Delay, stability, and resource tradeoffs in large distributed service systems

by

Martín Zubeldía Suárez

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

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

Author .....
Department of Electrical Engineering and Computer Science
August 19, 2019

Certified by .....
David Gamarnik
Nanyang Technological University Professor of Operations Research
Thesis Supervisor

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

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Delay, stability, and resource tradeoffs in large distributed service systems

by

Martín Zubeldía Suárez

Submitted to the Department of Electrical Engineering and Computer Science
on August 19, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Abstract

This thesis addresses fundamental tradeoffs in the design of dispatching policies in large-scale distributed service systems, motivated by examples such as cloud computing facilities and multi-core processors. A canonical framework for modeling such systems is provided by a parallel queueing model with n servers, where service requests arrive stochastically over time as a single stream of jobs of rate proportional to n, and where a central controller is responsible for all decisions.

The central controller makes decisions based on limited information about the state of the queues, which is conveyed through messages from servers to the dispatcher, and stored in a limited local memory. Our objective is to understand the best possible performance of such systems (in terms of stability region and delay) and to propose optimal policies, with emphasis on the asymptotic regime when both the number of servers and the arrival rate are large.

We study the tradeoffs between the resources available to the controller (memory size and message rate) and the achievable queueing delay performance and stability region of resource constrained dispatching policies. Our main findings are:

1. *Queueing delay vs. resources tradeoff.* We propose a family of dispatching policies, indexed by the size of their memories and by the average message rate, and show that the expected queueing delay vanishes as $n \to \infty$ when either (i) the number of memory bits is of the order of $\log(n)$ and the message rate grows superlinearly with n, or (ii) the number of memory bits grows superlogarithmically with n and the message rate is at least as large as the arrival rate (Chapter 3). Moreover, we develop a novel approach to show that, within a certain broad class of "symmetric" policies, every dispatching policy with a message rate of the order of n, and with a memory of the order of $\log(n)$ bits, results in an expected queueing delay which is bounded away from zero, uniformly as $n \to \infty$ (Chapter 4).

2. *Stability region vs. resources tradeoff.* We propose a dispatching policy that requires a memory size (in bits) of the order of $\log(n)$ and an arbitrarily small

(but positive) message rate, and show that it is stable for all possible server rates for which the entire system is underloaded. Moreover, we show that within a certain broad class of "weakly symmetric" policies, every dispatching policy with a message rate of the order of $o(n^2)$, and with a memory size that grows sublogarithmically with n, results in a reduced stability region (Chapter 5).

Thesis Supervisor: David Gamarnik
Title: Nanyang Technological University Professor of Operations Research

Thesis Supervisor: John N. Tsitsiklis
Title: Clarence J. Lebel Professor of Electrical Engineering

Acknowledgments

First and foremost, I would like to thank my advisors, David Gamarnik and John Tsitsiklis, for all their guidance, support, and patience over the past five years. They gave me the freedom that I needed to learn how to pick my own research problems, and the guidance to learn how to approach them with clarity and rigor. I consider myself to be very lucky to have had them as my advisors, and I will always be grateful to them.

Life at LIDS would not have been the same without the great officemates that I had over the years, Zied Ben Chaouch, Zhi Xu, Dogyoon Song, Sarah Cen, and Anish Agarwal, which made my journey all the more enjoyable. I also want to thank the administrative staff at LIDS for their help, especially Lynne Dell, Brian Jones, and Francisco James.

Last but not least, I would like to thank my parents for all their support and encouragement, which allowed me to pursue my dreams.

My doctoral studies at MIT were supported by the MIT Jacobs Presidential Fellowship, NSF Grant CMMI-1234062 and ONR Grant N0014-17-1-2790.

