exploit a connection between Y and the idling of the server at a later time.

The intuition is perhaps best seen pictorially, as depicted in Figure 6-2. Conditional on the occurrence of the events $\mathcal{E}_1$ through $\mathcal{E}_5$, and supposing that *no* diversion has been made, the queue length process $Q(t)$ would have "followed" the trajectory depicted in the figure and reached *zero* by time $U_3 + \phi w_\lambda$. Suppose now that a *large* number of diversions are made during the interval $[U_1, U_2)$ (line segment in blue). Then, the depletion of the queue implies that there must be an extended period of server idling prior to $U_3 + \phi w_\lambda$. Such idling, if it persists even as $\lambda \to 1$, can be problematic and will be shown to contradict the feasibility of the diversion policy. This in turn implies that the number of diversions in $[U_1, U_2)$ must be small.

The next proposition is the main result of this subsection, which formalizes the above intuition. There is however one adjustment: instead of conditioning on all five events, which has vanishingly small probability due to the presence of $\mathcal{E}_5$, we will condition only on $\mathcal{E}_1$ and $\mathcal{E}_2$, which occur with high probability. To do so, we will exploit several independence properties among the events, as in Lemma 6.6, and show that the impact of $S(0,\cdot)$'s downward excursion described by $\mathcal{E}_5$ is unavoidable when w_λ is too small, even without explicit conditioning on $\mathcal{E}_5$.

Proposition 6.7. *Fix $k > 0$, and let $B = kw_\lambda$. There exists $c > 0$, so that if*

$$w_\lambda \leq c \ln \frac{1}{1-\lambda}, \quad \textit{as } \lambda \to 1, \tag{6.32}$$

then for all $\tau > 0$,

$$\lim_{\lambda \to 1} \mathbb{P}\left(Y \geq \tau B \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = 0, \tag{6.33}$$

under any sequence of optimal stationary policies, where Y is the number of diversions during $[U_1, U_2)$, defined in Eq. (6.31).

Proof. We say that a service token generated at time t is *wasted*, if there is currently no job in the queue, i.e., $Q(t) = 0$. Let $\{\mathcal{J}(t) : t \in \mathbb{R}_+\}$ be the counting process of wasted service tokens, i.e.,

$$\mathcal{J}(t) = \#\text{ of wasted service tokens in } [0,t). \tag{6.34}$$

For the sake of contradiction, assume the following is true: if $w_\lambda \gtrsim \ln \frac{1}{1-\lambda}$ as $\lambda \to 1$, then there exist $\tau > 0$, and a sequence of optimal stationary policies, $\{\pi_\lambda\}$, under which

$$\liminf_{\lambda \to 1} \mathbb{P}\left(Y \geq \tau B \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = q > 0. \tag{6.35}$$

The following lemma is a key ingredient of the proof, and states that the number of wasted tokens must be substantial. The proof is based on the intuition explained in the passages above Proposition 6.7, and is given in Appendix C.1.4.

Lemma 6.8. *Fix $k > 0$, and let $B = kw_\lambda$. Suppose Eq.* (6.35) *is true for some sequence of optimal stationary policies, $\{\pi_\lambda\}$. Then, there exist $a, \gamma > 0$ (whose values can depend on k) such that*

$$\mathbb{E}\left(\mathcal{J}(aw_\lambda)\right) \gtrsim w_\lambda \exp\left(-\gamma w_\lambda\right), \tag{6.36}$$

as $\lambda \to 1$, under $\{\pi_\lambda\}$.

Consider an optimal stationary policy. Denote by $\mathcal{H}(t)$ the counting process representing the number of diversions in $[0, t)$, i.e.,

$$\mathcal{H}(t) = \sum_{m=1}^{\mathcal{N}(t)} H(m). \tag{6.37}$$

By the stationarity of $\{H(m) : m \in \mathbb{N}\}$ (Eq. (6.20)) and definition of $\mathcal{N}(t)$ (Eq. (6.5)), it is not difficult to show that, for all $t \geq 0$,

$$\begin{aligned} \frac{\mathbb{E}(\mathcal{H}(t))}{t} &= \frac{1}{t}\mathbb{E}\left(\sum_{m=1}^{\mathcal{N}(t)} H(m)\right) = (\lambda + 1 - p)\mathbb{E}(H(1)) \\ &= \limsup_{M \to \infty} \frac{(\lambda + 1 - p)\mathbb{E}\left(\sum_{m=1}^{M} H(m)\right)}{M}. \end{aligned} \tag{6.38}$$

By definition, we have that

$$Q(t) = Q(0) + S(0, t) + \mathcal{J}(t) - \mathcal{H}(t), \quad \forall t \in \mathbb{R}_+. \tag{6.39}$$

Taking expectation on both sides of the above equation, and letting $t = aw_\lambda$, where

a is given as in Lemma 6.8, we have that

$$
\begin{aligned}
&\frac{\mathbb{E}(\mathcal{H}(aw_\lambda))}{aw_\lambda} - p \\
=&\frac{1}{aw_\lambda}\left(\mathbb{E}\left(S\left(0,aw_\lambda\right)\right) + \mathbb{E}\left(\mathcal{J}(aw_\lambda)\right) + \mathbb{E}(Q(0)) - \mathbb{E}(Q(aw_\lambda))\right) - p \\
\overset{(a)}{=}&[\lambda - (1-p)] - p + \frac{1}{aw_\lambda}\mathbb{E}\left(\mathcal{J}(aw_\lambda)\right) \\
\overset{(b)}{\gtrsim}&(\lambda - 1) + \frac{1}{aw_\lambda}w_\lambda \exp\left(-\gamma w_\lambda\right) \\
\gtrsim&\exp\left(-\gamma w_\lambda\right) - (1-\lambda),
\end{aligned} \tag{6.40}
$$

where γ is given in Lemma 6.8. Step (a) follows from the fact that $\mathbb{E}(Q(0)) = \mathbb{E}(Q(aw_\lambda))$ by the stationarity of $Q(\cdot)$, and (b) from Eq. (6.36).

Letting $w_\lambda = c \ln \frac{1}{1-\lambda}$, with $c = 1/2\gamma$, we have that

$$
\exp\left(-\gamma w_\lambda\right) \gtrsim \sqrt{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{6.41}
$$

Combining Eqs. (6.40) and (6.41), we have that

$$
\frac{\mathbb{E}(\mathcal{H}(aw_\lambda))}{aw_\lambda} - p \gtrsim \sqrt{1-\lambda} - (1-\lambda) \gtrsim \sqrt{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{6.42}
$$

In particular, this implies that there exists $\lambda' \in (1-p, 1)$, such that

$$
\frac{\mathbb{E}(\mathcal{H}(aw_\lambda))}{aw_\lambda} > p, \quad \forall \lambda \in (\lambda', 1). \tag{6.43}
$$

Since the stationary diversion policies we consider are feasible, we must have that

$$
\frac{\mathbb{E}(\mathcal{H}(t))}{t} \overset{(a)}{=} \limsup_{M \to \infty} \frac{(\lambda + 1 - p)\mathbb{E}\left(\sum_{m=1}^{M} H(m)\right)}{M} \overset{(b)}{\leq} p, \tag{6.44}
$$

for all $\lambda \in (1-p, 1)$, and $t > 0$, where (a) and (b) follow from Eqs. (6.38) and (6.9), respectively. This leads to a contradiction with Eq. (6.43), which invalidates the assumption made in Eq. (6.35), and hence proves Proposition 6.7. □

6.3.4 Consequences of Too Few Diversions

Proposition 6.7 tells us that, under optimal stationary policies, the number of diversions in $[U_1, U_2)$ must be small when w_λ is small. Building on this observation, we now focus on policies that divert "very few" jobs during $[U_1, U_2)$, i.e., with Y scaling sub-linearly with respect to B, and show that they will necessarily lead to a large expected queue length in steady-state. The following proposition is the main result of this subsection.

Proposition 6.9. *Fix $p \in (0,1)$. There exists $c_l > 0$, so that if*

$$w_\lambda \leq c_l \ln \frac{1}{1-\lambda}, \quad \textit{as } \lambda \to 1, \tag{6.45}$$

then

$$\limsup_{\lambda \to 1} \mathbb{E}\left(L(0)\right) \leq \frac{1}{3}. \tag{6.46}$$

under any sequence of optimal stationary policies.

Proof. We will assume that $B = kw_\lambda$, with $k = 24$, and that $w_\lambda \leq c_l \ln \frac{1}{1-\lambda}$, where c_l is equal to the constant c in Proposition 6.7 for the corresponding value of k.

Consider an optimal stationary policy, with a resultant average queue length of q_λ. We will prove the claim by showing that if $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs *and* the number of diversions made in $[U_1, U_2)$ is small (cf. Eq. (6.33)), then, for a "long time" after U_1, the queue length will stay at a high level (i.e., $Q(t) > 2q_\lambda$). Recall that Y is the number of diversions made during the period $[U_1, U_2)$. We have the following inequality, derived from the queueing dynamics:

$$Q(t) \geq Q(U_1) + S(U_1, t) - Y, \quad \forall t \in [U_1, U_2), \tag{6.47}$$

where the equality holds if $Q(t) > 0$ for all $t \in [U_1, U_2)$. By the definition of $\mathcal{E}_1$ (Eq. (6.23)), Eq. (6.47), and the fact that $Q(U_1) \geq 0$, we have that

$$\mathbb{P}\left(Q(t) \geq [\lambda - (1-p) - \epsilon]t - \zeta - Y \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = 1, \quad \forall t \in [U_1, U_2). \tag{6.48}$$

Let V be the *last* time in $[U_1, U_2)$ when the queue length is less than $2q_\lambda$,

$$V = \sup\left\{t \in [0, B) : Q(U_1 + t) \leq 2q_\lambda\right\}. \tag{6.49}$$

Applying the definition of V in the context of Eq. (6.48) yields that

$$\mathbb{P}\left(V \leq \frac{1}{\lambda - (1-p) - \epsilon}\left(2q_\lambda + Y + \zeta + 1\right) \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = 1. \tag{6.50}$$

Recall from Proposition 6.7 that, conditional on $\mathcal{E}_1 \cap \mathcal{E}_2$ and assuming $w_\lambda \leq c_l \ln \frac{1}{1-\lambda}$, Y must be sub-linear in $B = kw_\lambda$. In particular, by Eq. (6.33), we have that, for all $\tau > 0$,

$$\lim_{\lambda \to 1} \mathbb{P}\left(Y \leq \tau k w_\lambda \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = 1. \tag{6.51}$$

Combining Eqs. (6.50) and (6.51), and the fact that $w_\lambda \to \infty$ as $\lambda \to 1$, we have that, there exists $v > 0$, such that for all $\tau > 0$,

$$\mathbb{P}\left(V \leq v q_\lambda + \tau k w_\lambda \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) = 1 - \delta(\lambda), \quad \forall \lambda \in (1-p, 1), \tag{6.52}$$

where $\delta(\cdot)$ is a function with $\lim_{x\to 1}\delta(x) = 0$. In other words, conditional on $\mathcal{E}_1 \cap \mathcal{E}_2$, $Q(t)$ will reach the level of $2q_\lambda$ soon after U_1, with high probability. Translating this into the expected value of V, we have that

$$\begin{aligned}
&\limsup_{\lambda\to 1} \frac{1}{U_2}\mathbb{E}(V) \\
\le &\limsup_{\lambda\to 1} \frac{1}{U_2}\big(\mathbb{E}(V \,|\, \mathcal{E}_1 \cap \mathcal{E}_2)\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) + U_2(1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2))\big) \\
\overset{(a)}{\le} &(1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) + \limsup_{\lambda\to 1} \frac{\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)}{U_2}\left(vq_\lambda + \tau k w_\lambda + U_2\delta(\lambda)\right) \\
\overset{(b)}{=} &(1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) + \frac{kw_\lambda}{U_2}\tau\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \\
\overset{(c)}{=} &(1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) + \frac{k}{k+1}\tau\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \\
\le &(1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) + \tau\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)
\end{aligned} \tag{6.53}$$

where (a) follows from Eq. (6.52), and (b) from the assumptions that $q_\lambda \ll w_\lambda$ and $\lim_{\lambda\to 1}\delta(\lambda) = 0$, and (c) from the fact that $U_2 = B + w_\lambda = (k+1)w_\lambda$. We now connect the behavior of $\mathbb{E}(V)$ to that of $\mathbb{E}(L(0)) = \mathbb{P}(Q(0) \le 2q_\lambda)$, as follows.

$$\begin{aligned}
\limsup_{\lambda\to 1}\mathbb{E}\left(L(0)\right) &\overset{(a)}{=} \limsup_{\lambda\to 1}\mathbb{E}\left(\frac{1}{U_2}\int_{t=0}^{U_2} L(t)dt\right) \\
&\overset{(b)}{=} \limsup_{\lambda\to 1}\mathbb{E}\left(\frac{1}{U_2}\int_{t=0}^{U1+V} L(t)dt\right) \\
&\overset{(c)}{\le} \limsup_{\lambda\to 1}\mathbb{E}\left(\frac{U_1+V}{U_2}\right) = \limsup_{\lambda\to 1}\frac{U_1+\mathbb{E}(V)}{U_2} \\
&\overset{(d)}{=} \frac{w_\lambda}{(k+1)w_\lambda} + \limsup_{\lambda\to 1}\frac{\mathbb{E}(V)}{U_2} \\
&\overset{(e)}{\le} \frac{1}{k+1} + \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)\tau + (1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) \\
&\le \frac{1}{k} + \tau + (1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2))
\end{aligned} \tag{6.54}$$

where step (a) follows from the stationarity of $Q(\cdot)$ and hence that of $L(\cdot)$. Step (b) follows from the that $L(t) = 0$, for all $t \in [U_1 + V, U_2)$, which is a consequence of the definition of V in Eq. (6.49). Step (c) is based on the fact that $L(t) \leq 1$, a.s. Steps (d) and (e) follow from the fact that $B = kw_\lambda$, and Eq. (6.53), respectively.

By Claim 3 of Lemma 6.4, and Claim 1 of Lemma 6.6, we have that

$$\liminf_{\lambda \to 1} \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) = \liminf_{\lambda \to 1} \mathbb{P}(\mathcal{E}_1) \mathbb{P}(\mathcal{E}_2) \geq \frac{5}{6}\theta, \tag{6.55}$$

where θ is given in Eq. (6.29). Set $\tau = k = 24$, and let ζ be sufficiently large so that $\theta \geq 10/9$. We have that

$$\limsup_{\lambda \to 1} (1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) \leq 1 - \frac{5}{6} \cdot \frac{9}{10} = 1/4. \tag{6.56}$$

From Eq. (6.54), we have that

$$\limsup_{\lambda \to 1} \mathbb{E}(L(0)) \leq \frac{1}{k} + \tau + (1 - \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2)) \leq \frac{1}{24} + \frac{1}{24} + \frac{1}{4} = \frac{1}{3}, \tag{6.57}$$

which completes the proof of Proposition 6.9. □

6.3.5 Proof of Theorem 6.1

We now complete the proof of Theorem 6.1. Assuming the validity of Property 6.2, Proposition 6.9 asserts that there exists $c_l > 0$, so that if $w_\lambda \leq c_l \ln \frac{1}{1-\lambda}$ as $\lambda \to 1$, we must have that $\limsup_{\lambda \to 1} \mathbb{E}(L(0)) \leq 1/3$ under any sequence of optimal stationary policies. However, this contradicts the requirement that $\mathbb{E}(L(0)) \geq 1/2$, given in Eq. (6.22), which holds independently of the validity of Property 6.2. Therefore, we conclude that Property 6.2 must be invalid.

The invalidity of Property 6.2 establishes the lower bound in Eq. (6.2). In particular, we have that, if $w_\lambda \leq c_l \ln \frac{1}{1-\lambda}$, as $\lambda \to 1$, then

$$C^*_{w_\lambda}(p,\lambda) \succcurlyeq \ln \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{6.58}$$

Finally, we show that this lower bound is achievable, i.e., that

$$C^*_{w_\lambda}(p,\lambda) \preccurlyeq \ln \left(\frac{1}{1-\lambda}\right), \quad \text{as } \lambda \to 1, \tag{6.59}$$

when $w_\lambda \leq c_l \ln \frac{1}{1-\lambda}$. To this end, we invoke Theorem 5.8 in Section 5.3.1, which shows that a deterministic queue-length-based diversion policy can achieve the scaling of Eq. (6.59), even when $w_\lambda = 0$.[3] This completes the proof of Theorem 6.1. □

6.4 Summary and Future Research

For a certain class of queueing admission control problems, we showed that a non-trivial amount of future information is *necessary* in order to achieve superior heavy-traffic delay performance compared to an online policy. Our proof exploited certain excursion properties of a transient random walk, which allowed us to connect a policy's diversion decisions to subsequent system idling. Because this line of argument relies mostly on the macroscopic properties of the input sample paths, our techniques and resulting insights seem to be fairly robust and can potentially be generalized to, for example, a setting where the arrival and service token processes are non-Poisson.

[3] As is described in the proof Theorem 5.8, the scaling in Eq. (6.59) can be achieved by the following simple threshold policy: divert the arrival if and only if the current queue length is equal to a threshold value x, where x is set to be the smallest value such that the resulting diversion rate is no more than p. Since the queue length process under this policy is simply a birth-death process truncated at state x, it is easy to verify, via a direct calculation of steady-state probabilities of $Q(t)$, that $q_\lambda \sim \ln \frac{1}{1-\lambda}$, as $\lambda \to 1$.

In light of the upper and lower bounds on future information provided by Theorems 5.13 and 6.1, respectively, an immediate open question is whether the constants c_h and c_l in the scaling of w_λ *coincide*. The granularity of our proof technique does not appear to be sufficient to answer this question, which likely demands a finer analysis.

Chapter 7

Decentralized Partial Resource Pooling

In the last two chapters, we have seen that having additional future information can significantly improve delay performance in flexible systems. In this chapter, we shall continue the investigation of information, but shift our attention from the axis of *time* to that of *space*. In particular, we would like to know if it is possible to design efficient scheduling policies in a partially flexible system, which use only *local* information, and achieve performance that is on par with an *optimal* centralized scheme that relies on complete information sharing across *all* parts of the system. Our investigation will be carried out within the family of Partial Pooling flexible systems, described in Section 2.2, where a fraction p of the system's total processing resources (servers) are *fully flexible* and are able to serve jobs of all types, while the remaining processing resources are *dedicated* and can only serve a specific job type (Figure 7-1).

The Partial Pooling model was first proposed and analyzed in prior work of the author [84, 91], where we showed that even a small amount of *flexibility* or *resource pooling* can bring significant performance benefits. Specifically, in the limit as the size of the system, n, tends to infinity, when no flexibility is available and all servers are dedicated, corresponding to the case of $p = 0$, the steady-state average queue length across the system scales as

$$\mathbb{E}(Q) \sim \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{7.1}$$

In sharp contrast, when a fraction of the servers is fully flexible, corresponding to the case of $p > 0$, the average queue length, under an *optimal* scheduling policy used by the flexible server pool, scales as

$$\mathbb{E}(Q) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \text{as } \lambda \to 1. \tag{7.2}$$

This demonstrates that even a small amount of resource pooling can lead to *exponential* improvement in the system's delay scaling in the heavy-traffic regime.

However, in order for the superior delay improvement in Eq. (7.2) to be harnessed, the resource pooling architecture proposed by [84] requires a non-trivial amount of *state information* about all parts of the system. In particular, the scheduling policies adopted by the flexible servers in Figure 7-1 is that of *longest-queue-first* (LQF): upon the completion of a previous job, a flexible server has to fetch a new job from one of the longest queues in the *whole* system. As the size of the system grows large, obtaining real-time system-wide state information can become increasingly expensive and difficult, if not entirely impossible.

The main contribution of the current chapter is to show that efficient resource pooling can be achieved in *a decentralized manner*, in a system with a constant

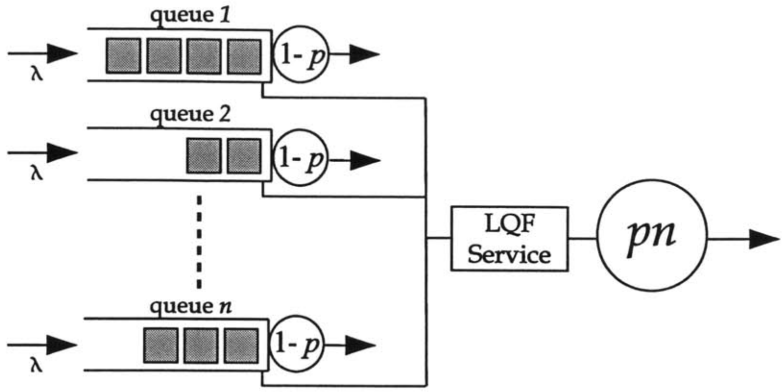

Figure 7-1: Partially centralized resource pooling architecture, from [84, 91].

fraction of flexible servers. Specifically,

1. We construct a class of *decentralized* scheduling policies, where the scheduling decision for a job arriving to queue queue i depends only on the length of queue i at the time of the job's arrival. We show that the *optimal delay scaling* in Eq. (7.2) can be achieved under this decentralized architecture. In a nutshell, instead of having the flexible servers "pull" jobs from the longest queues, the decentralized policies work by "pushing" (or *diverting*) arriving jobs to the pool of flexible servers if and only if the length of the local queue exceeds a certain threshold, hence eliminating the need of global state information when making scheduling decisions.

2. To analyze the delay under decentralized policies, the core of our argument rests upon a "Merging theorem," which yields an exponential tail bound on the steady-state queue length distribution of a queue, whose arrival process consists of the *superposition* of n independent sub-streams, in the limit when the number of sub-streams and the service capacity of the queue tend to infinity simultaneously (Theorem 7.2). This result is then used to show that the

queueing delay experienced by jobs that are sent to the flexible server pool is vanishingly small, in the limit as the system size tends to infinity.

It is well known that, under appropriate scaling, the superposition of n independent "well-behaved" stationary point processes converges to a Poisson process *locally* (i.e., over finite-length time intervals), as $n \to \infty$. Therefore, one may expect that the resulting queue length process, induced by sending the merged arrival process into a queue, should also resemble the one induced by an Poisson arrival process. However, the local convergence of the arrival process to a Poisson process does not automatically carry over to the behavior of the resulting queue length process, and it also says little about what happens to the queue length in steady-state. The Merging theorem essentially makes the above intuition rigorous, and it does so by characterizing the resulting queue length distribution in *steady-state*.

3. In addition to achieving decentralized scheduling, the framework and analysis developed here will allow us to generalize the results of [84, 91] along several directions, for instance, to handling non-exponential arrival and service time distributions (phase-type), as well as analyzing systems with non-uniform arrival rates.

4. Finally, the results in this chapter can be used to formally establish the equivalence between the admission control model studied in Chapters 5 and 6 with that of the Partial Pooling model. As a result, essentially all policies and performance guarantees (with future information or without) provided in the proceeding chapters apply to the flexible architectures in the Partial Pooling family as well.

Organization The remainder of the chapter is organized as follows. In Section 7.1, we state our main result on decentralized resource pooling, Theorem 7.1. We also give a proof of the result, while assuming the validity of a key property concerning the asymptotic expected length of the central queue. This property turns out to be the special case of a more general phenomenon concerning a queue fed with superposition of finite-state Markov-modulated arrival sub-processes, which we shall state as a "Merging theorem" in Section 7.3 (Theorem 7.2), whose modeling setup is provided in Section 7.2. We give two additional applications of the Merging theorem, in which we generalize the Partial Pooling model to non-Poisson arrival and service processes (Section 7.3.2), and rigorously establish a connection between the admission control problem with future information studied in Chapters 5 and 6 to that of Partial Pooling (Section 7.3.3). Section 7.4 reviews some prior research that is related to the Merging theorem. The rest of the chapter is devoted to the proof of of the Merging theorem, with a proof outline given in Section 7.5.

7.1 Decentralized Optimal Pooling

We state in this section our main result concerning decentralized resource pooling architectures. Before we do so, we shall first review the modeling assumptions for flexible architectures in the Partial Pooling family, depicted in Figure 7-1.

The system consists of n *local queues*, where each queue receives incoming jobs at rate λ according to an Poisson process. There are in total n units of processing resources, which take the form of either flexible or dedicated servers, both modeled using service tokens (cf. Section 2.1), as follows:

1. A fraction p of the total processing resources are *fully flexible*, and forms the **flexible server pool**. The server pool is modeled by a Poisson process that

generates **flexible service tokens** at rate pn. When a flexible service token is generated, it can be used to "serve" a job in any of the n queues, making it depart from the queue immediately. If all queues are empty, the service token will be "wasted".

2. The remaining fraction $1 - p$ of the resources are *dedicated* or *inflexible*. The dedicated resources are modeled by n *local servers* attached to the local queues. Each local server is modeled by a Poisson process that generates **inflexible service tokens** at rate $1-p$, and the service tokens generated by the ith local server can only be used to serve a job currently waiting in queue i, making the job depart immediately. If queue i is empty, the service token will be wasted.

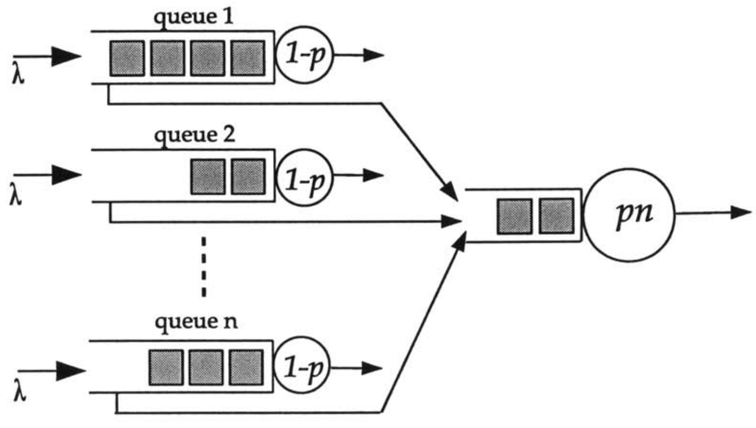

Figure 7-2: Modified Partial Pooling architecture with a central queue.

Under this modeling setup, **scheduling** refers to the allocation of service tokens. For an inflexible token, the task is straightforward, because the token can only be used to serve a job from the corresponding local queue. For a flexible token, however, the decision is not so simple, because there are now up to n local queues to choose from: which queue needs the token the most?

Centralized Longest-Queue-First Policy An intuitive scheduling policy is the so-called longest-queue-first (LQF) policy, where a flexible service token is always used to serve a job from one of the longest queues in the system. The LQF policy in [84, 91] achieves the $\log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ heavy-traffic delay scaling (see Eq. (7.2)). Moreover, it was shown in [84, 91] that the LQF policy is also *optimal* in a very strong sense: the resulting queue length processes under the LQF policy are stochastically dominated by (i.e., "smaller than") those induced by any other causal scheduling policy (which does not uses any future information).

While the longest-queue-first policy yields optimal performance, one of its major drawbacks lies in the demanding **information requirement**: each allocation decision for flexible service tokens involves knowing, in real-time, which queues are currently the longest. This requires the flexible servers to be constantly communicating with *all* local queues in real-time, which can become overly expensive or exceedingly difficult to implement, as the system size n grows large. This begs the question: can we avoid the hurdle posed by a growing need for information, and still achieve performance comparable to that of the LQF policy?

7.1.1 Main Result: Decentralized Resource Pooling

Our main result shows that it is indeed possible to circumvent the informational challenges: there exists a *decentralized* scheduling policy that uses only "local" queue length information, and yet still achieves the same *optimal* heavy-traffic delay scaling as that of the LQF policy.

Before we explain how this can be done, however, the notion of "decentralization" needs to be better defined. To do so, we shall consider a slightly modified Partial Pooling architecture, depicted in Figure 7-2, with the following features.

1. A **central queue** will be attached to the flexible server pool, whose length at time t is denoted by $Q^n(t)$. All flexible service tokens generated in the server pool will be used exclusively to serve jobs that are currently in the central queue.

2. A job in a local queue can be *diverted* to the central queue at *any time*, starting from the time of its arrival to the system, as long as it has not yet been served by an inflexible service token produced by the local server.

Under this modified Partial Pooling model, the scheduling policy becomes a *diversion policy*, which is now concerned with determining whether, and when, a job should be diverted to the central queue. We say that a scheduling policy, ϕ, is **decentralized**, if the decision of diverting a job at queue i at time t depends only on the value of $Q_i(t)$ (the length of the ith queue), and *centralized*, if it depends on the lengths of all n queues at the time.[1]

It is important to note that switching to the modified Partial Pooling model does *not* alter our model significantly: as far as the *total number of jobs in system* is concerned, which determines our main metric of average delay, the modified Partial Pooling model with a central queue (Figure 7-2) is actually *equivalent* to the original model (Figure 7-1):

1. The modified model is able to *simulate* the dynamics of the original model by simply ignoring the existence of the central queue. To do so, we can design the diverting policy in a way, so that a job from queue i is diverted to the central

[1] Because the system is Markovian, it is not difficult to show that the value of $Q_i(s)$ for $s < t$ is not helpful once $Q_i(t)$ is known.

queue at time t under the modified model, if any only if a flexible token is produced at time t, and would have been allocated to serve a job from queue i. In other words, the central queue is never used.

2. Conversely, the original model can also simulate the modified model, given any diversion policy. To do so, we shall "mark" a job, whenever a diversion would have been made in the modified model. The scheduling policy in the original model then simply allocates a flexible service token to any job in system that is currently "marked." Effectively, the "marking" simulates the membership at the central queue, without physically moving the jobs from their corresponding local queues.

We now state the main result of this chapter, which establishes the existence of a decentralized scheduling (diversion) policy for the modified Partial Pooling model, which, as $n \to \infty$, achieves the optimal heavy-traffic delay scaling, as in Eq. (7.2).

Theorem 7.1 (Optimal Decentralized Resource Pooling). *Fix $p \in (0,1)$. For every $\lambda \in (1-p, 1)$ and $n \in \mathbb{N}$, there exists a decentralized scheduling policy, so that*

$$\mathbb{E}(\overline{Q^n}) \leq \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} + 2 + c_\lambda/n, \tag{7.3}$$

where c is a positive constant that depends on λ, but not on n, and $\overline{Q^n}$ is the steady-state normalized total queue length, i.e.,

$$\overline{Q^n} = \frac{1}{n}\left(Q^n + \sum_{i=1}^{n} Q_i\right), \tag{7.4}$$

where the Q_i's and Q^n are the steady-state queue lengths for the local and central queues, respectively. In particular, Eq. (7.3), together with the lower bound given by *the optimal scaling in Eq.* (7.2), *implies that*

$$\limsup_{n\to\infty} \mathbb{E}(\overline{Q^n}) \sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}, \quad \textit{as } \lambda \to 1. \tag{7.5}$$

In addition to the advantage of having a decentralized scheduling policy, Theorem 7.1 also provides a quantitative strengthening of the results [84, 91], by showing that the *speed of convergence* of $\mathbb{E}(\overline{Q^n})$ is of order $\mathcal{O}(1/n)$. In [84, 91], the analysis of the LQF policy in the limit of $n \to \infty$ was carried out using a fluid model and weak convergence methods, and no rate of convergence is provided.[2]

It is important to note that the centralized LQF policy remains optimal in the modified Partial Pooling model. The decentralized scheduling policy used in Theorem 7.1 is in fact sub-optimal for a *finite-sized* system (small n), or when the system is under loaded (λ bounded away from 1). However, Theorem 7.1 ensures that this sub-optimality diminishes as $n \to \infty$ and $\lambda \to 1$ (in this order).

7.1.2 Proof of Theorem 7.1 - Part I

We now give a "partial" proof to Theorem 7.1. The proof is partial, in that we shall postpone the proof of one important claim, Condition (*b*), till Section 7.3.1, where our main technical result, the "Merging theorem," will have been introduced. Leaving this claim aside, the rest of our proof is self-contained, and captures the main intuition on how the decentralized scheduling policy is designed.

We shall find a decentralized scheduling policy that satisfies the following two conditions.

[2] Technically speaking, the convergence results of [84, 91] were stated in an almost-sure sense, and do not directly imply the convergence in expectation. However, extending them to the convergence of expectation is not difficult.

Condition (a). The steady-state queue length at local queue i, $\mathbb{E}(Q_i)$, satisfies

$$\mathbb{E}(Q_i) \le \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} + 2, \tag{7.6}$$

for all $i \in \{1, \dots, n\}$.

Condition (b). The steady-state queue length at the central queue, $\mathbb{E}(Q^n)$, stays *bounded* by some constant, c_λ, as $n \to \infty$, i.e.,

$$\limsup_{n\to\infty} \mathbb{E}(Q^n) \le c_\lambda. \tag{7.7}$$

Note that, because

$$\mathbb{E}(\overline{Q^n}) = \frac{1}{n}\left(\mathbb{E}(Q^n) + \sum_{i=1}^{n} \mathbb{E}(Q_i)\right), \tag{7.8}$$

the validity of both conditions will directly imply that of Eq. (7.3), hence proving Theorem 7.1.

For Condition (a), we will use a simple *threshold rule* as our decentralized scheduling policy: a job arriving to queue i at time t will be *immediately* diverted to the central queue if

$$Q_i(t-) \ge \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil + 1, \tag{7.9}$$

otherwise, the job will join queue i and ultimately be served by an inflexible service token from the local server. The resulting scheduling policy is clearly decentralized by definition. It also trivially guarantees the validity of Condition (a), because the length of queue i under this policy *never* exceeds $\left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil + 1$.

We would like to argue that Condition (b) also holds for this threshold-hold based scheduling policy, that is, after feeding the n streams of job diversions into the central queue, the resulting steady-state queue length Q^n maintains a bounded expected value as $n \to \infty$. A simple calculation shows that the total rate at which jobs are sent to the central queue is less than pn, guaranteeing stability. However, the boundedness of $\mathbb{E}(Q^n)$ is no longer so straightforward, because the process of job diversions under the threshold rule is no longer Poisson. To to show that Condition (b) is indeed true for the threshold-based policy, and in fact, for a more general class of diversion policies, we will need to establish a technical result, which we refer to as the "Merging theorem." The validity of Condition (b) will be shown in Section 7.3.1.

7.2 Model for Merging Thoerem

We describe in this section the queueing model for our main technical result, the Merging theorem (Theorem 7.2). The setup is fairly reminiscent of, and motivated by, the Partial Pooling architecture under a decentralized threshold-based scheduling policy, as described in Section 7.1.2. However, the arrival processes considered in the Merging theorem is a bit more general.

We consider a sequence of systems indexed by the *system size*, $n \in \mathbb{N}$, illustrated in Figure 7-3. The nth system contains an infinite buffer, or queue, whose length at time t is denoted by $Q^n(t)$. When the context is clear we shall omit the dependency on n and write $Q(t)$ instead.

The queue is served by a single server running at speed n. We shall use the service token model to describe service dynamics (see Section 2.1). Let $\{S^n(t) : t \in \mathbb{R}_+\}$ be the counting process associated with the service token generation in the nth system. We assume that S^n is a time-homogeneous Poisson process with rate n. When an event occurs in S^n at time t, we say that a service token is produced, and the length

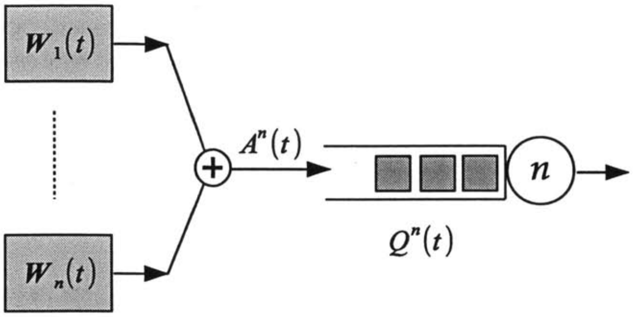

Figure 7-3: Queueing model for Theorem 7.2. The arrival process in the nth system, A^n, is the superposition of n independent Markov modulated Poisson processes (MMPP), each modulated by a finite-state Markov process, W_i.

of Q is decreased by 1 if $Q(t-) > 0$, and remains unchanged otherwise.

Arrivals to the queue in the nth system is represented by a counting process $\{A^n(t) : t \in \mathbb{R}_+\}$. The process A^n is the superposition of n independent Markov-modulated Poisson processes (MMPP):

$$A^n(t) = \sum_{i=1}^{n} A_i(t), \quad \forall t \in \mathbb{R}_+. \tag{7.10}$$

where each A_i is modulated by an independent Markov process with identical transition dynamics. For this reason, we will refer to A^n as the *merged arrival process.*

The underlying modulating Markov process for A_i takes values in a finite state space, $\{1, \ldots, M\}$, whose state at time t is denoted by $W_i(t)$. We assume that the W_is are uniformized, with a *transition rate* of ξ across all states, and that P is the transition matrix for the embedded discrete Markov chain. We will denote by $\pi = (\pi_1, \ldots, \pi_M)$ the steady-state distribution of W_1, where π_w is the probability of the process being at state w. Without loss of generality, we assume that the steady-state probabilities are non-zero, i.e.,

$$\min_{1\le w\le M} \pi_w = \tilde{\pi} > 0. \tag{7.11}$$

A finite *modulating rate* r_w is associated with each state $w \in \{1,\dots,M\}$, so that at time t, A_i generates arrivals at rate r_w if $W_i(t)$ is equal to w. We assume that the modulating rate of W_1 in steady-state is less than 1, i.e.,

$$\sum_{w=1}^{M} \pi_w r_w = \rho < 1. \tag{7.12}$$

This is an important, and necessary, assumption. Note that the long-term time-average rate of arrivals from A^n is equal to $n\rho$, which is less than the server speed, n, if any only if $\rho < 1$.

Finally, we will let $W(t)$ be the vector consisting of the states of all modulating Markov processes,

$$W(t) = (W_1(t),\dots,W_n(t)), \quad t \in \mathbb{R}_+. \tag{7.13}$$

With this representation, it is not difficult to check that the evolution of the system is *Markovian* with respect to the vector

$$\mathbf{X}(t) = (Q(t), W(t)), \quad \forall t \in \mathbb{R}_+. \tag{7.14}$$

7.3 Merging Theorem and Applications

We state below the main technical result of this chapter, the Merging theorem. We then apply it to devise optimal decentralized scheduling policies for partially flexible resource pooling.

Theorem 7.2 (Merging Theorem). *Consider the systems described in Section 7.2 and Figure 7-3. Denote by Q^n the steady-state queue length distribution in the nth system. For any $\rho \in (0,1)$, there exists $\theta > 0$, so that, for all $n \in \mathbb{N}$,*

$$\mathbb{P}(Q^n \geq x) \leq \exp(-\theta x), \quad \forall x \in \mathbb{N}. \tag{7.15}$$

As an immediate corollary of Theorem 7.2, we have that, for all $\rho \in (0,1)$, the steady-state expected queue length remains *bounded*, as $n \to \infty$.

Corollary 7.3. *Fix $\rho \in (0,1)$. We have that*

$$\limsup_{n\to\infty} \mathbb{E}(Q^n) < \infty. \tag{7.16}$$

7.3.1 Proof of Theorem 7.1 - Part II

With Theorem 7.2 at hand, we are now ready to complete the proof of Theorem 7.1 in Section 7.1.2, by filling in the missing piece, the validity of Condition (b).

Fix $p \in (0,1)$. Define

$$L(\lambda) = \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil, \quad \lambda \in (1-p, 1). \tag{7.17}$$

Recall from Section 7.1.2, that we will be emplying a threshold-based scheduling policy, so that a job arriving to queue i is diverted to the central queue if any only if the length of queue i is at $L(\lambda) + 1$. Condition (b) states that, under this policy, the resulting length of the central queue in steady-state satisfies

$$\limsup_{n\to\infty} \mathbb{E}(Q^n) \leq c_\lambda, \tag{7.18}$$

where c_λ is a positive constant that depends on p and λ, but is independent from n.

We will invoke Theorem 7.2 to show that Eq. (7.18) holds. Because the arrivals to the central queue is the superposition of all diverted jobs from the n local queues, it suffices for us to show that each stream of diverted jobs is a finite-state MMPP. Moreover, we will have to show that the steady-state modulating rate is strictly less than p, so that the traffic load at the central server, which has capacity pn, is bounded away from 1.

It is not difficult to verify that the point process corresponding to the diversions from each local queue is an MMPP with a *finite* number of underlying states. In particular, the state space of the modulating Markov chain corresponds to the set of all possible values that a local queue may take on under the threshold policy, that is, $\{0, 1, \ldots, L(\lambda) + 1\}$. Because an arrival to queue i is diverted to the central queue with probability 1 when Q_i is in state $L(\lambda) + 1$, and none is diverted otherwise, in the terminology of Section 7.2, the state-dependent modulating rates of the MMPP for the diversion process at queue i are given by

$$r_w = \begin{cases} \lambda, & w = L(\lambda) + 1, \\ 0, & \text{o.w.} \end{cases} \tag{7.19}$$

It remains to verify that the steady-state modulating rate of this MMPP is strictly less than p. Note that because the flexible server pool operates at speed pn, this is equivalent to the requirement that ρ be strictly less than 1 in Theorem 7.2. (Note that ρ depends on λ, but is independent of n.) To this end, we note that the threshold policy used at each local queue is identical to the L-threshold policies used in the admission control problem of Chapter 5. Therefore, we invoke Theorem 5.8 in Section 5.3.1, which states that the L-threshold policy with $L = L(\lambda) = \left\lceil \log_{\frac{\lambda}{1-p}} \frac{p}{1-\lambda} \right\rceil$ results in a steady-state rate of diversion that is *at most* p, i.e., feasible for the admission control problem. It is not difficult to show that the diversion rate is strictly decreasing as a function of the threshold, and therefore, increasing the threshold by 1 from $L(\lambda)$ ensures that the steady-state rate of diversion from each queue is strictly less than p. We have hence verified all conditions for Theorem 7.2 to be applicable. This validates Eq. (7.18), which, in turn, proves Theorem 7.1.

7.3.2 Generalizations of Partial Pooling

The Merging theorem, combined with the use of decentralized scheduling policies, also allows for analyzing several generalizations of the Partial Pooling architecture, which would have been very difficult to do using the fluid-model framework adopted in [84, 91]. At a high level, the main benefit comes from the fact that, under a decentralized scheduling policy, the dynamics of the n local queues become *independent* from each other. As a result, if the expected queue length of the central queue can be shown to stay bounded as the system size grows using the Merging theorem, then the delay experienced by the jobs being routed to the central queue is negligible. We have hence reduced the problem of analyzing the dynamics of n (potentially coupled) queues, in the case of a centralized policy, to that of a *single* queue that runs independently from the rest of the system.

For instance, consider the generalization where the arrival process, as well as the generation of service tokens at the local server, are both finite-state MMPPs (i.e., *phase-type processes*). The fluid model of [84, 91] can no longer be used to analyze the behavior of the LQF policy, because the lengths of the local queues are not Markovian under the phase-type arrival and service processes. On the other hand, the threshold-based scheduling policy described in Section 7.1.2 remains somewhat tractable, by noting that the the resulting diversion process can still be formulated as a finite-state MMPP, by incorporating the phases of the arrival and service processes into the state of the underlying modulating process. Therefore, when the threshold is sufficiently large, Theorem 7.2 continues to be valid in characterizing the behavior of the central queue. We conclude that the expected delay experienced by jobs at the central queue becomes negligible as $n \to \infty$, and it suffices to simply understand the delay experienced at one local queue. Due to the phase-type nature of the arrival and service processes, the dynamics of a local queue is now more complex than in the Poisson paradigm. Nevertheless, its steady-state distribution is still considerably more tractable compared to that of all n queues under a centralized scheduling policy, and we conjecture that an optimal heavy-traffic queue length scaling of $\sim \log_{\frac{1}{1-p}} \frac{1}{1-\lambda}$ holds for this setting as well.

For another generalization, consider a Partial Pooling system where the arrival rates are *non-uniform*. The fluid model used in [84, 91] heavily relies on the symmetry brought by the uniform arrival rates, and again does not apply to this setting. Similar to the previous example, we may again try applying a decentralized threshold-based diversion policy. The local queues are again decoupled, and fairly easy to characterize. Unfortunately, the current version of the Merging theorem, which requires the individual sub-arrival-process to be i.i.d., does not directly apply to this case, because the rates of the diversion processes from the local queues are no longer uniform, as a result of the non-uniform arrival rates. Still, proving a more general version of the Merging theorem, where the individual MMPPs can be independent but not necessarily identical, could resolve this issue, and it appears to be a more tractable approach than analyzing a centralized policy under non-uniform arrival rates.

7.3.3 From Admission Control to Partial Pooling

Theorem 7.2 can also be used to rigorously establish the connection from the admission control problem (Figure 7-4), introduced in Chapters 5, to the Partial Pooling model (Figure 7-2). In particular, we claim that essentially all diversion policies studied in Chapter 5 can be used as decentralized diversion scheduling policies for the modified Partial Pooling model, so that the resulting steady-state length of the central queue, Q^n, satisfies

$$\limsup_{n\to\infty} \mathbb{E}(Q^n) \le c_\lambda, \tag{7.20}$$

where $c_\lambda > 0$ is a constant independent of n. This further implies that, as $n \to \infty$, the queue delay scaling derived in Chapter 5 is also achievable in the Partial Pooling model.

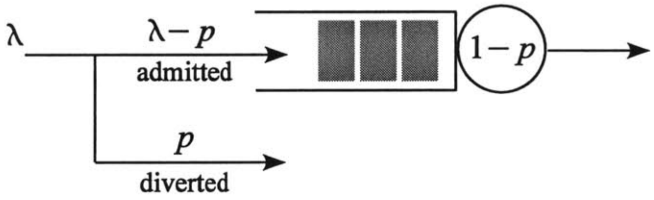

Figure 7-4: An illustration of the admission control problem first introduced in Chapter 5, rep. of Figure 5-1.

For the L-threshold online policies used in Theorem 5.8, the validity of Eq. (7.20) has already been established in the proof of Theorem 7.1 in Section 7.3.1.

We now look at the the π_{NOB} offline diversion policy give in Theorem 5.10. First, the rate of diversion under the π_{NOB} policy is equal to $p - (1 - \lambda) < p$ (Lemma 5.15), which satisfies the requirement of $\rho < 1$ in Theorem 7.2, since the rate of the flexible server pool is pn. We would next like to verify that the diversion process under the π_{NOB} policy is a finite-state MMPP. From Lemma 5.17 and Proposition

5.18, we know that the resulting (local) queue length process after applying π_{NOB} is a positive-recurrent random walk, and a diversion is made if any only if when an arrival occurs while the random walk is in state 0. Hence the diversion process is a MMPP, with the random walk being the underlying modulating process, $W(\cdot)$, and modulating rates

$$r_w = \lambda \cdot \mathbb{I}(w = 0), \quad w \in \mathbb{N}. \tag{7.21}$$

However, because the random walk can be unbounded, the state space of W is no longer finite, and the Merging theorem *cannot* be applied to π_{NOB} directly.

Fortunately, the unbounded-nature of the random walk is fairly easy to fix. We can consider a version of the offline diversion policy for the admissions control problem, which is a "hybrid" between the threshold policy and π_{NOB}.

Definition 7.4 **(π_{NOB} with Upper Threshold).** *Fix $L \in \mathbb{N}$. The L-π_{NOB} policy is defined by the following diversion rule. A job arriving to the queue is diverted if and only if at least one of the following holds:*

1. *The job would would be diverted under the π_{NOB} policy;*
2. *The current queue length, $Q(t-)$, is at least L.*

It is not difficult to show that the L-π_{NOB} effectively puts, at L, an upper boundary to the recurrent random random induced by the π_{NOB} policy. As a result, it can be shown that the modulating Markov chain for each diversion process is now a random walk with two boundaries, at 0 and L, with the same transition rates for all other states in $\{1,\ldots,L-1\}$, as given in Proposition 5.18. The modulating rates are now given by

$$r_w = \lambda \cdot \mathbb{I}(w = 0 \text{ or } L), \quad w \in \mathbb{N}. \tag{7.22}$$

The addition of the upper boundary resolves the issue of an unbounded state space, but it also increases the rate of diversion from $p-(1-\lambda)$. To ensure that the diversion rate is still below p, we simply set the L to be sufficiently large, so that the steady-state probability of the random walk being in state L, which decreases exponentially as a function of L, is well below $(1-\lambda)/\lambda$. By now, we have satisfied all conditions of the Merging theorem, which in turn allows us to establish Eq. (7.20).