Contents

1 Introduction **15**

1.1 Context 15

1.2 Related literature 17

1.3 Summary of main contributions 18

1.3.1 Delay, memory, and messaging tradeoffs 18

1.3.2 Stability vs resources tradeoff in heterogeneous systems 19

1.4 Organization of the thesis 20

2 Notation and models **21**

2.1 Notation 21

2.2 General queueing model 24

3 Efficient dispatching policies **27**

3.1 Model and main results 28

3.1.1 Modeling assumptions and performance metric 29

3.1.2 Policy description and high-level overview of the results 29

3.1.3 Stochastic and fluid descriptions of the system 32

3.1.4 Technical results 35

3.1.5 Asymptotic queueing delay and phase transitions 38

3.2 Proof of part of Theorem 3.1.1 44

3.2.1 Uniqueness of solutions 45

3.2.2 Existence, uniqueness, and characterization of an equilibrium . 51

3.2.3 Asymptotic stability of the equilibrium 52

3.3 Proof of Theorem 3.1.2 and of the rest of Theorem 3.1.1 60
3.3.1 Probability space and coupling 60
3.3.2 Tightness of sample paths 63
3.3.3 Derivatives of the fluid limits 67
3.4 Proofs of Proposition 3.1.3 and Theorem 3.1.4 77
3.4.1 Stochastic stability of the n-th system 77
3.4.2 Convergence of the invariant distributions 80
3.5 Conclusions and future work 84

4 Universal delay lower bound for dispatching policies 87
4.1 Model and main results 88
4.1.1 Modeling assumptions and performance metric 88
4.1.2 Unified framework for dispatching policies 89
4.1.3 Delay lower bound for resource constrained policies 97
4.1.4 Queueing delay vs resources tradeoff 99
4.2 Literature review 99
4.2.1 Memory, messages, and queueing delay 105
4.3 Proof of Theorem 4.1.1 105
4.3.1 Local limitations of finite memory 106
4.3.2 A sequence of "bad" events 111
4.3.3 Lower bound on the probability of "bad" events 113
4.3.4 Upper bound on the number of useful distinguished servers . . 121
4.3.5 Completing the proof 125
4.4 Additional proofs 127
4.4.1 A combinatorial inequality 127
4.4.2 Proof of Lemma 4.3.6 127
4.4.3 Proof of Lemma 4.3.9 131
4.5 Conclusions and future work 135

5 Stability vs resources tradeoff in heterogeneous systems 137
5.1 Model and main results 138

5.1.1 Modeling assumptions and performance metric 138
5.1.2 Universally stable policy 139
5.1.3 Unified framework for dispatching policies 141
5.1.4 Instability of resource constrained policies 147
5.1.5 Stability vs resources tradeoff 148
5.2 Proof of Theorem 5.1.1 148
5.3 Proof of Theorem 5.1.2 152
5.3.1 Local limitations of finite memory 152
5.3.2 High arrival rate to slow servers 153
5.4 Conclusions and future work 156

6 Concluding remarks **159**

List of Figures

1-1 Basic distributed service system. 17

2-1 General distributed service system. 24

3-1 Resource Constrained Pull-Based policy. Jobs are sent to queues associated with idle servers, based on tokens in the virtual queue. If no tokens are present, a queue is chosen at random. 30

3-2 Resource requirements of the three regimes, and the resulting asymptotic queueing delays. 31

3-3 Relationship between the stochastic system and the fluid model. . . . 38

3-4 Average queueing delay of the power-of-2-choices policy (red circles) vs. our policy (blue squares) vs. PULL (green asterisks). 44

3-5 An example of a non-differentiable solution for the High Message regime, with $\lambda = 0.9$, $s_1(0) = s_2(0) = s_3(0) = 0.7$, and $s_i(0) = 0$ for all $i \geq 4$. The solution is non-differentiable at the points indicated by the circles. 51

4-1 Resource requirements for vanishing queueing delays. 100

5-1 Resource requirements for stable policies. 148

List of Tables

3.1 The three regimes of our policy, and the resulting asymptotic queueing delays. 31

Chapter 1

Introduction

1.1 Context

Distributed service systems are ubiquitous, from passport control at the airport and checkout lines at the supermarket, to multi-core processing units and server farms for cloud computing. Common to these is a large number of processing units, and a stream of incoming service requests. Naturally, the performance and the stability of such systems depends critically on how service requests are dispatched to the different processing units.