We expect it to be possible to extend the above analysis to policies with a finite-lookahead window, by treating the content of the lookahead window as a part of the state-space of the modulating process. The detailed arguments will likely be more technical and complicated, since the number of events observed in a finite lookahead window is itself a random variable. We will leave such extensions to future research.

7.4 Related Research

We highlight some connections between Theorem 7.2 and the existing queueing literature. Queues with arrival processes that are a superposition of multiple sub-processes arise frequently in practice, and have been extensively studied in the literature. To get a sense of why such system can be difficult to analyze, notice that although the family of $G/G/1$ queues, whose arrival process is a renewal process, can be analyzed via the celebrated Kingman's inequality [51], the renewal assumption quickly breaks down when it comes to superposition processes. In facts, the superposition of even two renewal processes will no longer be a renewal process, unless both sub-processes are Poisson [71].

A number of techniques have been developed to understand the performance of queues with superposition processes. These include methods that aim to approximate superposition arrival processes with renewal processes [4, 78], heavy-traffic analysis for queues with superposition arrival processes [89], exponential tail bounds based on certain martingale inequalities [28], and matrix-analytic methods for the special case where the arrival sub-processes are phase type [30]. However, it appears difficult to extend these results to the regime of Theorem 7.2. One main reason is that many of the existing bounds provide performance estimates for cases where the number of arrival sub-processes, n, is *fixed*, and do not explicitly deal with the asymptotic behavior of the steady-state queue length distribution as the n intends to infinity.

More closely related to our work is a body of literature on large deviation principles (LDP) for queueing systems, in which the same scaling regime as Theorem 7.2 is considered. Here, one is concerned with a queue whose arrival process is the superposition of n independent sub-processes, and whose server speed is n. Assuming that the overall traffic intensity stays bounded away from 1, and all arrival sub-processes satisfy some form of LDP, one would like to conclude that, as $n \to \infty$, the resulting steady-state queue length should admit a certain LDP as well. This type of scaling is known in that literature as the many-flow scaling [17, 74, 90], which is also related to the notion of effective bandwidth [49]; the reader is referred to [34], and the references therein, for an overview of the topic. A crucial difference between this literature and Theorem 7.2 is that, typically, the LDP is established for the *scaled* queue length process, Q^n/n, where Q^n is the steady-state queue length in the nth system. In particular, the LDP is of the form

$$\frac{1}{n}\log \mathbb{P}(Q^n/n \geq x) \approx -J(x), \tag{7.23}$$

when n is large, where $J(\cdot)$ is a certain positive rate function (cf. Chapter 7 of [34]). The value of $\mathbb{P}(Q^n/n > x)$ is a relevant quantity in the LDP literature, because it translates into the probability of *buffer overflow* when the buffer size scales linearly in n. However, dividing Q^n by n makes the bound too weak for our purpose, in that it does not allow us to obtain $\limsup_{n\to\infty} \mathbb{E}(Q^n) < \infty$. Closer to the regime of Theorem 7.2, the work of [20] addresses the overflow probability of steady-state workload under a *finite* buffer size that does not scale with n. The authors show that, assuming a certain LDP on the sub-processes, for a fixed $x \geq 0$, the probability $\mathbb{P}(U^n \geq x)$ converges to $\mathbb{P}(\tilde{U}^n \geq x)$ as $n \to \infty$, where U^n and $\tilde{U}^n$ are the steady-state workload of the nth system (similar to Q^n in our setup), and the workload under a Poisson arrival process with the same average rate, respectively. Still, because the convergence result of [20] applies for a *fixed* x, it does not imply the boundedness of $\mathbb{E}(Q^n)$ as n tends to infinity, which would need such convergence to hold uniformly over all $x \in \mathbb{N}$. Compared to Eq. (7.23) and the result of [20], the bound in Theorem 7.2 applies to the unscaled queue length, Q^n, and it does so uniformly over all $x \in \mathbb{N}$, i.e.,

$$\log \mathbb{P}(Q^n \geq x) \leq -\theta x, \quad \text{for all } x \text{ and } n. \tag{7.24}$$

While we believe that it is possible to adapt some of the above-mentioned methods to establish and potentially extend Theorem 7.2, this is, however, beyond the scope of the current report.

7.5 Outline of Proof for Theorem 7.2

The remainder of the chapter is devoted to the proof of the Merging theorem, Theorem 7.2, and in this section we outline the main steps involved. For any fixed n, the merged arrival process, $A^n(\cdot)$, is a Markov-modulated Poisson process, whose instantaneous rate at time t is equal to the sum of the modulating rates from across all the W_is. The main intuition is that, because all W_is are ergodic and independent from each other, one may expect that, in the limit of $n \to \infty$, the *empirical distribution* of the $W_i(t)$s will become concentrated around its steady-state distribution after some finite time. When such concentration occurs, the rate of the superposition arrival process, $A^n(\cdot)$, shall not deviate too much from $n\rho$, where ρ is the modulating rate of W_1 in steady state, and the resulting queue length process at the central queue should be comparable to that of an $M/M/1$ queue with arrival rate ρn and service rate n.

To apply the above intuition in characterizing the steady-state behavior of the system, we will argue as follows.

1. We will focus on the evolution of queue length on a set of evenly spaced discrete time markers, $\{t_k\}$. The length of the time slots between adjacent t_ks is not arbitrary, and will be chosen appropriately to suit the needs of the subsequent analysis.

2. Using the concentration of $W(\cdot)$ around its steady-state distribution, we argue that the rate of $A^n(\cdot)$ is no greater than $(\rho + \epsilon)n$, for some small $\epsilon > 0$, for a significant portion of the time slot. As a result, we show that the displacement of $Q(\cdot)$ over a time slot, $\Delta[k] = Q(t_k) - Q(t_{k-1})$, is "well behaved", in that $\Delta[k]$ satisfies a condition of negative drift on its expected value, as well as an exponential upper bound on the its tail probabilities, both of which are *independent* of the system size, n.

3. Finally, we employ a well-known result of Hajek [39], and conclude that the conditions on $\Delta[k]$ imply that the steady-state distribution of $\{Q(t_k) : k \in \mathbb{N}\}$ satisfies the exponential tail bound in the form of Eq. (7.15), which then carries over to the steady-state of $Q(\cdot)$.

There are two main issues to be addressed in this line of argument. First, the empirical distribution of $W(t)$ may not start in a state that is close to π, and over a long time horizon, it may repeatedly escape from the neighborhood of π, albeit with small probability. To address this issue, we shall incorporate the "mixing" properties of the $W(\cdot)$ into our analysis, by arguing that the empirical distribution of $W(\cdot)$ quickly converges to π during each time slot, regardless of the state of W at the beginning of the time slot. Second, during the periods of time when the empirical distribution of $W(\cdot)$ happens to be far from π, the rate of $A^n(\cdot)$ could *exceed* the total service capacity by an amount that is of order $\Theta(n)$, which would lead to a fast buildup of $Q(\cdot)$. It is hence necessary for us to establish a strong *quantitative* bound on the level of concentration of $W(\cdot)$ around its steady-state distribution that is *exponential* in n, which is then used to show that the impact of such queue length buildup can be effectively controlled, due to its rare chance of occurrence.

The main part of our proof is further divided into two sections, which deal with the dynamics of $W(\cdot)$ and $Q(\cdot)$, respectively.

1. In Section 7.7, we show that, with high probability, the empirical distribution of $W(t)$ becomes concentrated around the steady-state distribution of π after some finite time (Lemma 7.9), and that it remains close to π thereafter (Proposition 7.10), both with exponentially high probability as a function of n.

2. The concentration results on $W(\cdot)$ are then used in Section 7.8 to analyze the evolution of the queue length process, $Q(\cdot)$. Here, we show that the queue length displacement $\Delta[k]$ always admits an exponential tail bound that is independent of n (Proposition 7.14), as well as that the expected displacement $\mathbb{E}(\Delta[k])$ is less than a negative constant, whenever $Q(t_{k-1})$ is greater than a

fixed threshold (Proposition 7.16).

7.5.1 Notation

For a vector $\mathbf{x} = (x_1, \ldots, x_n)$, we will denote by $\|\mathbf{x}\|_\infty$ the l_∞ norm of $\mathbf{x}$: $\|\mathbf{x}\|_\infty = \max_{1 \le i \le n} |x_i|$. Suppose that the coordinates of $\mathbf{x}$ take values in $\{1, \ldots, M\}$. We denote by $\mathbf{h}(x) = (h_1(\mathbf{x}), \ldots, h_M(\mathbf{x}))$ the *empirical distribution* induced by $\mathbf{x}$, represented by the associated empirical probability mass function (PMF),

$$h_w(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{I}\left(x_j = w\right), \quad \forall i \in \{1, \ldots, M\}. \tag{7.25}$$

For a random variable X taking values in a finite set, we will denote by $\mathbf{d}(X)$ its PMF: $d_w(X) = \mathbb{P}(X = w)$.

When necessary, we may use the notation $W_i(w, t)$ to denote the value of $W_i(t)$ given the initial condition $W_i(0) = w$. We will denote by $f(x-)$ the left limit: $f(x-) \triangleq \lim_{y \uparrow x} f(y)$.

7.6 Probability Preliminaries

Let X and Y be real-valued random variables. We write $X \succeq Y$ to mean that X stochastically dominates Y, i.e.,

$$\mathbb{P}(Y \le c) \ge \mathbb{P}(X \le c), \quad \forall c \in \mathbb{R}. \tag{7.26}$$

Similarly, let Y be measurable with respect to the σ-algebra $\mathcal{B}$, we write $(Y\,|\,\mathcal{B}) \preccurlyeq X$ to mean that

$$\mathbb{P}\left(Y \le c \middle| \mathcal{B}\right) \ge \mathbb{P}\left(X \le c\right), \quad \forall c \in \mathbb{R}, \text{ with probability 1.} \tag{7.27}$$

We next define the notion of stochastic dominance between two *counting processes.* Consider two counting processes A and A', where $A(t)$ is the total number of events that have occurred in $[0,t]$. We say that A stochastically dominates A' over $[0,t]$, denoted by $A \succcurlyeq A'$, if there exist counting processes B and B', defined on the same probability space, so that

1. B and B' have the same finite-dimensional distributions as A and A', respectively.
2. $\mathbb{P}\left(B(s) - B(v) \ge B'(s) - B'(v)\right) = 1$, for all $0 \le s < v \le t$.

The following basic fact will be useful.

Lemma 7.5. *Fix $t \in \mathbb{R}_+$. Let A and A' be the counting processes associated with two non-homogeneous Poisson processes, whose instantaneous rates at time s are $U(s)$ and $U'(s)$, respectively. If $\mathbb{P}\left(U(s) \ge U'(s), \forall s \in [0,t]\right) = 1$, then $A \succcurlyeq A'$ over $[0,t]$.*

The next two lemmas concern the stochastic dominance between queue lengths, which are derived from the dominance in arrival processes and initial conditions, respectively.

Let $Q_{A,S}(q_0,t)$ be the number of jobs at time t, for a queue with cumulative process arrival and service token processes $\{A(t) : t \in \mathbb{R}_+\}$ and $\{S(t) : t \in \mathbb{R}_+\}$, respectively, given an initial queue length of q_0. The following lemma states the simple fact that dominance in the arrival processes implies the dominance of queue length, given the same service token process and initial queue length.

Lemma 7.6. *Fix $t \in \mathbb{R}_+$ and $q_0 \in \mathbb{Z}_+$. If $A_{0,t} \succcurlyeq A'_{0,t}$, then*

$$Q_{A,S}(q_0,s) \succcurlyeq Q_{A',S}(q_0,s), \quad \forall s \in [0,t]. \tag{7.28}$$

The following elementary lemma states a form of stochastic dominance between queue lengths that is induced by differences in the initial queue length.

Lemma 7.7 (Impact of Initial Queue Length). *Let A and S be arbitrary arrival and service token processes. For any $a, b \in \mathbb{Z}_+$ and $t > 0$, we have that, almost surely,*

$$Q_{A,S}(a,t) \le Q_{A,S}(a+b,t) \le Q_{A,S}(a,t) + b. \tag{7.29}$$

Finally, the following useful lemma states that the tail probabilities of a Poisson random variable admit an *uniform* exponential upper bound in the regime above its mean. This can also been seen as a quantitative statement of how a Poisson random variable stays close to its mean.

Lemma 7.8 (Uniform Tail Bound of Poisson Distribution). *Let $X_{\alpha n}$ be an Poisson random variable with mean αn, where $\alpha > 0$. For any $\beta > \alpha$, there exists $\theta > 0$, so that*

$$\mathbb{P}(X_{\alpha n} > x) \le \exp(-\theta x), \quad \forall n \in \mathbb{N}, x \ge \beta n. \tag{7.30}$$

In addition, we have that

$$\lim_{n\to\infty} \mathbb{E}\left((X_{\alpha n} - \beta n)^+\right) = 0, \tag{7.31}$$

where $(x)^+ \triangleq \max\{x, 0\}$.

Proof. See Appendix D.1.2. □

7.7 Evolution of the Modulating States

We study in this section the dynamics of the modulating states, W, and present two main technical results. The first result states that, starting from any initial condition, the empirical distribution $\mathbf{h}(W(t))$ shall become close to the steady-state distribution of $W_1(\cdot)$, π, after some finite time, with exponentially high probability as $n \to \infty$. Note that the exponentially dependence of the probability on n stems from the fact that the individual chains are independent. The proof of the proposition is given in Appendix D.1.1.

Proposition 7.9. *Fix $\epsilon > 0$. There exist $\tilde{s}$, γ_1, and $\gamma_2 > 0$, so that, for all $s \geq \tilde{s}$,*

$$\sup_{\mathbf{w}_0 \in \{1,\ldots,M\}^n} \mathbb{P}\left(\left\|\mathbf{h}\left(W(\mathbf{w}_0, s)\right) - \pi\right\|_\infty \geq \epsilon\right) \leq \gamma_2 \exp(-\gamma_1 n), \quad \forall n \in \mathbb{N}. \tag{7.32}$$

The second result states that, if the process W is initialized in a state whose empirical distribution is "close" to π, then it will remain "close" through out a compact interval, with exponentially high probability as $n \to \infty$.

Proposition 7.10 (Exponential Concentration over Finite intervals). *Define $\mathcal{S}_\epsilon$ as*

$$\mathcal{S}_\epsilon = \{\mathbf{w} \in \{1,\ldots,M\}^n : \|\mathbf{h}(\mathbf{w}) - \pi\|_\infty \leq \epsilon\} \tag{7.33}$$

For any δ and $u > 0$, there exist ϵ, β_1, and $\beta_2 > 0$, such that

$$\sup_{\mathbf{w}_0 \in \mathcal{S}_\epsilon} \mathbb{P}\left(\sup_{t \in [0,u]} \left\|\mathbf{h}\left(W(\mathbf{w}_0, t)\right) - \pi\right\|_\infty > \delta\right) \leq \beta_1 \exp(-\beta_2 n), \quad \forall n \in \mathbb{N}. \tag{7.34}$$

Proof. The proof involves two main steps. We first argue that it suffices to prove our claim over a set of appropriately spaced discrete time markers, as opposed to the continuous time interval. We then use a concentration result for a fixed time, in combination of a union bound, to show the desirable concentration of W at all such time markers.

Define a set of (deterministic) discrete time markers,

$$T_k = \frac{\delta}{10\xi}k, \quad k \in \mathbb{Z}_+, \tag{7.35}$$

where ξ is the continuous-time transition rate of the (assumed uniformized) Markov process $W_1(\cdot)$, and let

$$\hat{k} = \max\{k : T_k \leq u\} = \left\lfloor \frac{u}{T_1} \right\rfloor. \tag{7.36}$$

The motivation for the definition of the T_ks is that we want the total number of state transitions during the interval $[T_{k-1}, T_k)$ to be small compared to δn. Specifically, let

$$H_k = \# \text{ of state transitions in } [T_{k-1}, T_k), \quad \forall k \in \mathbb{Z}_+, \tag{7.37}$$

and define $\mathcal{L}$ to be the event where the number of state transitions in $[T_{k-1}, T_k)$ is less than $\delta n/2$ for all k up to $\hat{k}$, that is

$$\mathcal{L} = \left\{ \max_{1 \leq k \leq \hat{k}} H_k \leq \delta n/2 \right\}. \tag{7.38}$$

Because each state transition changes the empirical distribution $\mathbf{h}(W(t))$ by at most $1/n$ (in the l_∞ sense), we have that, conditional on $\mathcal{L}$, the value of $\mathbf{h}(W(t))$ in $[T_{k-1}, T_k)$ can deviate from $\mathbf{h}(W(T_{k-1}))$ by at most $\delta/2$, for all $k \in \{1, \ldots, \hat{k}\}$. Therefore, by the triangle inequality, we have that, conditional on $\mathcal{L}$,

$$\sup_{0 \leq t \leq u} \|\mathbf{h}(W(\mathbf{w}_0, t) - \pi\|_\infty \leq \sup_{1 \leq k \leq \hat{k}} \|\mathbf{h}(W(\mathbf{w}_0, T_k) - \pi\|_\infty + \delta/2. \tag{7.39}$$

The above arguments imply that it suffices for us to show the following, which we shall carry out in detail in the remainder of the proof:

1. The empirical distribution of W at T_k, $\mathbf{h}(W(\mathbf{w}_0, T_k))$, remains close to π for all k up to $\hat{k}$ with high probability.

2. The event $\mathcal{L}$ occurs with high probability.

We first show that, for any $\hat{k} > 0$, there exist ϵ, a, and $b > 0$, such that

$$\sup_{\mathbf{w}_0 \in \mathcal{S}_\epsilon} \mathbb{P}\left(\sup_{0 \le k \le \hat{k}} \|\mathbf{h}\left(W(\mathbf{w}_0, T_k)\right) - \pi\|_\infty > \delta/2 \right) \le a \exp(-bn), \quad \forall n \in \mathbb{N}, \tag{7.40}$$

To this end, the following lemma is useful, which states that, for a *fixed* time t, $\mathbf{h}(W(\mathbf{w}_0, t))$ stays close to π with high probability, if $\mathbf{h}(\mathbf{w}_0)$ is sufficiently close to π.

Lemma 7.11. *For any $t > 0$, there exists $c > 0$, so that for any $\epsilon \in (0, \tilde{\pi}/2]$, there exist a and $b > 0$, so that*

$$\sup_{\mathbf{w}_0 \in \mathcal{S}_\epsilon} \mathbb{P}\left(\|\mathbf{h}\left(W(\mathbf{w}_0, t) - \pi\right)\|_\infty \ge c\epsilon \right) \le a \exp(-bn), \quad \forall n \in \mathbb{N}. \tag{7.41}$$

Proof. See Appendix D.1.3. □

In light of Lemma 7.11, we have that there exist ϵ, a', and $b' > 0$, so that

$$\sup_{\mathbf{w}_0 \in \mathcal{S}_\epsilon} \mathbb{P}\left(\|\mathbf{h}\left(W(\mathbf{w}_0, t) - \pi\right)\|_\infty > \delta/2 \right) \le a' \exp(-b'n), \quad \forall n \in \mathbb{N},\ k \in \{1, \ldots, \tilde{k}\}. \tag{7.42}$$

which further implies that,

$$\begin{aligned}
&\sup_{\mathbf{w}_0\in\mathcal{S}_\epsilon}\mathbb{P}\left(\sup_{1\le k\le\hat{k}}\|\mathbf{h}\left(W(\mathbf{w}_0,T_k)\right)-\pi\|_\infty>\delta/2\right)\\
&\overset{(a)}{\le}\sup_{\mathbf{w}_0\in\mathcal{S}_\epsilon}\sum_{k=1}^{\hat{k}}\mathbb{P}\left(\|\mathbf{h}\left(W(\mathbf{w}_0,T_k)\right)-\pi\|_\infty>\delta/2\right)\\
&\le\sum_{k=1}^{\hat{k}}\sup_{\mathbf{w}_0\in\mathcal{S}_\epsilon}\mathbb{P}\left(\|\mathbf{h}\left(W(\mathbf{w}_0,T_k)\right)-\pi\|_\infty>\delta/2\right)\\
&\le\tilde{k}a'\exp(-b'n),\quad\forall n\in\mathbb{N},
\end{aligned}\tag{7.43}$$

where step (a) follows from the union bound. This proves Eq. (7.40), by setting $a=\tilde{k}a'$ and $b=b'$.

We next turn to the probability of event $\mathcal{L}$ (Eq. (7.38). The total number of state transitions in $[T_k-1,T_k)$, H_k, is a Poisson random variable with

$$\mathbb{E}(H_k)=n\xi T_1=n\xi\frac{\delta}{10\xi}=\frac{\delta n}{10},\quad\forall k\in\mathbb{N},\tag{7.44}$$

where ξ is the rate of transition for each chain. We have that, there exists $\theta>0$, so that

$$\begin{aligned}
1-\mathbb{P}(\mathcal{L})=&\mathbb{P}\left(\max_{1\le k\le K^*}H_k>\delta n/2\right)\\
&\overset{(a)}{\le}\sum_{k=1}^{K^*}\mathbb{P}\left(H_k>\delta n/2\right)\\
&\overset{(b)}{\le}K^*\exp(-\theta\delta n/2)\\
&=\gamma_1\exp(-\gamma_2 n),\quad\forall n\in\mathbb{N},
\end{aligned}\tag{7.45}$$

where $\gamma_1=K^*\theta_1$ and $\gamma_2=\theta_2\delta/2$. Step (a) follows from a union bound, and (b) from

Lemma 7.8 and the fact that $\frac{\mathbb{E}(H_k)}{\delta n/2} = \frac{1}{5} < 1$ for all $n \in \mathbb{N}$.

We are now ready to prove the main claim of Eq. (7.34). Let $\overline{\mathcal{L}}$ be the complement of $\mathcal{L}$. We have that, for any δ and $u > 0$, there exist ϵ, a, b, γ_1, and $\gamma_2 > 0$, so that for all $\mathbf{w}_0 \in \mathcal{S}_\epsilon$,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{t\in[0,u]} \|\mathbf{h}\left(W(\mathbf{w}_0,t)\right) - \pi\|_\infty > \delta\right)\\
=&\mathbb{P}\left(\sup_{t\in[0,u]} \|\mathbf{h}\left(W(\mathbf{w}_0,t)\right) - \pi\|_\infty > \delta\,;\,\mathcal{L}\right) + \mathbb{P}\left(\sup_{t\in[0,u]} \|\mathbf{h}\left(W(\mathbf{w}_0,t)\right) - \pi\|_\infty > \delta\,;\,\overline{\mathcal{L}}\right)\\
\leq&\mathbb{P}\left(\sup_{t\in[0,u]} \|\mathbf{h}\left(W(\mathbf{w}_0,t)\right) - \pi\|_\infty > \delta\,;\,\mathcal{L}\right) + \mathbb{P}\left(\overline{\mathcal{L}}\right)\\
\overset{(a)}{\leq}&\mathbb{P}\left(\sup_{0\leq k\leq K^*} \|\mathbf{h}\left(W(\mathbf{w}_0,t_k)\right) - \pi\|_\infty > \delta/2\,;\,\mathcal{L}\right) + \mathbb{P}\left(\overline{\mathcal{L}}\right)\\
\leq&\mathbb{P}\left(\sup_{0\leq k\leq K^*} \|\mathbf{h}\left(W(\mathbf{w}_0,t_k)\right) - \pi\|_\infty > \delta/2\right) + \mathbb{P}\left(\overline{\mathcal{L}}\right)\\
\overset{(b)}{\leq}& a\exp(-bn) + \mathbb{P}\left(\overline{\mathcal{L}}\right)\\
\overset{(c)}{\leq}& a\exp(-bn) + \gamma_1\exp(-\gamma_2 n), \quad \forall n \in \mathbb{N},
\end{aligned} \tag{7.46}$$

where step (a) follows from Eq. (7.39), and steps (b) and (c) from Eqs. (7.43) and (7.45), respectively. This concludes the proof of Proposition 7.10, by setting $\beta_1 = \max\{a, \gamma_1\}$ and $\beta_2 = \min\{b, \gamma_2\}$ □

7.8 Evolution of the Queue Length

In this section, we will use the concentration results for W developed in Section 7.7, to analyze the evolution of the queue length process, Q. We will focus on the values of Q only over a set of discrete time markers:

$$t_k = Bk, \quad k \in \mathbb{Z}_+, \tag{7.47}$$

where B is a constant, whose value will be specified in the subsequent analysis. We will refer to the interval $[t_{k-1}, t_k)$ as the kth time slot. The main quantity of interest is the displacement of Q between during one time slot,

$$\Delta[k] = Q(t_k) - Q(t_{k-1}). \tag{7.48}$$

In particular, the main results in this section will establish that (1) the distribution of $\Delta[k]$ always admits an exponential tail bound that is independent of n (Proposition 7.14), and (2) that the expected displacement $\mathbb{E}(\Delta[k])$ is less than a negative constant, whenever $Q(t_{k-1})$ is greater than a fixed threshold (Proposition 7.16).

Most of our analysis in this section will heavily rely on $h(W(t))$ being close to π when n is large, which will, in turn, translate into a tractable behavior of $\Delta[k]$. However, due to stochasticity inherent in any finite system, $h(W(t))$ could always escape from π's neighborhood. Therefore, we shall further divide each time slot into two sub-slots, $[t_{k-1}, u_{k-1})$ and $[u_{k-1}, t_k)$, whose lengths are B_1 and B_2, respectively, where u_{k-1} denotes the end point of the first sub-slot in the kth slot:

$$B_1 = u_{k-1} - t_{k-1}, \tag{7.49}$$

$$B_2 = t_k - u_{k-1}, \tag{7.50}$$

$$B = B_1 + B_2. \tag{7.51}$$

The first sub-slot will serve as a "buffer," so that by the end of it, $h(W(t))$ is guaranteed to be sufficiently close to π with high probability, regardless of its value at the beginning of the interval, t_{k-1}. We may then expect that $h(W(t))$ remains close to π *throughout* the remainder of the second sub-slot, as is formalized in the following definition and subsequent lemma.

Definition 7.12. *Denote by $\mathcal{W}_g$ the "good" event concerning the process $W(\cdot)$ during the interval $[t_{k-1}, t_k)$, where*

$$\mathcal{W}_g = \left\{ \sup_{t \in [t_{k-1}+B_1, t_k]} \left\| \mathbf{h}\left(W(\mathbf{w}_0, t)\right) - \pi \right\|_\infty \leq \delta \right\}, \tag{7.52}$$

and by $\overline{\mathcal{W}_g}$ its complement.

The following lemma states that the event $\mathcal{W}_g$ occurs with high probability. The proof follows from combining Propositions 7.9 and 7.10.

Lemma 7.13. *Fix $\rho \in (0,1)$ and $B_2 > 0$, and let the length of the kth interval be $B = B_1 + B_2$. For any $\delta > 0$, there exist $\tilde{B}_1, \zeta_1$, and $\zeta_2 > 0$, so that for all $B_1 > \tilde{B}_1$,*

$$\sup_{\mathbf{w}_0 \in \{1,\ldots,M\}^n} \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, W(t_{k-1}) = \mathbf{w}_0\right) \leq \zeta_1 \exp(-\zeta_2 n), \quad \forall k, n \in \mathbb{N}. \tag{7.53}$$

Proof. See Appendix D.1.4. □

Below is the first main technical result of this section, which establishes an exponential tail bound on the distribution of $\Delta[k]$, whenever the lengths of both sub-slots are sufficiently long.

Proposition 7.14 (Exponential Tail Bound for $\Delta[k]$). *There exists a choice of $\tilde{B}_1$ and $\tilde{B}_2 > 0$, so that for all $B_1 \geq \tilde{B}_1$ and $B_2 \geq \tilde{B}_2$, we have that*

$$\left(\Delta[k] \,\middle|\, \mathbf{X}(t_{k-1})\right) \preccurlyeq Z, \quad \forall k \in \mathbb{N}, \tag{7.54}$$

where Z is a random variable with $\mathbb{E}\left(e^{\lambda Z}\right) = d < \infty$*, for some constant* $\lambda > 0$.

Proof. Fix $\epsilon > 0$, and denote by r^* the largest modulating rate across all M states of chain $W_1(\cdot)$

$$r^* = \max_{1 \le i \le M} r_i, \tag{7.55}$$

and let

$$\tilde{R} = B_1(\epsilon + r^*), \tag{7.56}$$

Note that, because $\epsilon > 0$, the definition of $\tilde{R}$ is ensures that the value of $\tilde{R}n$ is greater than the expected total number of arrivals in A^n during an interval of length B_1, by at least ϵn, regardless of the state of W.

We will develop upper bounds on the distribution of $\Delta[k]$ by considering two separate cases, depending on the occurrence of the event $\mathcal{W}_g$. We first consider the case where $\mathcal{W}_g$ occurs. Denote by $\Lambda^n(t)$ the instantaneously rate of $A^n(\cdot)$ at time t. Because r^* is the maximum modulating rate across all M states of W_1, we have that

$$\Lambda^n(t) \le r^* n, \quad \forall t \in [t_{k-1}, u_{k-1}). \tag{7.57}$$

By the definition of $\mathcal{W}_g$, it is not difficult to verify that conditional on $\mathcal{W}_g$,

$$\Lambda^n(t) \le \rho' n, \quad \forall t \in [u_{k-1}, t_k), \tag{7.58}$$

where

$$\rho' = \rho + M\delta r^*, \tag{7.59}$$

and δ is the constant in the definition of $\mathcal{W}_g$ (cf. Eq. (7.52)). The two upper bounds in Eqs. (7.57) and (7.58) are illustrated in Figure 7-5-(a).

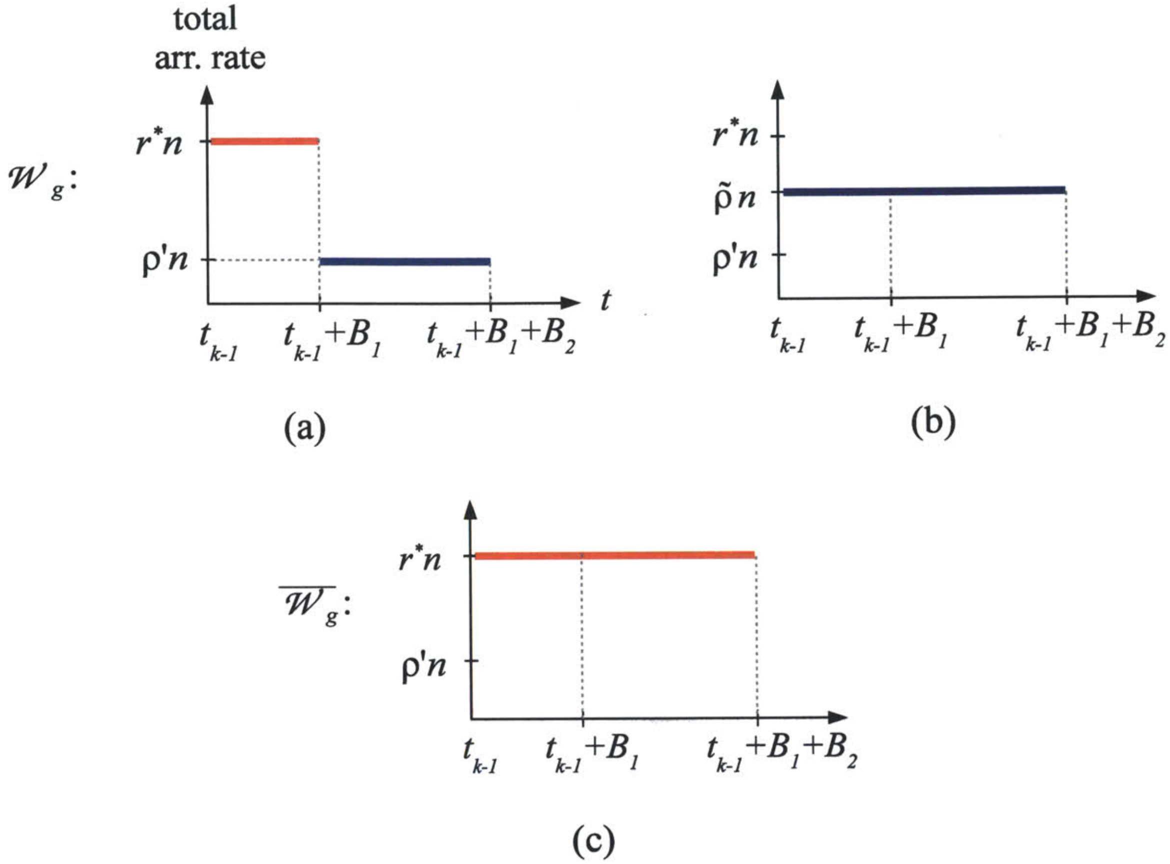

Figure 7-5: Illustrations of total arrival rates during one time slot.

Qualitatively, Eqs. (7.57) and (7.58) suggest that, conditional on $\mathcal{W}_g$, it suffices to think of the arrival process A^n as having a high rate of r^*n during the first sub-slot, and a relatively moderate rate of $\rho'n$ during the second sub-slot. Analyzing the resulting queueing dynamics from these two distinct phases can be, however, a bit cumbersome. Therefore, we shall argue next that it suffices to consider a *homogeneous* arrival process throughout the entire time slot $[t_{k-1}, t_k)$, whose rate is an appropriate average of the upper bounds in Eqs. (7.57) and (7.58). Define $\tilde{\rho}$ as

$$\tilde{\rho} = \frac{r^* B_1 + \rho' B_2}{B_1 + B_2}. \tag{7.60}$$

The following lemma states that, conditional on $\mathcal{W}_g$, a system in which the arrival process during the kth time slot is a homogeneous Poisson process of rate $\rho' n$, as illustrated in Figure 7-5-(b), yields a larger queue length at time t_k.

Lemma 7.15. *Fix $n \in \mathbb{N}$. Denote by $\tilde{A}$ a homogeneous Poisson process of rate $\rho' n$, independent of all other randomness in the system. Then*

$$\left(Q(t_k) \,\middle|\, Q(t_{k-1}) = j, \mathcal{W}_g\right) \preccurlyeq Q_{\tilde{A},S^n}(j,t), \quad \forall j \in \mathbb{Z}_+, \tag{7.61}$$

where $Q_{A,S}(j,t)$ represents the length of a queue at time t with j jobs at $t = 0$, under an arrival process A and service token process S.

Proof. Fix $n \in \mathbb{N}$. Let H be a Poisson process whose instantaneous rate is $r^* n$ and $\tilde{\rho} n$ during the interval $[0, B_1)$ and $[B_1, B_2)$, respectively (cf. Figure 7-5-(a)). By Lemma 7.6 and Eqs. (7.57) and (7.58), we have that

$$\left(Q(t_k) \,\middle|\, Q(t_{k-1}) = j, \mathcal{W}_g\right) \preccurlyeq Q_{H,S^n}(j,B), \tag{7.62}$$

for all $j \in \mathbb{Z}_+$. Hence it suffices to demonstrate that

$$Q_{H,S^n}(j,B) \preccurlyeq Q_{\tilde{A},S^n}(j,B), \quad \forall j \in \mathbb{N}. \tag{7.63}$$

We shall prove Eq. (7.63) by showing a more general result. Fix $T \geq 0$. Let Y and U be two Poisson processes, with instantaneous rates $r_Y(t)$ and $r_U(t)$, respectively,

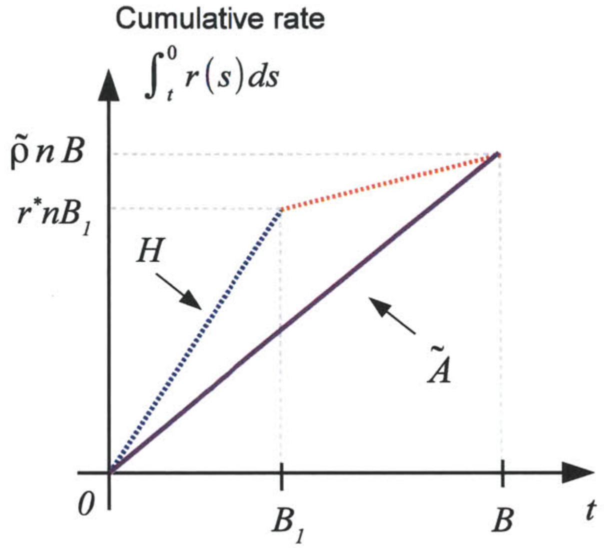

Figure 7-6: The cumulative rates for processes $\tilde{A}$ (solid) and H (dashed), during the interval $[0, B)$. The instantaneous rates for H and $\tilde{A}$ are given in Figures 7-5-(a) and 7-5-(b), respectively.

so that

$$\int_0^t r_Y(s)ds \geq \int_0^t r_U(s)ds, \quad \forall t \in (0,T), \tag{7.64}$$

and

$$\int_0^T r_Y(s)ds = \int_0^T r_U(s)ds. \tag{7.65}$$

In particular, note that Eqs. (7.64) and (7.65) are satisfied by the rates of H and $\tilde{A}$, with $Y = H$ and $U = \tilde{A}$ (cf. Figure 7-6). The claim is that for any point process S that is independent from Y or U, we have that

$$Q_{Y,S}(j,T) \preccurlyeq Q_{U,S}(j,T), \quad \forall j \in \mathbb{Z}_+. \tag{7.66}$$

We will show Eq. (7.66) via the following coupling between Y and U. Let $\{X_i :$

$i \in \mathbb{N}\}$ be a sequence of i.i.d. Exponential random variables with $\mathbb{E}(X_1) = 1$. Define the counting processes

$$Y'(t) = \max\left\{k : \sum_{i=1}^{k} X_i \leq \int_{s=0}^{t} r_Y(s)ds\right\}, \quad t \in [0,T], \tag{7.67}$$

and

$$U'(t) = \max\left\{k : \sum_{i=1}^{k} X_i \leq \int_{s=0}^{t} r_U(s)ds\right\}, \quad t \in [0,T], \tag{7.68}$$

By construction, Y' and U' admit the same finite-dimensional distribution as Y and U, respectively. By Eq. (7.64), we have that

$$\mathbb{P}\left(Y'(t) \geq U'(t), \forall t \in [0,T]\right) = 1, \tag{7.69}$$

where the probability is measured with respect to the randomness in $\{X_i\}$. This implies that, with probability one,

$$\begin{aligned} Y'(T) - Y'(t) &\overset{(a)}{=} U'(T) - Y'(t) \\ &\overset{(b)}{\leq} U'(T) - U'(t), \quad \forall t \in [0,T], \end{aligned} \tag{7.70}$$

where step (a) follows from Eq. (7.65), and step (b) from Eq. (7.69). We are now ready to establish Eq. (7.64). By Lindley's recursion, we have that

$$\begin{aligned} Q_{Y,S}(j,T) &\overset{d}{=} \sup_{t\in[0,T]} \left[j\mathbf{I}(t=0) + (Y'(T) - Y'(t)) - (S(T) - S(t))\right] \\ &\overset{(a)}{\preccurlyeq} \sup_{t\in[0,T]} \left[j\mathbf{I}(t=0) + (U'(T) - U'(t)) - (S(T) - S(t))\right] \\ &\overset{d}{=} Q_{U,S}(j,T), \end{aligned} \tag{7.71}$$

where step (a) follows from Eq. (7.69). This proves Eq. (7.64), which in turns proves our claim, by letting $Y = H$, $U = \tilde{A}$, and $S = S^n$. □

With Lemma 7.15 at hand, we are now ready to show an exponential tail bound on the distribution of $\Delta[k]$, conditional on $\mathcal{W}_g$. Let δ be sufficiently small so that

$$\rho' = \rho + M\delta r^* < 1. \tag{7.72}$$

Let $B_1 = 2\tilde{B}_1$, where $\tilde{B}_1$ is defined as in Lemma 7.13 given the above choice of δ. Set $\tilde{B}$ to be sufficiently large, so that

$$\tilde{\rho} = \frac{r^* B_1 + \rho' B_2}{B_1 + B_2} < 1, \quad \forall B_2 > \tilde{B}_2, \tag{7.73}$$

and let $B_2 = 2\tilde{B}_2$. We have that, for any $n \geq 1$,

$$\begin{aligned}
\left(\Delta[k] \,\middle|\, Q(t_{k-1}) = j, \mathcal{W}_g\right) &= \left(Q(t_k) - Q(t_{k-1}) \,\middle|\, Q(t_{k-1}) = j,, \mathcal{W}_g\right) \\
&= \left(Q(t_k) \,\middle|\, Q(t_{k-1}) = j, \mathcal{W}_g\right) - j \\
&\overset{(a)}{\preccurlyeq} Q_{\tilde{A},S^n}(j, B_2) - j \\
&\overset{(b)}{\preccurlyeq} Q_{\tilde{A},S^n}(0, B_2) + j - j \\
&\overset{(c)}{\preccurlyeq} G(\tilde{\rho}),
\end{aligned} \tag{7.74}$$

where $\tilde{A}$ is a Poisson process of rate $\tilde{\rho}n$, and $G(\tilde{\tilde{\rho}})$ is a geometric random variable with parameter ρ, i.e.,

$$\mathbb{P}(G(\tilde{\rho}) \geq x) = \tilde{\rho}^x, \quad \forall x \in \mathbb{Z}_+. \tag{7.75}$$

Step (a) follows from Lemma 7.15, and (b) from Lemma 7.7. For step (c), note that under arrival process $\tilde{A}$ and service token process S^n, the queue length process evolves as the total number of jobs in system for a $M/M/1$ queue with traffic intensity $\tilde{\rho}$. We then invoke the elementary fact that, in an initially empty $M/M/1$ queue with traffic intensity $\rho < 1$, the total number of jobs in system at any time $t > 0$ is stochastically dominated by its steady state distribution, which is geometrically distributed with parameter ρ.

We now turn to the case where the event $\mathcal{W}_g$ does not occur. Let R^* be defined as

$$R^* = B_1 r^* = B_1 \max_{1 \le i \le m} r_i. \tag{7.76}$$

Note that R^* is strictly less than $\tilde{R}$ (Eq. (7.56)). We shall use the following trivial upper bound

$$(Q(t_k) \,|\, \mathbf{X}(t_{k-1}), \overline{\mathcal{W}_g}) \preccurlyeq Q(t_{k-1}) + X_{R^* n}, \tag{7.77}$$

where $X_{R^* n}$ is a Poisson random variable with mean $R^* n$, independent of $Q(t_{k-1})$. Eq. (7.77) corresponds to the case where all arrival processes are generating arrivals at the maximum rate of R^* (cf. Figure 7-5-(c)), and there is no service during the interval $[t_{k-1}, t_k)$.

Having covered the two cases depending on whether $\mathcal{W}_g$ occurs, in Eqs. (7.74) and Eq. (7.77), respectively, we are now ready to establish the main upper bound on $\Delta[k]$. Fix $k \in \mathbb{N}$. Since we had set $B_1 = 2\tilde{B}_1$, by Lemma 7.13, there exist ζ_1 and $\zeta_2 > 0$, so that

$$\sup_{\mathbf{w}_0 \in \{1,\ldots,M\}^n} \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, W(t_{k-1}) = \mathbf{w}_0\right) \le \zeta_1 \exp(-\zeta_2 n), \quad \forall n \in \mathbb{N}. \tag{7.78}$$

We have that

$$\begin{aligned}\mathbb{P}\left(\Delta[k] \geq x \,\middle|\, \mathbf{X}(t_{k-1})\right) =& \mathbb{P}\left(\Delta[k] \geq x \,\middle|\, \mathbf{X}(t_{k-1}), \overline{\mathcal{W}_g}\right) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) \\ &+ \mathbb{P}\left(\Delta[k] \geq x \,\middle|\, \mathbf{X}(t_{k-1}), \mathcal{W}_g\right) \mathbb{P}(\mathcal{W}_g \,|\, \mathbf{X}(t_{k-1})) \\ \overset{(a)}{\leq}& \mathbb{P}(X_{R^* n} \geq x) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) \\ &+ \mathbb{P}\left(\Delta[k] \geq x \,\middle|\, \mathbf{X}(t_{k-1}), \mathcal{W}_g\right) \mathbb{P}(\mathcal{W}_g \,|\, \mathbf{X}(t_{k-1})) \\ \overset{(b)}{\leq}& \mathbb{P}(X_{R^* n} \geq x) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) + \mathbb{P}\left(G(\tilde{\rho}) \geq x\right) \\ =& \mathbb{P}(X_{R^* n} \geq x) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) + \tilde{\rho}^x, \end{aligned} \tag{7.79}$$

where step (a) follows from Eq. (7.77), and (b) from Eq. (7.74) and the fact that $\mathbb{P}(\mathcal{W}_g \,|\, \mathbf{X}(t_{k-1})) \leq 1$. For the first term in Eq. (7.79), we have that there exist $\zeta_1, \zeta_2, \theta_1$, and $\theta_2 > 0$, such that

$$\begin{aligned}& \mathbb{P}(X_{R^* n} > x) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) \\ \leq& \left(\mathbf{I}(x \leq \tilde{R} n) + \mathbb{P}(X_{R^* n} > x) \mathbf{I}(x > \tilde{R} n)\right) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) \\ \overset{(a)}{\leq}& \mathbf{I}(x \leq \tilde{R} n) \mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) + \theta_1 \exp(-\theta_2 x) \mathbf{I}(x > \tilde{R} n) \\ \overset{(b)}{\leq}& \zeta_1 \exp(-\zeta_2 n) \mathbf{I}(x \leq \tilde{R} n) + \theta_1 \exp(-\theta_2 x) \mathbf{I}(x > \tilde{R} n) \\ \leq& \zeta_1 \exp\left(-\frac{\zeta_2}{\tilde{R}} x\right) \mathbf{I}(x \leq \tilde{R} n) + \theta_1 \exp(-\theta_2 x) \mathbf{I}(x > \tilde{R} n) \\ \leq& (\zeta_1 + \theta_1) \exp\left(-\min\{\zeta_2/\tilde{R}, \theta_2\} x\right), \quad \forall n \in \mathbb{N}, \end{aligned} \tag{7.80}$$

where step (a) follows from Lemma 7.8 with $\alpha = R^*$ and $\beta = \tilde{R}$, and the fact that $\tilde{R} > R^*$, and (b) from Eq. (7.78) and the assumption that $B_1 > \tilde{B}_1$. Substituting

(7.80) into Eq. (7.79), we obtained that

$$\begin{aligned}\mathbb{P}\left(\Delta[k] \geq x \,\middle|\, \mathbf{X}(t_{k-1})\right) \leq & \mathbb{P}(X_{R^*n} \geq x)\mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, \mathbf{X}(t_{k-1})\right) + \tilde{\rho}^x \\ \leq & (\zeta_1 + \theta_1) \exp\left(-\min\{\zeta_2/\tilde{R}, \theta_2\}x\right) + \tilde{\rho}^x \\ \leq & \gamma_1 \exp(-\gamma_2 x), \end{aligned} \tag{7.81}$$

where $\gamma_1 = \zeta_1 + \theta_1 + 1$ and $\gamma_2 = \min\{\zeta_2/\tilde{R}, \theta_2, \ln(1/\tilde{\rho})\}$. We have thus proven Proposition 7.14, by letting Z be such that $\mathbb{P}(Z \geq x) = \gamma_1 \exp(-\gamma_2 x)$ for all $x \geq 0$, and $\lambda = \gamma_2/2$. □

The next proposition is the second main technical result of this section, which states that the expected displacement $\mathbb{E}(\Delta[k])$ is *negative* and bounded away from zero, whenever $Q(t_{k-1})$ exceeds a fixed threshold.

Proposition 7.16 (Conditional Negative Drift of $\Delta[k]$). *Fix $\rho \in (0,1)$, and $B_1 \geq \tilde{B}_1$, where $\tilde{B}_1$ was defined in Proposition 7.14. There exists $\hat{B}_2, q, d$, and $n_0 > 0$, such that*

$$\mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), Q(t_{k-1}) = j) \leq -d, \quad \forall k \in \mathbb{N}, \tag{7.82}$$

for all $j > q$, $n \geq n_0$, and almost all realizations of $W(t_{k-1})$.