The importance of distributed service systems, compounded with a relatively poor understanding of the fundamental limitations and tradeoffs of the dispatching policies used to operate them, has been the main motivation of this thesis. That being said, the notion of distributed service systems is rather general, and there are many different types of systems that fall under the same term. In this thesis, our focus will be on distributed service systems that have the following features:

1. **Non-trivial dynamics.** We will study systems that involve non-trivial dynamics (as opposed to static models), where decisions have to be made repeatedly over time. As a result, our performance metrics will also involve quantities that depend on this dynamic nature, such as average delays and stability regions.

2. **Large-scale.** We will focus on the regime where the size of the system (e.g.,

number of processing units and service requests) grows to infinity. Our main motivation for studying this regime stems from the fact that most applications of interest have a very large number of processing units. Moreover, while most dynamical service systems are fairly intractable, asymptotic analyses are often possible, and can provide significant architectural insights.

3. **Centralized control.** We will study systems whose operation is dictated by a centralized decision maker, who follows policies that are prescribed beforehand. This is in contrast to game-theoretic models, where the dynamics of the system can be the result of strategic interactions among different parties.

In order to take advantage of the multiple processing units available, the decision maker can benefit from information about the current state of the system (e.g., which processing units are idle). For such information to be available when a decision has to be made, it is necessary for the decision maker to periodically obtain information about the current state of the system and/or have sufficient memory that allows it to extrapolate from the available information. This requires resources in the form of bandwidth (to obtain information) and/or physical memory (to store information).

A canonical framework that has emerged for modeling the systems mentioned above is provided by a parallel queueing system, consisting of a large number of servers that operate in parallel, where each one has a queue associated to it. In principle, these servers can have different processing rates. In addition, service requests arrive to the system in the form of a single stream of discrete jobs. Upon arrival, the individual jobs are dispatched by a central decision maker to a suitable queue. This stylized model is depicted in Figure 1-1.

In this setting, the design of good dispatching policies remains an important challenge. In this thesis, we focus on performance-related tradeoffs. More specifically, we are interested in advancing existing methods of performance analysis and generating new insights into existing policies considered in the literature, as well as designing new policies to achieve good performance. For both of these objectives, we will see that system performance depends crucially on the amount of information available to

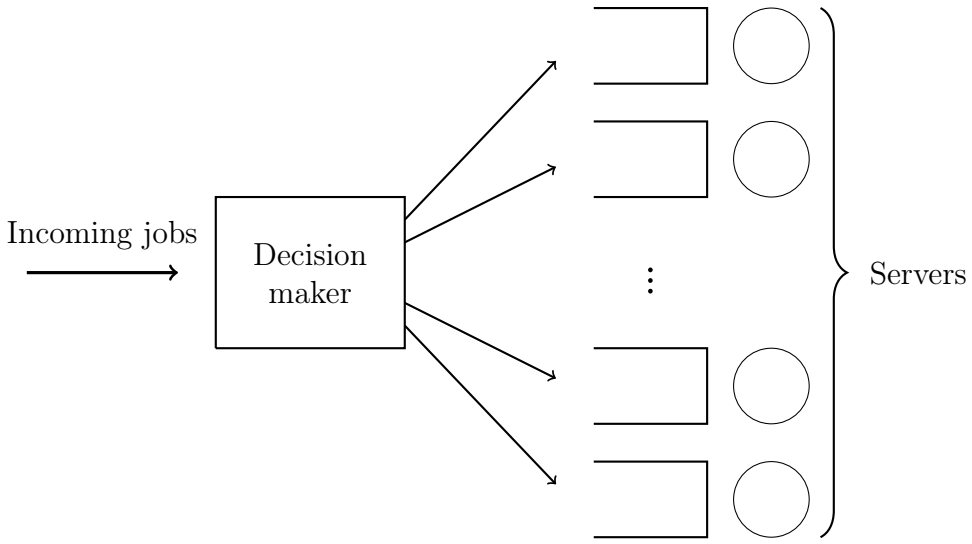

Figure 1-1: Basic distributed service system.

the decision maker, and on the system load.