Proof. We shall use a line of arguments similar to that in the proof of Proposition 7.14. First, consider the case where the event $\mathcal{W}_g$ occurs. Set $B_1 > \tilde{B}_1$ and $B_2 > \tilde{B}_2$ as in Proposition 7.14. We have that

$$\begin{aligned}\mathbb{E}\left(\Delta[k] \,\middle|\, \mathcal{W}_g, Q(t_{k-1}) = j\right) = & \mathbb{E}\left(Q(t_k) \,\middle|\, \mathcal{W}_g, Q(t_{k-1}) = j\right) - j \\ \leq & \mathbb{E}\left(Q_{\tilde{A}, S^n}(j, B_2)\right) - j, \end{aligned} \tag{7.83}$$

where $\tilde{A}$ is a homogeneous Poisson process with rate $\tilde{\rho}n$, with $\tilde{\rho} = \frac{r^* B_1 + \rho' B_2}{B_1 + B_2} < 1$ (Eq. (7.73)), and the inequality follows from Lemma 7.15.

We next recall an elementary fact: denote by $Q^M(t)$ the total number of jobs in system at time t in an $M/M/1$ queue with arrival rate $\rho < 1$ and service rate 1, and by $\mathbb{E}(Q^M)$ its expected value in steady state. Then there exist d and $\tilde{b} > 0$, so that if the systems starts with j jobs at $t = 0$, with $j > \mathbb{E}(Q^M)$, we have

$$\mathbb{E}(Q^M(t)|Q^M(0) = j) \leq j - d, \quad \forall t \geq \tilde{b}. \tag{7.84}$$

Note that $Q_{\tilde{A},S^n}$ evolves as the number of jobs in system for an $M/M/1$ queue with with arrival rate $\tilde{\rho}n$ and service rate n. Scaling time by a factor of n, we see that the distribution of $Q_{\tilde{A},S^n}(j, B_2)$ is the same as the number of jobs in system for an $M/M/1$ queue with arrival rate $\tilde{\rho}$ and service rate 1 at time nB_2, with j jobs initially. Applying Eq. (7.84) to (7.83), we have that there exist $\hat{B}_2, q$ and $d > 0$, so that for any $B_2 > \hat{B}_2$, we have that

$$\begin{aligned}\mathbb{E}\left(\Delta[k] \,|\, \mathcal{W}_g, Q(t_{k-1}) = j\right) \leq & \mathbb{E}\left(Q_{\tilde{A},S^n}(j, B_2)\right) - j \\ \leq & \left(j - \frac{3}{2}d\right) - j \\ = & -\frac{3}{2}d, \end{aligned} \tag{7.85}$$

for all $n \geq \mathbb{N}$, and $j > q$.

In the case where the event $\overline{\mathcal{W}_g}$ occurs, we will use the trivial upper bound:

$$\mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), \overline{\mathcal{W}_g}, Q(t_{k-1}) = j) \leq r^* Bn + j, \quad \forall j \in \mathbb{Z}_+, n \in \mathbb{N}, \tag{7.86}$$

where $r^* Bn$ corresponds to the expected number of arrivals in $[t_{k-1}, t_k)$, assuming all arrival processes remain in the state with the highest modulating rate, r^* (cf. Eq. (7.55)).

Combining Eqs. (7.85) and (7.86), we have that, whenever $B_1 > \tilde{B}_1$ and $B_2 > \hat{B}_2$,

$$\begin{aligned}
&\mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), Q(t_{k-1}) = j) \\
=&\mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), \mathcal{W}_g, Q(t_{k-1}) = j)\mathbb{P}\left(\mathcal{W}_g \,|\, W(t_{k-1}), Q(t_{k-1}) = j\right) \\
&+ \mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), \overline{\mathcal{W}_g}, Q(t_{k-1}) = j)\mathbb{P}\left(\overline{\mathcal{W}_g} \,|\, W(t_{k-1}), Q(t_{k-1}) = j\right) \\
\overset{(a)}{\leq}& -\frac{3}{2}q + \mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), \overline{\mathcal{W}_g}, Q(t_{k-1}) = j)\mathbb{P}\left(\overline{\mathcal{W}_g} \,|\, W(t_{k-1}), Q(t_{k-1}) = j\right) \\
\overset{(b)}{\leq}& -\frac{3}{2}q + r^* B n \mathbb{P}\left(\overline{\mathcal{W}_g} \,|\, W(t_{k-1}), Q(t_{k-1}) = j\right) \\
\overset{(c)}{\leq}& -\frac{3}{2}q + r^* B \zeta_1 n \exp(-\zeta_2 n),
\end{aligned} \tag{7.87}$$

for all $j > q$ and $n \in \mathbb{N}$. Steps (a) and (b) follow from Eqs. (7.85) and (7.86), respectively, and (c) from Lemma 7.13. Since for all $\zeta > 0$, we have that $n \exp(-\zeta n) \to 0$ as $n \to \infty$, we conclude from Eq. (7.87) that there exists $n_0 > 0$, such that

$$\mathbb{E}(\Delta[k] \,|\, W(t_{k-1}), Q(t_{k-1}) = j) \leq -\frac{3}{2}d + R^* B \zeta_1 n \exp(-\zeta_2 n) \leq -d, \tag{7.88}$$

for all $j > q$ and $n \geq n_0$, which proves Proposition 7.16. □

7.9 Proof of the Merging Theorem

We now complete the proof of Theorem 7.2. Let $\{Y_k : k \in \mathbb{Z}_+\}$ be a sequence of random variables defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, adapted to an increasing sequence of σ-algebra, $\{\mathcal{F}_k : k \in \mathbb{Z}_+\}$, where Y_k is $\mathcal{F}_k$-measurable. We we will use the following well-known result, adapted from [39], which establishes an exponential tail bound on the steady-state distribution of $\{Y_k\}$.

Proposition 7.17 ([39]). *Suppose $Y_0 = 0$, and that the following conditions hold.*

Condition (a). *There exists a random variable Z and a constant $\lambda > 0$, so that $\mathbb{E}(e^{\lambda Z}) < \infty$, and*

$$\left(Y_k - Y_{k-1} \middle| \mathcal{F}_k\right) \preccurlyeq Z, \quad \forall k \in \mathbb{Z}_+. \tag{7.89}$$

Condition (b). *There exist constants C and $\epsilon_d > 0$, so that*

$$\mathbb{E}\left(Y_k - Y_{k-1} + \epsilon_d; Q[k] > C \middle| \mathcal{F}_k\right) \leq 0, \quad \forall k \in \mathbb{Z}_+. \tag{7.90}$$

Then, there exist θ_1, θ_2, and $k_0 > 0$, so that, for all $k \geq k_0$,

$$\mathbb{P}(Y_k \geq x) \leq \theta_1 \exp(-\theta_2 x), \quad \forall x \in \mathbb{R}_+. \tag{7.91}$$

Fix $B_1 \geq \tilde{B}_1$ and $B_2 \geq \max\{\hat{B}_2, \tilde{B}_2\}$, where $\tilde{B}_1$ and $\tilde{B}_2$ were defined in Proposition 7.14 and $\hat{B}_2$ in Proposition 7.16. We now apply Proposition 7.17 to our setting, by letting $Y_k = Q[k]$. In our case, $Q[k]$ is measurable with respect to the σ-algebra generated by $\mathbf{X}(t_k)$, and Conditions (a) and (b) in Proposition 7.17 are satisfied because of Propositions 7.14 and 7.16, respectively. We then obtain that there exist k_0, θ_1, and $\theta_2 > 0$, so that for all $k \geq k_0$,

$$\mathbb{P}(Q(Bk) \geq x) \leq \theta_1 \exp(-\theta_2 x), \quad \forall x \in \mathbb{R}_+. \tag{7.92}$$

It is easy to verify that, for any $n \in \mathbb{N}$, the Markov process $\mathbf{X}$ is irreducible and aperiodic. As a result, the positive recurrence of the discrete chain $\{Q(Bk) : k \in \mathbb{N}\}$, which is implied by Eq. (7.92), also implies the positive recurrence, and hence ergodicity, of the continuous-time process $\{Q(t) : t \in \mathbb{R}_+\}$. The ergodicity of $Q(\cdot)$ combined with Eq. (7.92) proves Theorem 7.2.

7.10 Summary and Future Work

In this chapter, we studied the role of real-time state information in the Partial Pooling flexible architecture analyzed in [84, 91]. We demonstrate that, in the regime of $n \to \infty$, it is possible to use a class of *decentralized* scheduling policies and achieve the same *optimal heavy-traffic delay scaling* as the centralized longest-queue-first (LQF) policy analyzed by [84, 91].

The decentralized policy is based on diverting incoming jobs from local queues to a central queue by comparing the local queue length to a fixed threshold. Jobs in the central queue are served exclusively by the flexible server pool. Our main technical contribution goes into characterizing the resulting steady-state queue length at the central queue. To do so, we prove a Merging theorem (Theorem 7.2), which yields an exponential tail bound on the steady-state queue length distribution of a queue, whose arrival process is the *superposition* of n independent sub-streams, in the limit where the number of sub-streams and the service capacity of the queue tend to infinity simultaneously.

The Merging theorem also opens up new possibilities for generalizing the original Partial Pooling model in different ways, such as allowing for phase-type arrival and service processes at the local queues (Section 7.3.2), and potentially, for analyzing systems with non-uniform arrival rates. Similar arguments using the Merging theorem also allowed us to rigorously interpret the admission control problem (both with and without future information) as the decision problems faced by the local queues in the Partial Pooling model (Section 7.3.3), and hence carrying the performance guarantees developed for the admission control model in Chapter 5 to the Partial Pooling model.

At a higher level, the use of the Merging theorem and the architecture with a central queue provides a simpler, and somewhat deeper, conceptual picture of flexibility's role in the effectiveness of Partial Pooling models. In effect, the Merging theorem allows us to *decouple* the local resource allocation decisions from that of the flexible server pool. Viewed from this angle, we see that in a large-scale system, the presence of a small fraction of fully flexible processing resources translates into the ability to simply *divert* a fraction of the incoming arrivals from each of the local queues. The threshold policy, or one that takes into account future information, ensures that such diversions are exercised only when there is a sufficient indication of congestion.

The current version of the Merging theorem applies only to cases where the sub-arrival processes are finite-state Markov-modulated Poisson processes, all with the same average rate. An interesting and relevant question deserving further study is whether the Merging theorem can be extended to incorporate more general stationary point processes as sub-arrival processes, and with possibly different rates. Such extensions would allow us to analyze Partial Pooling systems with more general arrival processes, and scenarios where the arrival rates are non-uniform (Section 7.3.2), both of which may be more relevant for practical applications. We believe that an approach similar to ours, by showing a strong concentration bound for the "states" of the sub-arrival processes, is promising. However, the arguments can be more difficult, because one would have to track a more complex state evolution than that induced by a collection of finite-state Markov chains.

Chapter 8

Concluding Remarks

The present report is centered around the role of partial flexibility in large-scale dynamic resource allocation problems. Our results demonstrate that, with an appropriate architecture, scheduling policy, and adequate amount of (future) information, a system with even a little flexibility can often significantly outperform its inflexible counterpart in terms of delay and capacity, and sometimes be almost as good as a fully flexible system.

Some of the open problems that concern specific models have been stated at the end of the corresponding chapters. Instead of restating them here, we will focus on some higher-level issues that could be interesting directions for future research.

Fundamental Limitations of Partial Flexibility. While most of our present investigation points towards the power of partially flexible systems compared to inflexible systems, it is equally important to understand whether there are fundamental limitations of partial flexibility, too. In particular, are there situations in which a system with limited flexibility performs *significantly more poorly* than a system in which flexibility is abundant? More generally, does there exists a non-trivial level of flexibility that is necessary and sufficient for achieving desirable performance?

One such consideration has already surfaced as a main open problem in our study of the Sparse Flexibility architectures in Chapters 3 and 4. We have seen that in the fully flexible system (e.g., an $M/M/n$ queue), an optimal average delay scaling can be achieved *for any* arrival rate vector that satisfies the rate condition, under any work-conserving scheduling policy (cf. discussions in Section 3.1). In contrast, obtaining parallel results for partially flexible systems appears to be substantially more difficult, and we have not been to do so for any of the three proposed partially flexible architectures. Our delay guarantees for the Random Graph and Random Modular architectures hold only in a probabilistic sense (cf. Theorems 3.6 and 3.9), i.e., for most arrival rate vectors in $\Lambda_n(u_n)$, while the Expanded Modular architecture provides a worst-case delay guarantee at the expense of a reduced capacity region (cf. Theorem 3.11). Roughly speaking, these difficulties suggest a potential fundamental limitation of partially flexible systems: that a large capacity region in a partially flexible system *necessarily* comes at the expense of a weaker worst-case delay guarantee (although we conjectured otherwise, cf. Conjecture 3.15).

Flexibility in time. The types of flexibility that we have studied are built into the system *spatially*, in the form of servers that can process different type of demands, and are fixed over time. However, there is also a *temporal* aspect of flexibility, which we have not covered with the existing models. For instance, one could consider flexible systems where

1. The flexibility of a resource or demand may vary over time. Such scenarios could arise, for instance, in the modeling of human systems, where the skill sets of the agents gradually broaden over time as a result of learning.

2. The times of arrivals or service availability can be adjusted, within certain constraint, around their original realizations. For instance, the decision maker may be able to encourage a job to arrive earlier than its original time of arrival (advancement), or "store" a service token for a short amount of time after it has been generated, to be used for serving a job that comes at a later time (inventory).

Analogous to the types of questions raised in this report, one may ask whether a small amount of flexibility *across the time horizon* has a significant impact on system performance.

Flexibility in multi-stage systems. This report is solely concerned with *parallel* resource allocation systems, where a job immediately *departs* from the system after it has been processed by a server. This family of models, however, does not capture many *multi-stage* service systems, where a job needs to receive service from multiple servers *sequentially* before departing (e.g., queues in tandem).

In the context of multi-stage systems, we have the additional possibility of having flexibility across different stages of the system. For instance, for a system with a sequence of queues in tandem, each server may be able to process jobs from different stages. Does having flexibility help in such systems, and how does performance vary with respect to the number of "stages" that a server is able to process? The dynamics induced by the sequential nature of a multi-stage system appears to be significantly different from those considered in the present report, and we suspect that to answer these questions one would likely need a very different set of analytical techniques and problem formulations.

Bibliography

[1] O. Akgun, R. Righter, and R. Wolff. Multiple server system with flexible arrivals. *Advances in Applied Probability*, 43(4):985–1004, 2011.

[2] O. Akgun, R. Righter, and R. Wolff. Understanding the marginal impact of customer flexibility. *Queueing Systems*, 70:1–19, 2012.

[3] M. Alanyali and M. Dashouk. *On power-of-choice in downlink transmission scheduling.* Inform. Theory and Applicat. Workshop, U.C. San Diego, 2008.

[4] S. Albin. On poisson approximations for superposition arrival processes in queues. *Management Science*, 28(2):126–137, 1982.

[5] E. Altman and A. Swartz. Markov decision problems and state-action frequencies. *SIAM J. Control and Optimization*, 29(4):786–809, July 1991.

[6] A. S. Asratian, T. M. J. Denley, and R. Haggkvist. *Bipartite graphs and their applications.* Cambridge University Press, 1998.

[7] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive on-line routing. In *Proceedings of Foundations of Computer Science*, 1993.

[8] Y. Azar. On-line load balancing. *Online Algorithms, Springer Berlin Heidelberg*, pages 178–195, 1998.

[9] A. Bassamboo, R. S. Randhawa, and J. A. V. Mieghem. Optimal flexibility configurations in news vendor networks: Going beyond chaining and pairing. *Management Science*, 56(8):1285–1303, 2010.

[10] A. Bassamboo, R. S. Randhawa, and J. A. V. Mieghem. A little flexibility is all you need: on the asymptotic value of flexible capacity in parallel queuing systems. *Operations Research*, 60(6):1423–1435, December 2012.

[11] S. L. Bell and R. J. Williams. Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: asymptotic optimality of a threshold policy. *Ann. Appl. Probab*, 11(3):608–649, 2001.

[12] F. J. Beutler and K. W. Ross. Time-average optimal constrained semi-Markov decision processes. *Adv. Appl. Prob*, 18:341–359, 1986.

[13] E. Bish, A. Muriel, and S. Biller. Managing flexible capacity in a make-to-order environment. *Management Science*, 51(2):167–180, 2005.

[14] S. R. Bodas. *High-performance scheduling algorithms for wireless networks.* PhD thesis, University of Texas at Austin, December 2010.

[15] V. S. Borkar and S. K. Mitter. Lqg control with communication constraints. *Communications, Computation, Control and Signal Processing*, pages 365–373, 1997.

[16] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 2005.

[17] D. Botvich and N. G. Duffield. Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. *Queueing Systems*, 20(3-4):293–320, 1995.

[18] M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):275–286, 2010.

[19] D. B. Brown, J. E. Smith, and P. Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4):785–801, 2010.

[20] J. Cao and K. Ramanan. A poisson limit for buffer overflow probabilities. In *IEEE INFOCOM*, page 1, 2002.

[21] M. Carr and B. Hajek. Scheduling with asynchronous service opportunities with

applications to multiple satellite systems. *IEEE Trans. Automatic Control*, 38:1820–1833, 1993.

[22] M. Chou, G. A. Chua, C.-P. Teo, and H. Zheng. Design for process flexibility: efficiency of the long chain and sparse structure. *Operations Research*, 58(1):43–58, 2010.

[23] M. Chou, C.-P. Teo, and H. Zheng. Process flexibility revisited: the graph expander and its applications. *Operations Research*, 59(5):1090–1105, 2011.

[24] E. G. Coffman, P. J. Jr., and B. Poonen. Reservation probabilities. *Adv. Perf. Anal*, 2:129–158, 1999.

[25] T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

[26] D. F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Trans. Automatic Control*, 35(8):916–924, 1990.

[27] V. V. Desai, V. F. Farias, and C. C. Moallemi. Pathwise optimization for optimal stopping problems. *Management Science*, 58(12):2292–2308, 2012.

[28] N. G. Duffield. Exponential bounds for queues with Markovian arrivals. *Queueing Systems*, 17(3-4):413–430, 1994.

[29] P. Erdös and A. Rènyi. On random matrices. *Magyar Tud. Akad. Mat. Kutato Int. Kozl*, 8:455–461, 1964.

[30] W. Fischer and K. Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, 1993.

[31] M. Fisher and A. Raman. Reducing the cost of demand uncertainty through accurate response to early sales. *Operations Research*, 44(1):87–99, 1996.

[32] G. J. Foschini and J. Salz. A basic dynamic routing problem and diffusion. *IEEE Trans. on Comm*, 26:320–327, 1978.

[33] R. G. Gallager. *Discrete stochastic processes.* Kluwer, Boston, 1996.

[34] A. J. Ganesh, N. O'Connell, and D. J. Wischik. *Big queues.* Springer, 2004.

[35] J. Gonzlez-Hernández and C. E. Villarreal. Optimal policies for constrained average-cost Markov decision processes. *TOP*, 19(1):107–120, 2011.

[36] C. Graham. Chaoticity on path space for a queueing network with selection of the shortest queue among several. *Journal of Appl. Prob*, 37:198–211, 2000.

[37] S. C. Graves and B. T. Tomlin. Process flexibility in supply chains. *Management Science*, 49:289–328, 2003.

[38] S. Gurumurthi and S. Benjaafar. Modeling and analysis of flexible queueing systems. *Management Science*, 49:289–328, 2003.

[39] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, pages 502–525, 1982.

[40] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–588, 1981.

[41] J. M. Harrison and M. J. Lopez. Heavy traffic resource pooling in parallel-server systems. *Queueing Systems*, 33:39–368, 1999.

[42] Y.-T. He and D. G. Down. On accommodating customer flexibility in service systems. *INFOR: Information Systems and Operational Research*, 47(4):289–295, 2009.

[43] O. Hernández-Lerma, J. Gonzalez-Hernández, and R. R. López-Martinez. Constrained average cost Markov control processes in Borel spaces. *SIAM Journal on Control and Optimization*, 42(2):442–468, 2003.

[44] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[45] W. Hopp, E. Tekin, and M. P. V. Oyen. Benefits of skill chaining in serial production lines with cross-trained workers. *Management Science*, 51(3):83–98, 2004.

[46] S. M. Iravani, M. P. V. Oyen, and K. T. Sims. Structural flexibility: A new perspective on the design of manufacturing and service operations. *Management Science*, 51(2):151–166, 2005.

[47] W. Jordan and S. C. Graves. Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594, 1995.

[48] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: measurements & analysis. In *Proceedings of the 9th ACM SIGCOMM.* ACM, 2009.

[49] F. P. Kelly. Effective bandwidths at multi-class queues. *Queueing systems*, 9(1-2):5–15, 1991.

[50] S. C. Kim and I. Horowitz. Scheduling hospital services: The efficacy of elective surgery quotas. *Omega*, 30:335–346, 2002.

[51] J. Kingman. The single server queue in heavy traffic. In *Proc. Cambridge Philos. Soc*, volume 57, pages 902–904. Cambridge Univ Press, 1961.

[52] S. Kunniyur and R. Srikant. *Analysis and design of an adaptive virtual queue.* ACM SIGCOMM, 2001.

[53] M. Leconte, M. Lelarge, and L. Massoulie. Bipartite graph structures for efficient balancing of heterogeneous loads. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):41–52, 2012.

[54] R. Levi and C. Shi. Revenue management of reusable resources with advanced reservations. *submitted to Operations Research*, 2011.

[55] Y. Lu and A. Radovanovic. Asymptotic blocking probabilities in loss networks with subexponential demands. *J. Appl. Probab*, 44(4):1088–1102, 2007.

[56] M. Luczak and C. McDiarmid. On the power of two choices: balls and bins in continuous time. *Journal of Appl. Prob*, 15(3):1733–1764, 2005.

[57] M. Luczak and C. McDiarmid. On the maximum queue length in the supermarket model. *Journal of Appl. Prob*, 34(2):493–527, 2006.

[58] H. Mak and Z. Shen. Stochastic programming approach to process flexibility design. *Flexible Services and Manufacturing Journal*, 21:75–91, 2009.

[59] A. Mandelbaum and M. I. Reiman. On pooling in queueing networks. *Management Science*, 44(7):971–981, 1998.

[60] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule. *Operations Research*, 52(6):836–855, 2004.

[61] J. B. Martin and Y. M. Suhuov. Fast jackson networks. *Journal of Appl. Prob*, 9(4):840–854, 1999.

[62] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *IEEE Trans. on Comm*, 47(8):1260–1267, 1999.

[63] M. Mitzenmacher. *The power of two choices in randomized load balancing.* PhD thesis, U.C. Berkeley, 1996.

[64] M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: a survey of techniques and results. *Handbook of Randomized Computing: Volume 1*, pages 255–312, 2001.

[65] G. N. Nair and R. J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM J. Control and Optimization*, 43(2):413–436, 2004.

[66] N. W. Nawijn. Look-ahead policies for admission to a single server loss system. *operations research*, 38:854–862, 1990.

[67] M. Neely, E. Modiano, and Y. Cheng. Logarithmic delay for n×n packet switches under the crossbar constraint. *IEEE/ACM Trans. Netw*, 15(3):657–668, 2007.

[68] M. I. Reiman. Some diffusion approximations with state space collapse. *Modelling and performance evaluation methodology, Springer Berlin Heidelberg*, pages 207–240, 1984.

[69] L. C. G. Rogers. Pathwise stochastic optimal control. *SIAM J. Control Optim.*, 46(3):1116–1132, 2007.

[70] A. Sahai and S. K. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link. part i: scalar systems. *IEEE Trans. Inform. Th*, 52(8):3369–3395, 2006.

[71] S. Samuels. A characterization of the poisson process. *Journal of Applied Probability*, pages 72–85, 1974.

[72] A. K. Sethi and S. P. Sethi. Flexibility in manufacturing: a survey. *International Journal of Flexible Manufacturing Systems*, 2:289–328, 1990.

[73] D. Simchi-Levi and Y. Wei. Understanding the performance of the long chain and sparse designs in process flexibility. *Operations Research*, 60(5):1125–1141, 2012.

[74] A. Simonian and J. Guibert. Large deviations approximation for fluid queues fed by a large number of on/off sources. *IEEE Journal on Selected Areas in Communications*, 13(6):1017–1027, 1995.

[75] B. L. Smith, B. M. Williams, and R. K. Oswald. Comparison of parametric and non-parametric models for traffic flow forecasting. *Cold Spring Harbor Symp. Quant. Biol.*, 10(4):303–321.

[76] G. Soundararajan, C. Amza, and A. Goel. *Database replication policies for dynamic content applications.* Proc. of EuroSys, 2006.

[77] J. Spencer, M. Sudan, and K. Xu. Queueing with future information. *to appear in Annals of Applied Probability*, 2013.

[78] K. Sriram and W. Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *Selected Areas in Communications, IEEE Journal on*, 4(6):833–846, 1986.

[79] D. A. Stanford and W. K. Grassmann. The bilingual server system: a queueing model featuring fully and partially qualified servers. *INFOR*, 31:261–277, 1993.

[80] S. Stidham. Optimal control of admission to a queueing system. *IEEE Trans. Automatic Control*, 30(8):705–713, 1985.

[81] R. Talreja and W. Whitt. Fluid models for overloaded multi-class manyservice queueing systems with fcfs routing. *Management Science*, 54(8):1513–1527, 2008.

[82] S. Tatikonda, A. Sahai, and S. K. Mitter. Stochastic linear control over a communication channel. *IEEE Trans. Automatic Control*, 49(9):1549–1561, 2004.

[83] J. N. Tsitsiklis and D. Shah. Bin packing with queues. *J. Appl. Prob*, 45(4):922–939, 2008.

[84] J. N. Tsitsiklis and K. Xu. On the power of (even a little) resource pooling. *Stochastic Systems*, 2(1):1–66, 2012.

[85] J. N. Tsitsiklis and K. Xu. Queueing system topologies with limited flexibility. In *Proceedings of the ACM SIGMETRICS*. ACM, 2013.

[86] J. Visschers, I. Adan, and G. Weiss. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems*, 70:269–298, 2012.

[87] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: an asymptotic approach. *Probl. Inf. Transm*, 32(1):20–34, 1996.

[88] R. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management*, 7:276–294, 2005.

[89] W. Whitt. Queues with superposition arrival processes in heavy traffic. *Stochastic processes and their applications*, 21(1):81–91, 1985.

[90] D. J. Wischik. Sample path large deviations for queues with many inputs. *The Annals of Applied Probability*, 11(2):379–404, 2001.

[91] K. Xu. On the power of centralization in distributed processing. Master's thesis, Massachusetts Institute of Technology, 2011. S.M. thesis.

[92] K. Xu. Necessity of future information for effective admissions control. *submitted*, 2014.

[93] K. Xu and C. W. Chan. Using future information to reduce waiting times in the emergency department. *submitted*, 2014.

[94] U. Yechiali. On optimal balking rules and toll charges in the $GI/M/1$ queuing process. *Operations Research*, 19(2):349–370, 1971.

Appendix A

Appendix: Queueing System Architectures with Limited Flexibility

A.1 Additional Proofs

A.1.1 Proof of Lemma 3.4

Proof. Fix $\lambda = (\lambda_1, \ldots, \lambda_n) \in \Lambda_n(u_n)$, and let g_n be an $(\gamma/u_n, u_n)$-expander, with $\gamma \geq \rho$. By the max-flow-min-cut theorem, and the fact that all servers have unit capacity, it suffices to show that

$$\sum_{i \in S} \lambda_i \leq |\mathcal{N}(S)|, \quad \forall S \subset I. \tag{A.1}$$

We consider two cases depending on the size of S.

1. Suppose that $|S| \leq \frac{\gamma}{u_n}n$. By the expansion property of g_n, we have that

$$\mathcal{N}(S) \geq u_n|S| \geq \sum_{i\in S}\lambda_i, \tag{A.2}$$

where the second inequality follows from that fact that $\lambda_i \leq u_n$ for all $i \in I$.

2. Suppose that $|S| > \frac{\gamma}{u_n}n$. We have

$$\mathcal{N}(S) \overset{(a)}{\geq} \gamma n \geq \rho n \overset{(b)}{\geq} \sum_{i\in S}\lambda_i, \tag{A.3}$$

where step (a) follows from the fact that S must contain a subset of size $\frac{\gamma}{u_n}n$, and step (b) from the assumption that $\sum_{i\in I}\lambda_i \leq \rho n$.

This completes the proof. □

A.1.2 Proof of Lemma 3.12

Proof. For any $S \subset I$ and $T \subset J$, define $B_{S,T}$ as the event that all neighbours S in G are contained in T. Since G is an an $(n, d_n/n)$ random bipartite graph, we have that

$$\mathbb{P}(B_{S,T}) = (1 - d_n/n)^{(n-|T|)|S|}. \tag{A.4}$$

To show that G is an expander, it suffices to verify the expansion properties of all subsets of I with size no greater than $\frac{\gamma}{\beta_n}n$. Using Eq. (A.4) and the union bound, we have

$$
\begin{aligned}
&1-\mathbb{P}\left(G \text{ is an } \left(\tfrac{\gamma}{\beta_n},\beta_n\right)\text{-expander}\right)\\
=&\mathbb{P}\left(\exists S\subset I, |S|\leq \frac{\gamma}{\beta_n}n, |\mathcal{N}(S)|\leq \beta_n|S|\right)\\
\leq&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\binom{n}{s}\binom{n}{\beta_n s}\left(1-d_n/n\right)^{(n-\beta_n s)s}\\
\overset{(a)}{\leq}&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\left(\frac{ne}{s}\right)^{s}\left(\frac{ne}{\beta_n s}\right)^{\beta_n s}\left(1-d_n/n\right)^{(n-\beta_n s)s}\\
\leq&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\left(\frac{ne}{s}\right)^{2\beta_n s}\left(1-d_n/n\right)^{(n-\beta_n s)s}\\
=&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\exp\left(s\left[2\beta_n\left(\ln n-\ln s+1\right)+\left(n-\beta_n s\right)\ln\left(1-d_n/n\right)\right]\right)\\
\leq&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\exp\left(s\left[2\beta_n\left(\ln n+1\right)+\left(n-\beta_n s\right)\ln\left(1-d_n/n\right)\right]\right)\\
\overset{(b)}{\lesssim}&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\exp\left(s\left[2\beta_n\ln n-(n-\beta_n s)d_n/n\right]\right)\\
\leq&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\exp\left(s\left[2\beta_n\ln n-(n-\gamma n)d_n/n\right]\right)\\
\overset{(c)}{=}&\sum_{s=1}^{\frac{\gamma}{\beta_n}n}\exp\left(-s\frac{1-\gamma}{2}d_n\right)\\
\leq&\frac{\exp\left(-\frac{1-\gamma}{2}d_n\right)}{1-\exp\left(-\frac{1-\gamma}{2}d_n\right)}\\
\lesssim&\exp\left(-\frac{1-\gamma}{2}d_n\right)\\
\lesssim&n^{-\frac{1}{2}}.
\end{aligned}
\tag{A.5}
$$

where step (a) follows from the bound $\binom{n}{k}\leq\left(\frac{ne}{k}\right)^k$ and the fact that $\beta_n\geq 1$, step (b) from the approximation that $\ln(1+x)\sim x$ as $x\uparrow 0$, step (c) from the assumption that $\beta_n = \frac{1-\gamma}{4} d_n \ln^{-1} n$. This completes the proof. □

A.1.3 Proof of Lemma 3.10

Proof. Lemma 3.10 is a consquence of the following standard result (c.f., [6]), by letting $\alpha\beta = \rho + \frac{1}{2}(1-\rho) = \frac{1+\rho}{2}$.

Lemma A.1. *Fix $n \geq 1$, $\beta \geq 1$ and $\alpha\beta < 1$. There exists an (α, β)-expander with maximum degree d, if*

$$d \geq \frac{1 + \log_2 \beta + (\beta + 1) \log_2 e}{-\log_2(\alpha\beta)} + \beta + 1. \tag{A.6}$$

□

A.1.4 Proof of Lemma 4.9

Proof. Let G be an $(n, d_n/n)$ random bipartite graph. We will prove the lemma by a counting argument combined with the union bound. By the max-flow min-cut theorem, $\lambda \in \mathbf{R}(G)$ is equivalent to having

$$\sum_{i \in S} \lambda_i \leq |\mathcal{N}(S)|, \quad \forall S \subset I. \tag{A.7}$$

Fix $\lambda \in \Lambda_n(u_n)$, and let

$$p_n \triangleq d_n/n. \tag{A.8}$$

We consider two cases, depending on the size of S.

1. $|S| \leq \rho n/u_n$. Since $\max_{1\leq i\leq I} \lambda_i = u_n$, we have, via the union bound, that

$$\begin{aligned}\mathbb{P}\left(\sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) \leq & \mathbb{P}\left(|\mathcal{N}(S)| \leq u_n|S|\right) \\ \leq & \sum_{\substack{B\subset J,\\ |B|=u_n|S|}} \mathbb{P}\left(\mathcal{N}(S)\subset B\right) \\ = & \binom{n}{u_n|S|}(1-p_n)^{(n-u_n|S|)|S|}. \end{aligned} \tag{A.9}$$

2. $|S| > \rho n/u_n$. Since $\sum_{i\in I}\lambda_i = \rho n$, we have that

$$\begin{aligned}\mathbb{P}\left(\sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) \leq & \mathbb{P}\left(|\mathcal{N}(S)| \leq \rho n\right) \\ \leq & \sum_{\substack{B\subset J,\\ |B|=\rho n}} \mathbb{P}\left(\mathcal{N}(S)\subset B\right) \\ = & \binom{n}{\rho n}(1-p_n)^{(1-\rho)n|S|}. \end{aligned} \tag{A.10}$$

Combining the two cases, we have that

$$\begin{aligned} & \mathbb{P}(\lambda \notin \mathbf{R}(G)) \\ = & \mathbb{P}\left(\exists S\subset I, \text{ such that } \sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) \\ \leq & \sum_{S\subset I}\mathbb{P}\left(\sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) \\ = & \sum_{\substack{S\subset I\\ |S|\leq \rho n/u_n}} \mathbb{P}\left(\sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) + \sum_{\substack{S\subset I\\ |S|>\rho n/u_n}} \mathbb{P}\left(\sum_{i\in S}\lambda_i > |\mathcal{N}(S)|\right) \\ \overset{(a)}{\leq} & \sum_{s=1}^{\rho n/u_n}\binom{n}{s}\binom{n}{u_n s}(1-p_n)^{(n-u_n s+1)s} + \sum_{s=\rho n/u_n+1}^{n}\binom{n}{s}\binom{n}{\rho n}(1-p_n)^{(1-\rho)ns} \end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{\leq} \sum_{s=1}^{\rho n/u_n} \exp\left(s\left(1+\ln\frac{n}{s}\right)+u_n s\left(1+\ln n-\ln(u_n s)\right)\right)\cdot\exp\left(-\left(\ln\frac{1}{1-p_n}\right)(n-u_n s)s\right)\\
&\quad+\sum_{s=\rho n/u_n+1}^{n} \exp\left(s\left(1+\ln\frac{n}{s}\right)+\rho n\left(1+\ln n-\ln n-\ln\rho\right)\right)\cdot\exp\left(-\left(\ln\frac{1}{1-p_n}\right)(1-\rho)ns\right)\\
&\leq \sum_{s=1}^{\rho n/u_n} \exp\left(2s\ln n+2u_n s\ln\left(\frac{n}{u_n}\right)-\left(\ln\frac{1}{1-p_n}\right)(n-u_n s)s\right)\\
&\quad+\sum_{s=\rho n/u_n+1}^{n} \exp\left(2s\ln n+\rho(1+\ln(1/\rho))n-s\left(\ln\frac{1}{1-p_n}\right)(1-\rho)n\right)\\
&\leq \sum_{s=1}^{\rho n/u_n} \exp\left(-s\left[\left(\ln\frac{1}{1-p_n}\right)(n-u_n s)-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\right]\right)\\
&\quad+\sum_{s=\rho n/u_n+1}^{\infty} \exp\left(-s\left[\left(\ln\frac{1}{1-p_n}\right)(1-\rho)n-\rho(1+\ln(1/\rho))\frac{n}{s}-2\ln n\right]\right) \qquad \text{(A.11)}
\end{aligned}
$$

where step (a) follows from Eqs. (A.9) and (A.10), and step (b) is based on the fact that $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k = \exp(k(1+\ln n-\ln k))$. We now verify that the exponents in the summands in Eq. (A.11) are negative. For the first exponent, we have that whenever $s \in \{1,\ldots,\rho n/u_n\}$,

$$
\begin{aligned}
&\left(\ln\frac{1}{1-p_n}\right)(n-u_n s)-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\\
&\overset{(a)}{\geq}\left(\ln\frac{1}{1-p_n}\right)(1-\rho)n-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\\
&\overset{(b)}{=}(1-\rho)n\ln\frac{n}{n-d_n}-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\\
&=(1-\rho)n\ln\left(1+\frac{d_n}{n-d_n}\right)-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\\
&\overset{(c)}{\sim}(1-\rho)d_n-2u_n\ln\left(\frac{n}{u_n}\right)-2\ln n\\
&\overset{(d)}{\geq}\frac{1-\rho}{2}d_n-2u_n\ln\left(\frac{n}{u_n}\right)\\
&\overset{(e)}{>}\frac{1-\rho}{4}d_n. \qquad \text{(A.12)}
\end{aligned}
$$

where step (a) on is based on the assumption that $s \leq \rho n/u_n$, step (b) on the definition that $p_n = d_n/n$, step (c) on the fact that $d_n \ll n$, and that $\ln(1+x) \sim x$ as $x \downarrow 1$, step (d) on the assumption that $d \geq \frac{4}{1-\rho}\ln n$, and step (e) on the fact that $1 \ll u_n < \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n}$.

For the second exponent in Eq. (A.11), we have that whenever $s \in \{\rho n/u_n + 1, \ldots, n\}$,

$$\begin{aligned}
&\left(\ln \frac{1}{1-p_n}\right)(1-\rho)n - \rho(1+\ln(1/\rho))\frac{n}{s} - 2\ln n \\
&\overset{(a)}{\geq} (1-\rho)n \ln \frac{1}{1-p_n} - \left(1 + \frac{1}{\rho e}\right)u_n - 2\ln n \\
&\overset{(b)}{\sim} \frac{1-\rho}{2}d_n - \left(1 + \frac{1}{\rho e}\right)u_n - 2\ln n \\
&\overset{(c)}{\sim} \frac{1-\rho}{2}d_n \\
&> \frac{1-\rho}{4}d_n, \qquad \text{(A.13)}
\end{aligned}$$

where step (a) is based on the assumption that $s \geq \rho n/u_n$, and the fact that $\max_{\rho\in(0,1)} \rho \ln \frac{1}{\rho} = 1/e$, step (b) on the fact that $p_n = d_n/n$, $d_n \ll n$, and $\ln(1+x) \sim x$ as $x \downarrow 1$, and step (c) on the assumption that $\ln n \ll d_n$ and $u_n < \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \ll d_n$.

Substituing Eqs. (A.12) and (A.13) into Eq. (A.11), we have that, for all sufficiently large n,

$$\begin{aligned}
\mathbb{P}(\lambda \notin \mathbf{R}(G)) &\leq \sum_{s=1}^{\rho n/u_n} \exp\left(-s\left[\left(\ln \frac{1}{1-p_n}\right)(n - u_n s) - 2u_n \ln\left(\frac{n}{u_n}\right) - 2\ln n\right]\right) \\
&\quad + \sum_{s=\rho n/u_n+1}^{\infty} \exp\left(-s\left[\left(\ln \frac{1}{1-p_n}\right)(1-\rho)n - \rho(1+\ln(1/\rho))\frac{n}{s} - 2\ln n\right]\right) \\
&< \sum_{s=1}^{\infty} \exp\left(-s\frac{1-\rho}{4}d_n\right) \\
&\sim \exp\left(-\frac{1-\rho}{4}d_n\right), \qquad \text{(A.14)}
\end{aligned}$$

where the last step follows from the fact that $d_n \to \infty$. Since Eq. (A.14) holds for any $\lambda \in \Lambda_n(u_n)$, this completes the proof of the lemma. □

A.1.5 Proof of Lemma 4.12

Proof. There is a total of ρb_n arriving jobs in a single batch, and for each arriving job

$$\mathbb{P}(\text{job arrives to queue } i) = \frac{\lambda_i}{\sum_{1 \leq i \leq I} \lambda_i} = \frac{\lambda_i}{\rho n} \leq \frac{u_n}{\rho n} \leq \frac{1-\rho}{16\rho n} \cdot \frac{d_n}{\ln n}, \tag{A.15}$$

for all i, where the last inequality follows from the assumption that $u_n \leq \frac{1-\rho}{16} \cdot \frac{d_n}{\ln n}$. Therefore, the distribution of M_i is stochastically dominated by the binomial random variable $\tilde{M} \overset{d}{=} \text{Bino}(\rho b_n, \frac{1-\rho}{16\rho n} \cdot \frac{d_n}{\ln n})$, with

$$\mathbb{E}\left(\tilde{M}\right) = \rho b_n \frac{1-\rho}{16\rho n} \cdot \frac{d_n}{\ln n} = \frac{1}{2}\left(\frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n}\right). \tag{A.16}$$

Let $b_n = K_n \frac{n \ln n}{d_n}$, with $K_n \geq \frac{24}{1-\rho}(\alpha+1)\ln n$. We have that

$$\begin{aligned}
&\mathbb{P}\left(M_i \geq \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n}\right) \\
\leq &\mathbb{P}\left(\tilde{M} \geq \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n}\right) \\
= &\mathbb{P}\left(\tilde{M} \geq 2\mathbb{E}\left(\tilde{M}\right)\right) \\
\overset{(a)}{\lesssim} &\exp\left(-\frac{1}{3}\mathbb{E}\left(\tilde{M}\right)\right) \\
= &\exp\left(-\frac{1}{3} \cdot \frac{1-\rho}{8} \cdot \frac{d_n}{\ln n} \cdot \frac{b_n}{n}\right) \\
= &\exp\left(-\frac{1-\rho}{24} K_n\right) \\
\overset{(b)}{\leq} &n^{-(\alpha+1)},
\end{aligned} \tag{A.17}$$

where step (a) follows from the Chernoff bound, $\mathbb{P}(X \geq (1+\epsilon)\mu) \leq \exp(-\frac{\epsilon^2}{2+\epsilon}\mu)$, where X is a binomial random variable with $\mathbb{E}(X) = \mu$ and $\epsilon > 0$, and step (b) from the condition that $K_n \geq \frac{24}{1-\rho}(\alpha+1)\ln n$. Using a union bound, Eq. (A.17) yields that

$$\begin{aligned}
&\mathbb{P}\left(\max_{1\leq i\leq I} M_i \geq \frac{1-\rho}{8}\cdot\frac{d_n}{\ln n}\cdot\frac{b_n}{n}\right)\\
\leq &\sum_{1\leq i\leq I}\mathbb{P}\left(M_i \geq \frac{1-\rho}{8}\cdot\frac{d_n}{\ln n}\cdot\frac{b_n}{n}\right)\\
\lesssim &n\cdot n^{-(\alpha+1)}\\
= &n^{-\alpha}.
\end{aligned} \tag{A.18}$$

This completes the proof of Lemma 4.12. □

Appendix B

Appendix: Queueing with Future Information

B.1 Additional Proofs

B.1.1 Proof of Lemma 5.15

Proof. **(Lemma 5.15)** Since $\lambda > 1-p$, with probability one, there exists $T < \infty$ such that the continuous-time queue length process without diversion satisfies $Q^0(t) > 0$ for all $t \geq T$. Therefore, without any diversion, all service tokens are matched with some job after time T. By the stack interpretation, π_{NOB} only diverts jobs that would not have been served, and hence does not change the original matching of service tokens to jobs. This prove the first claim.

By the first claim, since all subsequent service tokens are matched with a job after some time T, there exists some $N < \infty$, such that

$$\tilde{Q}[k] = \tilde{Q}[N] + (A[k] - A[N]) - (S[k] - S[N]) - I\left(M^{\Psi}, k\right), \tag{B.1}$$

for all $k \geq N$, where $A[k]$ and $S[k]$ are the cumulative numbers of arrival and service tokens by slot k, respectively. The second claim follows by multiplying both sides of Eq. (B.1) by $\frac{1}{k}$, and using the fact that $\lim_{k\to\infty} \frac{1}{k}A[k] = \frac{\lambda}{\lambda+1-p}$ and $\lim_{k\to\infty} \frac{1}{k}S[k] = \frac{1-p}{\lambda+1-p}$ a.s., $\tilde{Q}[k] \geq 0$ for all k, and $\tilde{Q}[N] < \infty$ a.s. □

B.1.2 Proof of Lemma 5.17

Proof. **(Lemma 5.17)**

1. Recall the point-wise diversion map, $D_P(Q,k)$, defined in Definition 5.2. For any initial sample path Q^0, let $Q^1 = D_P(Q^0, m)$ for some $m \in \mathbb{N}$. It is easy to see that, for all $k > m$, $Q^1[k] = Q^0[k] - 1$, if and only if $Q^0[s] \geq 1$ for all $s \in \{m+1, \ldots, k\}$. Repeating this argument $I(M,k)$ times, we have that

$$Q[k] = \tilde{Q}[k + m_1] = Q^0[k + m_1] - I(M, k + m_1), \tag{B.2}$$

if any only if for all $l \in \{1, \ldots, I(M, k + m_1)\}$,

$$Q^0[s] \geq l, \quad \text{for all } s \in \{m_l + 1, \ldots, k + m_1\}. \tag{B.3}$$

Note that Eq. (B.3) is implied by (and in fact, equivalent to) the definition of the m_l's (Definition 5.9), namely, that for all $l \in \mathbb{N}$, $Q^0[s] \geq l$ for all $s \geq m_l + 1$. This proves the first claim.

2. Suppose $Q[k] = Q[k-1] = 0$. Since $\mathbb{P}(Q^0[t] \neq Q^0[t-1] \mid Q^0[t-1] > 0) =$

1 for all $t \in \mathbb{N}$ (c.f., Eq. (5.2)), at least one diversion occurs on the slots $\{k-1+m_1, k+m_1\}$. If the diversion occurs on $k+m_1$, we are done. Suppose a diversion occurs on $k-1+m_1$. Then $Q^0[k+m_1] \geq Q^0[k-1+m_1]$, and hence

$$Q^0[k+m_1] = Q^0[k-1+m_1] + 1,$$

which implies that a diversion must also occur on $k+m_1$, for otherwise $Q[k] = Q[k-1]+1 = 1 \neq 0$. This shows that $k = m_i - m_1$ for some $i \in \mathbb{N}$.

Now, suppose that $k = m_i - m_1$ for some $i \in \mathbb{N}$. Let

$$k_l = \inf\left\{k \in \mathbb{N} : Q^0[k] = l, \text{ and } Q^0[t] \geq l, \forall t \geq k\right\}. \tag{B.4}$$

Since the random walk Q^0 is transient and the magnitude of its step size is at most 1, it follows that $k_l < \infty$ for all $l \in \mathbb{N}$ a.s, and that $m_l = k_l, \forall l \in \mathbb{N}$. We have

$$\begin{aligned}
&Q[k] \\
\overset{(a)}{=}&Q^0[k+m_1] - I(M, k+m_1) \\
=&Q^0[m_i] - I(M, m_i) \\
\overset{(b)}{=}&Q^0[k_i] - i \\
=&0,
\end{aligned} \tag{B.5}$$

where (a) follows from Eq. (B.2), and (b) from the fact that $k_i = m_i$. To show that $Q[k-1] = 0$, note that since $k = m_i - m_1$, an arrival must have occurred in Q^0 on slot m_i, and hence $Q^0[k-1+m_1] = Q^0[k+m_1] - 1$. Therefore, by the definition of m_i,

$$Q^0[t] - Q^0[k-1+m_1] = (Q^0[t] - Q^0[k+m_1]) + 1 \geq 0, \quad \forall t \geq k+m_1,$$

which implies that $k-1 = m_{i-1} - m_1$, and hence $Q[k-1] = 0$, in light of Eq. (B.5). This proves the claim.

3. For all $k \in \mathbb{Z}_+$, we have

$$\begin{aligned} Q[k] =& Q\left[m_{I(M,k+m_1)} - m_1\right] + \left(Q[k] - Q\left[m_{I(M,k+m_1)} - m_1\right]\right) \\ \overset{(a)}{=}& Q[k] - Q\left[m_{I(M,k+m_1)} - m_1\right] \\ \overset{(b)}{=}& Q^0[k+m_1] - Q^0\left[m_{I(M,k+m_1)}\right] \\ \overset{(c)}{=}& 0, \end{aligned} \tag{B.6}$$

where (a) follows from the second claim (c.f., Eq. (B.5)), (b) from the fact that there is no diversion on any slot in $\{I(M,k+m_1), \ldots, k+m_1\}$, and (c) from the fact that $k+m_1 \geq I(M,k+m_1)$ and Eq. (5.14).

□

B.1.3 Proof of Lemma 5.19

Proof. **(Lemma 5.19)** Since the random walk X lives in $\mathbb{Z}_+$ and can take jumps of size at most 1, it suffices to verify that

$$\mathbb{P}\left(X[k+1] = x_1 + 1 \,\middle|\, X[k] = x_1, \min_{r \geq k+1} X[r] = 0\right) = 1 - q,$$

for all $x_1 \in \mathbb{Z}_+$. We have

$$\begin{aligned}
&\mathbb{P}\left(X[k+1] = x_1 + 1 \,\middle|\, X[k] = x_1, \min_{r \geq k+1} X[r] = 0\right) \\
&= \frac{\mathbb{P}\left(X[k+1] = x_1 + 1, \min_{r \geq k+1} X[r] = 0 \,\middle|\, X[k] = x_1\right)}{\mathbb{P}\left(\min_{r \geq k+1} X[r] = 0 \,\middle|\, X[k] = x_1\right)} \\
&\overset{(a)}{=} \frac{\mathbb{P}\left(X[k+1] = x_1 + 1 \,\middle|\, X[k] = x_1\right) \cdot \mathbb{P}\left(\min_{r \geq k+1} X[r] = 0 \,\middle|\, X[k+1] = x_1 + 1\right)}{\mathbb{P}\left(\min_{r \geq k+1} X[r] = 0 \,\middle|\, X[k] = x_1\right)} \\
&\overset{(b)}{=} q \cdot \frac{h(x_1 + 1)}{h(x_1)},
\end{aligned} \tag{B.7}$$

where

$$h(x) = \mathbb{P}\left(\min_{r \geq 2} X[r] = 0 \,\middle|\, X[1] = x\right),$$

and steps (a) and (b) follow from the Markov property and stationarity of X, respectively. The values of $\{h(x) : x \in \mathbb{Z}_+\}$ satisfy the set of harmonic equations

$$h(x) = \begin{cases} q \cdot h(x+1) + (1-q) \cdot h(x-1), & x \geq 1, \\ q \cdot h(1) + 1 - q, & x = 0, \end{cases} \tag{B.8}$$

with the boundary condition

$$\lim_{x \to \infty} h(x) = 0. \tag{B.9}$$

Solving Eqs. (B.8) and (B.9), we obtain the unique solution

$$h(x) = \left(\frac{1-q}{q}\right)^x,$$

for all $x \in \mathbb{Z}_+$. By Eq. (B.7), this implies that

$$\mathbb{P}\left(X[k+1] = x_1 + 1 \middle| X[k] = x_1, \min_{r \geq k+1} X[r] = 0\right) = q \cdot \frac{1-q}{q} = 1 - q,$$

which proves the claim. □

B.1.4 Proof of Lemma 5.25

Proof. **(Lemma 5.25)** By the definition of $\overline{F}_{X_1}^{-1}$ and the strong law of large numbers (SLLN), we have

$$\lim_{k \to \infty} \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right) = \mathbb{E}\left(\mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right)\right) < \alpha, \quad a.s. \tag{B.10}$$

Denote by $S_{k,l}$ set of top l elements in $\{X_i : 1 \leq i \leq k\}$. By Eq. (B.10) and the fact that $H_k \lesssim \alpha k$ a.s., there exists $N > 0$ such that

$$\mathbb{P}\left\{\exists N, \text{ s.t. } \min S_{k,H_k} \geq \overline{F}_{X_1}^{-1}(\alpha), \forall k \geq N\right\} = 1,$$

which implies that

$$\begin{aligned} &\limsup_{k \to \infty} f\left(\{X_i : 1 \leq i \leq k\}, H_k\right) \\ \leq &\limsup_{k \to \infty} \frac{1}{k} \sum_{i=1}^{k} X_i \cdot \mathbb{I}\left(X_i \geq \overline{F}_{X_1}^{-1}(\alpha)\right) \\ = &\mathbb{E}\left(X_1 \cdot \mathbb{I}\left(X_1 \geq \overline{F}_{X_1}^{-1}(\alpha)\right)\right) \quad a.s., \end{aligned} \tag{B.11}$$

where the last equality follows from the SLLN. This proves our claim. □

B.1.5 Proof of Lemma 5.27

Proof. **(Lemma 5.27)**

We begin by stating the following fact:

Lemma B.1. *Let $\{X_i : i \in \mathbb{N}\}$ be i.i.d random variables taking values in $\mathbb{R}_+$, such that for some $a, b > 0$, $\mathbb{P}(X_1 \geq x) \leq a \cdot \exp(-b \cdot x)$ for all $x \geq 0$. Then*

$$\max_{1 \leq i \leq k} X_i = o(k), \quad a.s.,$$

as $k \to \infty$.

Proof.

$$\begin{aligned}\lim_{k\to\infty} \mathbb{P}\left(\max_{1\leq i\leq k} X_i \leq \frac{2}{b}\ln k\right) &= \lim_{k\to\infty} \mathbb{P}\left(X_1 \leq \frac{2}{b}\ln k\right)^k \\ &\leq \lim_{k\to\infty} \left(1 - a\cdot\exp(-2\ln k)\right)^k \\ &= \lim_{k\to\infty} \left(1 - \frac{a}{k^2}\right)^k \\ &= 1. \end{aligned} \tag{B.12}$$

In other words, $\max_{1\leq i\leq k} X_i \leq \frac{2}{b}\ln k$ a.s. as $k \to \infty$, which proves the claim. □

Since the $|E_i|$'s are i.i.d with $\mathbb{E}(|E_1|) = \frac{\lambda+1-p}{\lambda-(1-p)}$ (Proposition 5.20), we have that, almost surely,

$$m_K^\Psi = \sum_{i=0}^{K-1} |E_i| \sim \mathbb{E}(|E_1|) \cdot K = \frac{\lambda + 1 - p}{\lambda - (1-p)} \cdot K, \quad \text{as } K \to \infty, \tag{B.13}$$

by the strong law of large numbers. By Lemma B.1 and Eqs. (5.75), we have

$$\max_{1\leq i\leq K}|E_i| = o(K), \quad a.s., \tag{B.14}$$

as $K \to \infty$. By Eq. (B.14) and the fact that $I\left(M^{\Psi}, m_K^{\Psi}\right) = K$, we have

$$\begin{aligned}
K - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right) =& K - I\left(M^{\Psi}, m_K^{\Psi} - \max_{1\leq i\leq K}|E_i|\right) \\
&\overset{(a)}{\leq} K - I\left(M^{\Psi}, m_K^{\Psi}\right) + \max_{1\leq i\leq K}|E_i| \\
&= \max_{1\leq i\leq K}|E_i| \\
&= o(K), \quad a.s.,
\end{aligned} \tag{B.15}$$

as $K \to \infty$, where (a) follows from the fact that at most one diversion can occur in a single slot, and hence $I(M, k+m) \leq I(M,k) + m$ for all $m, k \in \mathbb{N}$. Since $\tilde{M}$ is feasible,

$$I\left(\tilde{M}, k\right) \lesssim \frac{p}{\lambda + 1 - p}\cdot k, \tag{B.16}$$

as $k \to \infty$. We have,

$$\begin{aligned}
h(K) &= \left(K - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right)\right) + \left(I\left(\tilde{M}, m_K^{\Psi}\right) - I\left(M^{\Psi}, m_K^{\Psi}\right)\right) \\
&\overset{(a)}{\lesssim} \left(K - I\left(M^{\Psi}, d\left(m_K^{\Psi}\right)\right)\right) + \frac{p}{\lambda + 1 - p}\cdot m_K^{\Psi} - K \\
&\overset{(b)}{\sim} \left(\frac{p}{\lambda + 1 - p}\cdot\frac{\lambda + 1 - p}{\lambda - (1-p)} - 1\right)\cdot K, \\
&= \frac{1-\lambda}{\lambda - (1-p)}\cdot K, \quad a.s.,
\end{aligned}$$

as $K \to \infty$, where (a) follows from Eqs. (B.13) and (B.16), (b) from Eqs. (B.13) and (B.15), which completes the proof. □

Appendix C

Appendix: Necessity of Future Information

C.1 Additional Proofs

C.1.1 Proof of Lemma 6.4

Proof. Recall from Eq. (6.6) that $S(s,t)$ is defined as the difference between the numbers of arrivals and service tokens in $[s,t)$. Since the arrival and service tokens processes are independent Poisson processes with rate λ and $1-p$, respectively, it is not difficult to verify that

$$S(s,t) \overset{d}{=} \sum_{m=1}^{N_{s,t}} X_m, \tag{C.1}$$

where $N_{s,t}$ is a Poisson random variable with mean $(\lambda+1-p)(t-s)$, which corresponds to the total number of events in $[t,s)$, and the X_ms are i.i.d., with

$$X_1 = \begin{cases} 1, & \text{w.p. } \frac{\lambda}{\lambda+1-p}, \\ -1, & \text{otherwise}, \end{cases} \tag{C.2}$$

By Eq. (C.1), and the fact that $\lim_{B\to\infty} \frac{N_{s,s+B}}{B} = \lambda + 1 - p$ almost surely, Claim 1 follows from a variation of the standard Functional Law of Large Numbers (FLLN) for the sum of bounded i.i.d. random variables. Claim 3 follows from the Weak Law of Large Numbers applied to the sum of i.i.d. Poisson random variables, and our assumption that $W_\lambda \to \infty$ as $\lambda \to 1$ (Eq. (6.11)). Finally, Claim 2 follows from the Markov's inequality, in the same way as in Eq. (6.22), by noting that $\mathbb{E}(Q(0)) = q_\lambda$ under a optimal stationary policy. □

C.1.2 Proof of Lemma 6.5

Proof. Based on the stationarity of $\mathcal{A}$ and $\mathcal{S}$, and the assumption that $B = kw_\lambda$ and $q_\lambda \ll w_\lambda$, it suffices for us to show, that for any $a, b > 0$, there exists $\gamma > 0$

$$\mathbb{P}\left(S(0, aw_\lambda) \le -bw_\lambda\right) \succcurlyeq \exp(-\gamma w_\lambda), \quad \text{as } \lambda \to 1. \tag{C.3}$$

By definition, the distribution of $S(0,t)$ can be written as

$$S(0,t) \overset{d}{=} A_{\lambda t} - D_{(1-p)t}, \tag{C.4}$$

where $A_{\lambda t}$ and $D_{(1-p)t}$ are independent Poisson random variables with mean λt and $(1-p)t$, respectively. The following lemma follows from the standard large-deviation principles of Poisson random variables, and its proof is omitted.

Lemma C.1. *Let D_x be a Poisson random variable with mean x. Then, for all $c_1 > 0$, there exists $c_2 > 0$, such that*

$$\mathbb{P}\left(D_x \geq c_1 x\right) \succeq \exp(-c_2 x), \quad \textit{as } x \to \infty. \tag{C.5}$$

Combining Lemma C.1 and the fact that $w_\lambda \to \infty$ as $\lambda \to 1$, we have that there exists $\gamma > 0$, such that

$$\mathbb{P}\left(D_{(1-p)aw_\lambda} \geq (b+2a)w_\lambda\right) \succeq \exp(-\gamma w_\lambda) \tag{C.6}$$

as $\lambda \to 1$. We have that

$$\begin{aligned}
&\mathbb{P}\left(S(0, aw_\lambda) \leq -bw_\lambda\right) \\
\geq &\mathbb{P}\left(\left\{A_{\lambda aw_\lambda} < 2aw_\lambda\right\} \cap \left\{D_{(1-p)aw_\lambda} \geq (b+2a)w_\lambda\right\}\right) \\
\overset{(a)}{=} &\mathbb{P}\left(A_{\lambda aw_\lambda} < 2aw_\lambda\right) \mathbb{P}\left(D_{(1-p)aw_\lambda} \geq (b+2a)w_\lambda\right) \\
\overset{(b)}{\geq} &\mathbb{P}\left(A_{\lambda aw_\lambda} < 2\lambda aw_\lambda\right) \mathbb{P}\left(D_{(1-p)aw_\lambda} \geq (b+2a)w_\lambda\right) \\
\overset{(c)}{\geq} &\frac{1}{2}\mathbb{P}\left(D_{(1-p)aw_\lambda} \geq (b+2a)w_\lambda\right) \\
\overset{(d)}{\succeq} &\exp(-\gamma w_\lambda),
\end{aligned} \tag{C.7}$$

as $\lambda \to 1$, where step (a) follows from the independence between $A_{\lambda aw_\lambda}$ and $D_{(1-p)aw_\lambda}$, (b) from the fact that $\lambda < 1$, (c) from the Markov's inequality, and (d) from Eq. (C.6). This proves Eq. (6.5), and hence Lemma 6.5. □

C.1.3 Proof of Lemma 6.6

Proof. For Claim 1, observe that each of the event concerns only the behavior of the arrival and service token processes over an interval, and that these intervals are disjoint from each other. Claim 1 follows by noting that both $\mathcal{A}$ and $\mathcal{S}$ are Poisson processes and hence memoriless. For Claim 2, because the policy has access to a lookahead window of length w_λ, the queue length at time t is hence $\mathcal{F}_{t+w_\lambda}$ measurable, where $\mathcal{F}$ is the natural filtration induced by the input processes. The claim follows again from the memoryless property of Poisson processes. Claim 3 follows from the same arguments as for Claim 2. □

C.1.4 Proof of Lemma 6.8

Proof. Consider the sequence of optimal stationary policies, $\{\pi_\lambda\}$. Let ϕ be defined as in Eq. (6.28). Fix $\phi > 0$, and let

$$K = U_3 + \phi w_\lambda \overset{(a)}{=} (k + \phi + 2) w_\lambda, \tag{C.8}$$

where step (a) follows from the fact that $U_3 = B + 2w_\lambda$ and $B = kw_\lambda$. The main idea for the proof is based on the following observation: conditional on $\cap_{i=1}^5 \mathcal{E}_i$, the queue length process, $Q(t)$, would have reached zero before time K, even if *no* diversion had been made in $[0, K)$ (illustrated in Figure 6-2). Therefore, each diversion made in $[U_1, U_2)$ will necessarily lead to a *waste service token* in $[0, K)$, and hence

$$\mathbb{P}\left(\mathcal{J}(K) \geq \tau B \,\middle|\, \cap_{i=1}^5 \mathcal{E}_i\right) \geq \mathbb{P}\left(Y \geq \tau B \,\middle|\, \cap_{i=1}^5 \mathcal{E}_i\right). \tag{C.9}$$

We next give a lower bound on the above probability, as follows:

$$\begin{aligned}
&\mathbb{P}\left(\mathcal{J}(K)\geq\tau B\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
\geq&\mathbb{P}\left(\mathcal{J}(K)\geq\tau B,\cap_{i=3}^{5}\mathcal{E}_i\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
=&\mathbb{P}\left(\mathcal{J}(K)\geq\tau B\middle|\cap_{i=1}^{5}\mathcal{E}_i\right)\mathbb{P}\left(\cap_{i=3}^{5}\mathcal{E}_i\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
\overset{(a)}{\geq}&\mathbb{P}\left(Y\geq\tau B\middle|\cap_{i=1}^{5}\mathcal{E}_i\right)\mathbb{P}\left(\cap_{i=3}^{5}\mathcal{E}_i\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
=&\mathbb{P}\left(Y\geq\tau B,\cap_{i=3}^{5}\mathcal{E}_i\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
\overset{(b)}{=}&\mathbb{P}\left(\mathcal{E}_5\right)\mathbb{P}\left(Y\geq\tau B,\mathcal{E}_3\cap\mathcal{E}_4\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)\\
\geq&\mathbb{P}\left(\mathcal{E}_5\right)\left(\mathbb{P}\left(Y\geq\tau B\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)+\mathbb{P}\left(\mathcal{E}_3\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)+\mathbb{P}\left(\mathcal{E}_4\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)-2\right)\\
\geq&\mathbb{P}\left(\mathcal{E}_5\right)\left(\mathbb{P}\left(Y\geq\tau B\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)+\mathbb{P}\left(\mathcal{E}_3\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)+\frac{\mathbb{P}\left(\mathcal{E}_4\right)+\mathbb{P}\left(\mathcal{E}_1\cap\mathcal{E}_2\right)-1}{\mathbb{P}\left(\mathcal{E}_1\cap\mathcal{E}_2\right)}-2\right)\\
\overset{(c)}{=}&\mathbb{P}\left(\mathcal{E}_5\right)\left(\mathbb{P}\left(Y\geq\tau B\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)+\mathbb{P}\left(\mathcal{E}_3\right)+\frac{\mathbb{P}\left(\mathcal{E}_4\right)-1}{\mathbb{P}\left(\mathcal{E}_1\cap\mathcal{E}_2\right)}-1\right)
\end{aligned}\tag{C.10}$$

where step (a) follows from Eq. (C.9), and (b) and (c) from the independence between $\mathcal{E}_5$ and $\mathcal{E}_1\cap\mathcal{E}_2$, and between $\mathcal{E}_3$ and $\mathcal{E}_1\cap\mathcal{E}_2$, respectively (Lemma 6.6). We have also used the inequality that $\mathbb{P}(A\cap B)\geq\mathbb{P}(A)+\mathbb{P}(B)-1$, for any events A and B.

By Claim 3 of Lemma 6.4, we have that

$$\lim_{\lambda\to1}\mathbb{P}\left(\mathcal{E}_3\right)=\lim_{\lambda\to1}\mathbb{P}\left(\mathcal{E}_4\right)=1.\tag{C.11}$$

Combing the assumption (Eqs. (6.35))

$$\liminf_{\lambda\to1}\mathbb{P}\left(Y\geq\tau B\middle|\mathcal{E}_1\cap\mathcal{E}_2\right)=q>0.\tag{C.12}$$

with Eqs. (C.10) and (C.11), we have that there exists $\tilde{\lambda} \in (0,1)$, such that

$$\begin{aligned}\mathbb{P}\left(\mathcal{J}(K) \geq \tau B \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) \geq & \mathbb{P}(\mathcal{E}_5)\, \mathbb{P}\left(Y \geq \tau B \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) \\ \geq & \mathbb{P}(\mathcal{E}_5)\, q,\end{aligned} \tag{C.13}$$

for all $\lambda \in \left(\tilde{\lambda}, 1\right)$. We have that

$$\begin{aligned}\mathbb{E}(\mathcal{J}(K)) \geq & \tau B \cdot \mathbb{P}(\mathcal{J}(K) \geq \tau B) \\ \geq & \tau B \cdot \mathbb{P}(\mathcal{J}(K) \geq \tau B, \mathcal{E}_1 \cap \mathcal{E}_2) \\ = & \tau B \cdot \mathbb{P}\left(\mathcal{J}(K) \geq \tau B \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right) \cdot \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \\ \overset{(a)}{\gtrsim} & B \mathbb{P}(\mathcal{E}_5)\, \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \\ \overset{(b)}{\gtrsim} & B \mathbb{P}(\mathcal{E}_5) \\ \overset{(c)}{\gtrsim} & B \exp(-\gamma w_\lambda),\end{aligned} \tag{C.14}$$

for some $\gamma > 0$, as $\lambda \to 1$, where step (a) follows from Eq. (C.13), (b) from Claims 1 and 2 of Lemma 6.4 and the independence of the events $\mathcal{E}_1$ and $\mathcal{E}_2$ (Claim 1 of Lemma 6.6), i.e., that

$$\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) = \mathbb{P}(\mathcal{E}_1)\, \mathbb{P}(\mathcal{E}_2) \geq \frac{5}{6}\theta, \tag{C.15}$$

and (c) from Lemma 6.5. This proves Lemma 6.8, by setting $a = k + \phi + 2$. □

Appendix D

Appendix: Decentralized Partial Resource Pooling

D.1 Additional Proofs

D.1.1 Proposition 7.9

Proof. Because each Markov chain $W_i(\cdot)$ is ergodic and has a finite number of states, we have that $W_i(t)$ converges to its steady-state distribution as $t \to \infty$, uniformly across all states. In particular, there exists $\tilde{s} > 0$ such that

$$\sup_{w \in \{1,\ldots,M\}} \left\| \mathbf{d}\left(W_i(w,s)\right) - \pi \right\|_\infty \le \epsilon/2, \quad \forall s \ge \tilde{s}. \tag{D.1}$$

Fix $s \ge \tilde{s}$. Since all chains are independent, by the Chernoff bound, we have that there exist ω_1 and $\omega_2 > 0$, such that for all $w \in \{1,\ldots,M\}$,

$$\sup_{\mathbf{w}_0 \in \{1,\ldots,M\}^n} \mathbb{P}\left(\left| \mathbf{h}_w(W(\mathbf{w}_0,s) - \pi_w \right| \ge \epsilon \right) \le \omega_1 \exp(-\omega_2 n), \quad \forall n \in \mathbb{N}. \tag{D.2}$$

Applying the union bound to the above equation over all states yields that

$$
\begin{aligned}
&\sup_{\mathbf{w}_0\in\{1,\dots,M\}^n} \mathbb{P}\left(\left\|\mathrm{h}\left(W(\mathbf{w}_0,s)\right)-\pi\right\|_\infty \geq \epsilon\right)\\
\leq &\sup_{\mathbf{w}_0\in\{1,\dots,M\}^n} \mathbb{P}\left(\sup_{w\in\{1,\dots,W\}} |\mathbf{h}_w(W(\mathbf{w}_0,s)-\pi_w| \geq \epsilon\right)\\
\leq &\sup_{\mathbf{w}_0\in\{1,\dots,M\}^n} \sum_{w=1}^{M} \mathbb{P}\left(|\mathbf{h}_w(W(\mathbf{w}_0,s)-\pi_w| \geq \epsilon\right)\\
\leq &M\omega_1\exp(-\omega_2 n), \quad \forall n\in\mathbb{N},
\end{aligned}
\tag{D.3}
$$

which proves our claim, by setting $\gamma_1 = M\omega_1$, and $\gamma_2 = \omega_2$. □

D.1.2 Lemma 7.8

Proof. For all $n\in\mathbb{N}$, we can write X_{an} as the sum of n i.i.d. Poisson random variables $\{Y_j : 1 \leq j \leq n\}$

$$X_{\alpha n} \overset{d}{=} \sum_{j=1}^{n} Y_j, \tag{D.4}$$

where $\mathbb{E}(Y_1) = \alpha$. From the basic theory of large deviations for the sum of i.i.d. Poisson random variables, we have that, for all $\gamma > \alpha$

$$\mathbb{P}\left(\sum_{j=1}^{n} Y_j \geq \gamma n\right) \leq \exp\left(-l_\alpha(\gamma)n\right), \quad \forall n\in\mathbb{N}, \tag{D.5}$$

where $l_\alpha(\cdot)$ is the Legendre transform of the moment-generating-function of Y_1, which is a Poisson distribution with mean α,

$$l_\alpha(\gamma) = \gamma(\ln(\gamma/\alpha)-1)+\alpha. \tag{D.6}$$

Note that for all $\alpha > 0$, $l_\alpha(\gamma) > 0$ for all $\gamma > 0$, and its derivative is given by

$$\frac{d}{d\gamma} l_\alpha(\gamma) = \ln \frac{\gamma}{\alpha}, \tag{D.7}$$

which is greater than zero for all $\gamma > \alpha$ and strictly increasing in γ. We hence conclude that, for all $\tilde{\gamma} > \alpha$, there exists $c > 0$, so that

$$l_\alpha(x) \geq c\gamma, \quad \forall \gamma > \tilde{\gamma}, \tag{D.8}$$

which, when substited into Eq. (D.5), implies that

$$\mathbb{P}\left(\sum_{j=1}^{n} Y_j \geq \gamma n\right) \leq \exp\left(-c\gamma n\right), \quad \forall \gamma \geq \tilde{\gamma}, n \in \mathbb{N}. \tag{D.9}$$

Fix $x \geq \beta n$. By Eqs. (D.9) and (D.4), where we let $\tilde{\gamma} = \beta$, and $\gamma = x/n = \beta + \frac{x-\beta n}{n}$, we have that there exists $c > 0$, so that

$$\begin{aligned}
&\mathbb{P}\left(X_{\alpha n} \geq x\right) \\
=&\mathbb{P}\left(\sum_{j=1}^{n} Y_j \geq x\right) \\
=&\mathbb{P}\left(\sum_{j=1}^{n} Y_j \geq \left(\beta + \frac{x-\beta n}{n}\right) n\right) \\
\overset{(a)}{\leq}& \exp\left(-c\left(\beta + \frac{x-\beta n}{n}\right) n\right) \\
\leq& \exp\left(-cx\right) \quad \forall x \geq \beta n, n \in \mathbb{N}.
\end{aligned} \tag{D.10}$$

where step (a) follows from Eq. (D.9), and the fact that $\beta + \frac{x-\beta n}{n} \geq \beta > \alpha$. This proves the claim in Eq. (7.30). Eq. (7.31) follows from Eq. (7.30) directly via an elementary calculation. □

D.1.3 Lemma 7.11

Proof. Let N_i be the number of events occurring to chain i during $[0, T_1)$. Since all $W_i(\cdot)$ are uniformized, we have that the N_j are i.i.d. Poisson random variables with mean ξt.

Fix $j \in \mathbb{Z}_+$, and $\mathbf{w}_0 \in \mathcal{S}_\epsilon$. Let $f_j = \mathbb{P}(N_1 = j)$. Denote by $C_{j,w} \subset \{1, \ldots, n\}$ the set of chains that:

1. are in state w at time 0, and

2. made j state transitions by time t,

and let $|C_{j,w}|$ be its cardinality. Note that there are $n\mathbf{h}_w(\mathbf{w}_0)$ chains in state w at 0, and each of them has a probability of f_j to have made j state transitions. Based on this reasoning, we know that $|C_{j,w}|$ is a binomial random variable with parameter $(n\mathbf{h}_w(\mathbf{w}_0), f_j)$. Furthermore, since $\mathbf{w}_0 \in \mathcal{S}_\epsilon$, we have that

$$|\mathbf{h}_w(\mathbf{w}_0) - \pi_w| \le \epsilon. \tag{D.11}$$

We conclude that, by the Chernoff bound, for any $\epsilon_1 > 0$, there exists α_2, so that

$$\mathbb{P}\left(\frac{1}{n}||C_{j,w}| - \pi_w f_j n| > \epsilon f_j + \epsilon_1\right) \le \exp(-\alpha_2 n), \quad \forall n \in \mathbb{N}, j \in \{1, \ldots, j^*\}, w \in \{1, \ldots, M\}. \tag{D.12}$$

Recall that P is the transition matrix for the embedded discrete-time Markov chain of W_i. By the definition of $C_{j,w}$, for all $i \in C_{j,w}$, $W_i(t)$ is distributed according to

$$\mathbb{P}\left(W_i(t) = x\right) = (P^j)_{w,x}, \tag{D.13}$$

where $(P^j)_{w,x}$ is the entry on the wth row and xth column of the matrix P^j, and $W_i(t)$ is independent from all other chains. Let $D_{j,w,x}$ be the number of chains in

$C_{j,w}$ whose state at time t is x, i.e.,

$$D_{j,w,x} = \sum_{i \in C_{j,w}} \mathbf{I}(W_i(t) = x). \tag{D.14}$$

We have, from Eq. (D.13), that for each realization of $|C_{j,w}|$, $D_{j,w,x}$ has a binomial distribution with parameters $(|C_{j,w}|, (P^j)_{w,x})$. Combining this fact with Eq. (D.12), we can show, via a Chernoff bound, that for all $\epsilon, \epsilon_2 > 0$ and $\mathbf{w}_0 \in \mathcal{S}_\epsilon$, there exists $\alpha_3 > 0$, so that

$$\mathbb{P}\left(\frac{1}{n}|D_{j,w,x} - \pi_w f_j(P^j)_{w,x} n| > \epsilon f_j(P^j)_{w,x} + \epsilon_2 \,\middle|\, W(0) = \mathbf{w}_0\right) \le \exp(-\alpha_3 n), \tag{D.15}$$

for all $n \in \mathbb{N}$, $j \in \{1, \dots, j^*\}$, and $x, w \in \{1, \dots, M\}$. Letting $\epsilon_2 = \epsilon$, and using the fact that $f_j(P^j)_{w,x} \le 1$, Eq. (D.15) can be further simplified to yield that, for all $\epsilon > 0$ and $\mathbf{w}_0 \in \mathcal{S}_\epsilon$, there exist $\alpha_3 > 0$, so that

$$\mathbb{P}\left(\frac{1}{n}|D_{j,w,x} - \pi_w f_j(P^j)_{w,x} n| > 2\epsilon \,\middle|\, W(0) = \mathbf{w}_0\right) \le \exp(-\alpha_3 n), \tag{D.16}$$

We next argue that, with high probability, most chains have no more than a certain number of transitions. Let

$$j^* = \min\{j : \mathbb{P}(N_1 \ge j) \le \chi/10\}. \tag{D.17}$$

Since the N_is are i.i.d. Poisson random variables, by the Chernoff bound, there exists $\theta > 0$, such that

$$\mathbb{P}\left(\sum_{i=1}^{n} \mathbf{I}(N_i \ge j^*) \ge (2\chi/10)n\right) \le \exp(-\theta n), \quad \forall n \in \mathbb{N}. \tag{D.18}$$

or, equivalently, that

$$\mathbb{P}\left(\sum_{w=1}^{M}\sum_{j\geq j^*+1} C_{j,w} \geq (2\chi/10)n\right) \leq \exp(-\theta n), \quad \forall n \in \mathbb{N}. \tag{D.19}$$

We have that

$$\begin{aligned}\mathbf{h}_v(W(t)) =& \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}(W_i(t) = v)\\ =& \frac{1}{n}\sum_{w=1}^{M}\sum_{j\geq 0} D_{w,v,j}\\ =& \left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j=0}^{j^*} D_{w,v,j}\right) + \left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j\geq j^*+1} D_{w,v,j}\right).\end{aligned} \tag{D.20}$$

Because $\sum_{w=1}^{M}\sum_{j\geq j^*+1} D_{w,v,j} \leq \sum_{w=1}^{M}\sum_{j\geq j^*+1} C_{w,j}$, by Eq. (D.19) and (D.20), we have that

$$\mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j=0}^{j^*} D_{w,v,j}\right)\right| \geq 2\chi/10\right) \leq \exp(-\theta n), \quad \forall n \in \mathbb{N}. \tag{D.21}$$

We now combine Eq. (D.21) and (D.16) via the union bound, over all choices of w

and j. Using the fact that $f_j(P^j)_{w,v} \le 1$, we have that, for all $\epsilon > 0$ and $\mathbf{w}_0 \in \mathcal{S}_\epsilon$,

$$\begin{aligned}
&\mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \left(\sum_{w=1}^{M}\sum_{j=1}^{j^*}\pi_w f_j(P^j)_{w,v}\right)\right| \ge 2\chi/10 + 2M(j^*+1)\epsilon\right) \\
&\overset{(a)}{\le} \mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j=1}^{j^*} D_{w,v,j}\right)\right| \ge 2\chi/10\right) \\
&\quad + \mathbb{P}\left(\left|\left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j=1}^{j^*} D_{w,v,j}\right) - \left(\sum_{w=1}^{M}\sum_{j=1}^{j^*}\pi_w f_j(P^j)_{w,v}\right)\right| \ge 2M(j^*+1)\epsilon \,\middle|\, W(0) = \mathbf{w}_0\right) \\
&\overset{(b)}{\le} \mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \left(\frac{1}{n}\sum_{w=1}^{M}\sum_{j=0}^{j^*} D_{w,v,j}\right)\right| \ge 2\chi/10\right) \\
&\quad + \sum_{w=1}^{M}\sum_{j=1}^{j^*}\mathbb{P}\left(\left|\left(\frac{1}{n}D_{w,v,j}\right) - \pi_w f_j(P^j)_{w,v}\right| \ge 2\epsilon \,\middle|\, W(0) = \mathbf{w}_0\right) \\
&\overset{(c)}{\le} \exp(-\theta n) + M(j^*+1)\exp\left(-\alpha_3 n\right) \\
&\le \nu_1 \exp(-\nu_2 n),
\end{aligned} \tag{D.22}$$

where $\nu_1 = 1 + M(j^* + 1)$ and $\nu_2 = \min\{\theta, \alpha_3\}$. Step (a) follows from the triangle inequality, (b) from the union bound, and (c) from Eqs. (D.16) and (D.21).

Recall that π is the steady-state distribution for W_1, and hence we have, for all $v \in \{1, \dots, M\}$ and $j \in \mathbb{Z}_+$, that

$$\sum_{w=1}^{M} \pi_w (P^j)_{w,v} = \pi_v, \tag{D.23}$$

which yields that

$$\begin{aligned}\sum_{w=1}^{M}\sum_{j=0}^{j^*}\pi_w f_j (P^j)_{w,v} =& \sum_{j=0}^{j^*} f_j \left(\sum_{w=1}^{M} \pi_v (P^j)_{w,v}\right)\\ =&\pi_v \sum_{j=1}^{j^*} f_j = \pi_v \mathbb{P}\left(N_1 \le j^*\right)\\ \overset{(a)}{\in}& \left[(1-\chi/10)\pi_v\,,\,\pi_v\right],\end{aligned} \tag{D.24}$$

where step (a) follows from the definition of j^*, in Eq. (D.17). From Eq. (D.22) and (D.24), we have that

$$\begin{aligned}&\mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \pi_v\right| \ge 3\chi/10 + 2M(j^*+1)\epsilon\right)\\ \le&\mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \left(\sum_{w=1}^{M}\sum_{j=1}^{j^*}\pi_w f_j (P^j)_{w,v}\right)\right| \ge 2\chi/10 + 2M(j^*+1)\right)\\ &+\mathbb{P}\left(\left|\left(\sum_{w=1}^{M}\sum_{j=1}^{j^*}\pi_w f_j (P^j)_{w,v}\right) - \pi_v\right| \ge \chi/10\right)\\ \le&\nu_1 \exp(-\nu_2 n) + \exp(-\theta n)\\ \le&\max\{\nu_1, 1\}\exp(-\min\{\theta,\nu_2\}n).\end{aligned} \tag{D.25}$$

Applying a union bound to Eq. (D.25) over all choices of v, and setting $\chi = \epsilon$, we conclude that, for all $\epsilon > 0$, there exists ν_1, ν_2 and $\theta > 0$, so that

$$\begin{aligned}&\mathbb{P}\left(\left\|\mathbf{h}(W(\mathbf{w}_0,t)) - \pi\right\|_\infty \ge [3/10 + 2M(j^*+1)]\epsilon\right)\\ \le&\sum_{v=1}^{M}\mathbb{P}\left(\left|\mathbf{h}_v(W(\mathbf{w}_0,t)) - \pi_v\right| \ge [3/10 + 2M(j^*+1)]\epsilon\right)\\ \le&M\max\{\nu_1, 1\}\exp(-\min\{\theta,\nu_2\}n),\end{aligned} \tag{D.26}$$

for all $n \in \mathbb{N}$. This proves our claim, by letting $a = M\max\{\nu_1, 1\}$, $b = \min\{\theta, \nu_2\}$, and

$c = 3/10 + 2M(j^* + 1)$. □

D.1.4 Lemma 7.13

Proof. Because the Markov chains $\{W_i(\cdot)\}$ are time-homogeneous, it suffices to prove our claim for $k = 1$. Fix $\mathbf{w}_0 \in \{1, \dots, M\}^n$. Let $\mathcal{S}_\epsilon = \{\mathbf{w} \in \{1, \dots, M\}^n : \|\mathbf{h}(\mathbf{w}) - \pi\|_\infty \leq \epsilon\}$, as defined as in Eq. (7.33). We have that, for all $\xi \in (0,1)$,

$$
\begin{aligned}
&\mathbb{P}\left(\overline{\mathcal{W}_g} \,\middle|\, W(t_{k-1}) = \mathbf{w}_0\right) \\
=&1 - \mathbb{P}\left(\sup_{t \in [B_1, B_1 + B_2)} \|\mathbf{h}\left(W(\mathbf{w}_0, t)\right) - \pi\|_\infty \leq \delta \,\middle|\, W(t_{k-1}) = \mathbf{w}_0\right) \\
\overset{(a)}{\leq}& 1 - \mathbb{P}\left(\|\mathbf{h}(W(\mathbf{w}_0, B_1) - \pi\|_\infty \leq \xi\delta\right)\left(\inf_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [B_1, B_1 + B_2)} \|\mathbf{h}(W(t) - \pi\|_\infty \leq \delta \,\middle|\, W(B_1) = w'\right)\right) \\
\leq& (1 - \mathbb{P}\left(\|\mathbf{h}(W(\mathbf{w}_0, B_1) - \pi\|_\infty \leq \xi\delta\right)) \\
&+ \left(1 - \inf_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [B_1, B_1 + B_2)} \|\mathbf{h}(W(t) - \pi\|_\infty \leq \delta \,\middle|\, W(B_1) = w'\right)\right) \\
=&\mathbb{P}\left(\|\mathbf{h}(W(\mathbf{w}_0, B_1) - \pi\|_\infty > \xi\delta\right) + \sup_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [B_1, B_1 + B_2)} \|\mathbf{h}(W(t) - \pi\|_\infty > \delta \,\middle|\, W(B_1) = w'\right),
\end{aligned}
\tag{D.27}
$$

where step (a) follows from the Markov properties of the W_i. By Proposition 7.10, for any $\delta > 0$, there exist ξ, β_1 and $\beta_2 > 0$, so that

$$
\begin{aligned}
&\sup_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [B_1, B_1 + B_2)} \|\mathbf{h}(W(t) - \pi\|_\infty > \delta \,\middle|\, W(B_1) = w'\right) \\
\overset{(a)}{=}& \sup_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [0, B_2)} \|\mathbf{h}(W(t) - \pi\|_\infty > \delta \,\middle|\, W(0) = w'\right) \\
\leq& \beta_1 \exp(-\beta_2 n), \quad \forall n \in N,
\end{aligned}
\tag{D.28}
$$

where in step (a) we used the time-homogeneity of the Markov chains $\{W_i(\cdot)\}$. By Proposition 7.9, for all ξ and $\delta > 0$, there exists $\tilde{B}_1, \gamma_1$ and γ_2, so that for all $B_1 \geq \tilde{B}_1$,

$$\mathbb{P}\left(\|\mathbf{h}(W(\mathbf{w}_0, B_1) - \pi\|_\infty > \xi\delta \right) \leq \gamma_1 \exp(-\gamma_2 n), \quad \forall n \in \mathbb{N}. \tag{D.29}$$

Substituting Eqs. (D.28) and (D.29) into Eq. (D.27), we have that, for all B_2 and $\delta > 0$, there exists $\xi, \beta_1, \beta_2, \gamma_1$ and $\gamma_2 > 0$, so that

$$\begin{aligned}
&\mathbb{P}\left(\overline{W_g} \,\middle|\, W(t_{k-1}) = \mathbf{w}_0\right) \\
\leq &\mathbb{P}\left(\|\mathbf{h}(W(\mathbf{w}_0, B_1) - \pi\|_\infty > \xi\delta \right) + \sup_{\mathbf{w}' \in \mathcal{S}_{\xi\delta}} \mathbb{P}\left(\sup_{t \in [B_1, t_1)} \|\mathbf{h}(W(t) - \pi\|_\infty > \delta \,\middle|\, W(B_1) = w' \right) \\
\leq &\beta_1 \exp(-\beta_2 n) + \gamma_1 \exp(-\gamma_2 n), \quad \forall n \in \mathbb{N},
\end{aligned} \tag{D.30}$$

which holds for all $\mathbf{w}_0$. This proves our claim, by letting $\zeta_1 = \max\{\beta_1, \gamma_1\}$ and $\zeta_2 = \min\{\beta_2, \gamma_2\}$. □

Lemma D.1. *For all $j \in \mathbb{Z}_+$, we have that*

$$\left(Q^M(t) \,\middle|\, Q^M(0) = j, T_0 \leq t\right) \preccurlyeq Q^M_\infty. \tag{D.31}$$

For the case of $T_0 > t$, we claim that, for all $t \geq 0$,

$$\mathbb{P}\left(Q^M(t) \geq l \,\middle|\, Q^M(0) = j, T_0 > t\right) \leq \mathbb{P}\left(j + A_t \geq l\right), \quad \forall l \in \mathbb{Z}_+, \tag{D.32}$$

which follows from the fact that, by definition,

$$\mathbb{P}\left(Q^M(t) \leq A_t + j \,\middle|\, Q^M(0) = j\right) = 1, \quad \forall t \geq 0, a.s. \tag{D.33}$$

We have that, for all $t \geq 0$

$$\begin{aligned}
&\mathbb{P}\left(T_0 > t \,\middle|\, Q^M(0) = an\right) \\
=&\mathbb{P}\left(\inf_{0\leq s\leq t} A_s - S_s \geq -an\right) \\
\leq&\mathbb{P}\left(A_t - S_t \geq -an\right) \\
=&\mathbb{P}\left(\sum_{i=1}^{n}\left(X_i^{t\rho'} - Y_i^t\right) \geq -an\right),
\end{aligned} \tag{D.34}$$

where the $X_i^{t\rho'}$s and Y_i^ts are independent Poisson random variables, with mean $t\rho'$ and t, respectively. In particular, $\mathbb{E}(X_1^{t\rho'} - Y_1^t) = -(1-\rho')t$. Since $\rho' < 1$, by the Chernoff bound, we have, from Eq. (D.34), that for all $b > a/(1-\rho')$, there exists $\psi > 0$, so that

$$\mathbb{P}\left(T_0 > b \,\middle|\, Q^M(0) = an\right) \leq \mathbb{P}\left(\sum_{i=1}^{n}\left(X_i^{b\rho'} - Y_i^b\right) \geq -an\right) \leq \exp(-\psi n), \tag{D.35}$$

for all $n \in \mathbb{N}$.

We have that

$$\begin{aligned}
&\mathbb{P}\left(Q^M(b) \geq l \,\middle|\, Q^M(0) = an\right) \\
\leq&\mathbb{P}\left(Q^M(b) \geq l \,\middle|\, Q^M(0) = an, T_0 \leq b\right)\mathbb{P}\left(T_0 \leq b \,\middle|\, Q^M(0) = an\right) \\
&+ \mathbb{P}\left(Q^M(b) \geq l \,\middle|\, Q^M(0) = an, T_0 > b\right)\mathbb{P}\left(T_0 > b \,\middle|\, Q^M(0) = an\right) \\
\overset{(a)}{\leq}&\mathbb{P}\left(Q_\infty^M \geq l\right) + \mathbb{P}\left(Q^M(b) \geq l \,\middle|\, Q^M(0) = an, T_0 > b\right)\mathbb{P}\left(T_0 > b \,\middle|\, Q^M(0) = an\right) \\
\overset{(b)}{\leq}&\mathbb{P}\left(Q_\infty^M \geq l\right) + \mathbb{P}\left(A_b \geq an + l\right)\mathbb{P}\left(T_0 > b \,\middle|\, Q^M(0) = an\right),
\end{aligned} \tag{D.36}$$

where steps (a) and (b) follow from Lemma D.1 and Eq. (D.32), respectively. For the second term in Eq. (D.36), we fix $b = 2a/(1-\rho') > a$. There exist ϕ, θ_1 and $\theta_2 > 0$, so that for all $n \in \mathbb{N}$,

$$
\begin{aligned}
&\mathbb{P}\left(A_b \geq an + l\right) \mathbb{P}\left(T_0 > b \middle| Q^M(0) = an\right) \\
\overset{(a)}{\leq} &\mathbb{P}\left(A_b \geq an + l\right) \exp(-\psi n) \\
= &\mathbb{P}\left(A_b \geq 2b\rho' n + l - \left(2b\rho' - a\right) n\right) \exp(-\psi n) \\
\overset{(b)}{\leq} &\Big(\mathbf{I}\left(l \leq (2b\rho' - a)n\right) + \mathbf{I}\left(l > (2b\rho' - a)n\right) \theta_1 \exp(-\theta_2 l)\Big) \exp(-\psi n) \\
\leq &c_1 \exp\left(-c_2 l\right), \quad \forall l \in \mathbb{Z}_+,
\end{aligned}
\tag{D.37}
$$

where $c_1 = \max\{1, \theta_1\}$ and $c_2 = \min\{\psi/(b\rho' - a), \theta_2\}$, where step (a) follows from Eq. (D.34), and (b) from Lemma 7.8, and the fact that A_b is a Poisson random variable with mean $b\rho$. Combining Eqs. (D.36) and (D.37), and the fact that Q^M_∞ is a geometric random variable and admits an exponential bound on its tail probabilities, we have that there exist $h_1, h_2 > 0$, such that

$$
\begin{aligned}
&\mathbb{P}\left(Q^M(b) \geq l \middle| Q^M(0) = an\right) \\
\leq &\mathbb{P}\left(Q^M_\infty \geq l\right) + c_1 \exp\left(-c_2 l\right) \\
\leq &h_1 \exp(-h_2 l) + c_1 \exp\left(-c_2 l\right) \\
\leq &\max\{h_1, c_1\} \exp\Big(-\min\{h_2, c_2\} l\Big), \quad \forall l \in \mathbb{Z}_+.
\end{aligned}
\tag{D.38}
$$

This proves our claim, by letting $u_1 = \max\{h_1, c_1\}$ and $u_2 = \min\{h_2, c_2\}$. □

D.1.5 Lemma D.1

Proof. It is not difficult to show that, for all integers x and y that satisfy $x \geq y \geq 0$,

$$
\left(Q^M(t) \middle| Q^M(0) = x\right) \succcurlyeq \left(Q^M(t) \middle| Q^M(0) = y\right), \quad \forall t \geq 0. \tag{D.39}
$$

By initializing $Q^M(0)$ with the steady distribution, Q^M_∞, and letting $y = 0$, we have that, for all $t \geq 0$,

$$\begin{aligned}
&\mathbb{P}\left(Q^M_\infty \geq l\right)\\
=&\sum_{x=0}^{\infty} \mathbb{P}\left(Q^M(t) \geq l \,\middle|\, Q^M(0) = x\right) \mathbb{P}\left(Q^M_\infty = x\right)\\
\geq&\sum_{x=0}^{\infty} \mathbb{P}\left(Q^M(t) \geq l \,\middle|\, Q^M(0) = 0\right) \mathbb{P}\left(Q^M_\infty = x\right)\\
=&\mathbb{P}\left(Q^M(t) \geq l \,\middle|\, Q^M(0) = 0\right), \quad \forall l \in \mathbb{Z}_+.
\end{aligned} \tag{D.40}$$

Fix $j \in \mathbb{Z}_+$. Since $Q^M(\cdot)$ is a time-homogeneous (strong) Markov process and T_0 a stopping time, T_0 is independent of the evolution of $\tilde{Q}^M(t) = Q^M(T_0 + t)$, $t \geq 0$. Therefore, we have that

$$\left(Q^M(t) \,\middle|\, Q^M(0) = j, T_0 \leq t\right) \overset{d}{=} \left(Q(T_0') \,\middle|\, Q(0) = 0\right), \tag{D.41}$$

where T_0' be a random variable, independent of the evolution of $Q^M(\cdot)$, with distribution $T_0' \overset{d}{=} \left(t - T_0 \,\middle|\, T_0 \leq t\right)$. Combing Eqs. (D.40) and (D.41), we have that, for all $t \geq 0$,

$$\begin{aligned}
&\mathbb{P}\left(Q^M(t) \geq l \,\middle|\, Q^M(0) = j, T_0 \leq t\right)\\
\overset{(a)}{=}&\mathbb{P}\left(Q^M(T_0') \geq l \,\middle|\, Q^M(0) = 0\right)\\
\overset{(b)}{=}&\int_{s=0}^{\infty} \mathbb{P}\left(Q^M(s) \geq l \,\middle|\, Q^M(0) = 0\right) \mu_{T_0'}(ds)\\
\overset{(c)}{\leq}&\int_{s=0}^{\infty} \mathbb{P}\left(Q^M_\infty \geq l\right) \mu_{T_0'}(ds)\\
=&\mathbb{P}\left(Q^M_\infty \geq l\right), \quad \forall l \in \mathbb{Z}_+,
\end{aligned} \tag{D.42}$$

where $\mu_{T_0'}$ is the probability measure induced by T_0'. Step (a) follows from Eq. (D.41), (b) from the independence between T_0' and the evolution of $Q^M(\cdot)$, and (c) from Eq. (D.40). This proves our claim. □