1.2 Related literature

In this section, we review relevant research on the design and analysis of dispatching policies for distributed service systems. There is a variety of ways in which such system can be operated, which correspond to different decision making architectures and policies. At one extreme, if no information about the state of the system is available, it is known that the best policy is to dispatch incoming jobs according to a Round-Robin policy [39]. This policy has no informational requirements but incurs a substantial delay because it does not take advantage of resource pooling. At the other extreme, if complete information about the state of the system is available, it is known that the best policy is to send incoming jobs to one of the queues with the smallest workload [4]. Such policy has very good performance (small queueing delay), but relies on substantial information exchange.

Many intermediate policies have been explored in the literature, and they achieve different performance levels while using varying amounts of resources, including local memory and communication overhead. For example, the power-of-d-choices [34, 44] and its variations [33, 36, 47, 35, 25] have been extensively studied, including the case

of non-exponential service time distributions [13, 14, 2]. More recently, pull-based policies like Join-Idle-Queue [8, 31] have been drawing attention, including extensions for heterogeneous servers [40, 5], multiple dispatchers [32, 43, 41], heavy-traffic [22], and general service time distributions [17]. A more extensive review of this literature is given in Chapter 4.

On the other hand, tradeoffs similar to the ones we study in this thesis have been analyzed in the context of the balls into bins model [6], in which n balls are to be placed sequentially into n bins. In particular, the tradeoff between the number of messages exchanged and the maximum number of balls in any one bin was recently characterized in [29, 1]. Furthermore, the tradeoff between memory size and maximum number of balls in any one bin was studied in [3, 9].

Note that the balls into bins model and the dynamical model that we consider are similar, in the sense that sequential decisions about the destination of balls (or tasks) have to be made and that there are many policies that are used and perform well in both settings (e.g., the policy where balls/tasks join one of the shortest bins/queues). However, the fact that the balls (unlike the jobs) never leave the system makes these problems substantially different. This difference also leads to most of the literature studying the balls into bins model using different performance metrics, such as the maximum load among the bins after all balls were placed. Because of these differences, results from one setting can rarely be translated into the other.

1.3 Summary of main contributions

We now summarize the main contributions of this thesis.

1.3.1 Delay, memory, and messaging tradeoffs

Our first contribution is the study of the effect of different resource levels at the dispatcher (local memory and communication overhead) on the expected delay of a typical job. This is studied in the setting of resource constrained dispatching systems,

with homogeneous servers, and jobs that consist of a single task that cannot be replicated.

Efficient dispatching policies. In Chapter 3 we propose a family of dispatching policies, parameterized by the size of the memory used by the dispatcher and by the average rate of messages exchanged between the dispatcher and the servers. For this family of policies, we show that the expected queueing delay of a typical job vanishes (as the number of servers and the arrival rate go to infinity) as long as the resources are above a certain level.

In particular, we show that if either (i) the average message rate is superlinear in the arrival rate and the memory size (in bits) is of the order of the logarithm of the number of servers, or (ii) the average message rate greater than or equal to the arrival rate and the memory size (in bits) is superlogarithmic in the number of servers, then the expected queueing delay of a typical job vanishes as the system size increases.

Universal delay lower bound for dispatching policies. In Chapter 4, we consider a broad family of decision making architectures and policies, which includes the ones introduced in Chapter 3 along with most of those considered in the earlier literature, and work towards characterizing the minimum amount of resources required in order to obtain vanishing queueing delays, as the system size increases.

In particular, we show that if the average message rate is at most of the order of the arrival rate, and the memory size (in bits) is at most of the order of the logarithm of the number of servers, then *every* decision making architecture and policy, within the class of dispatching policies considered, results in a queueing delay that does **not** vanish as the system size increases. This complements the results obtained in Chapter 3.

1.3.2 Stability vs resources tradeoff in heterogeneous systems

Our second contribution is the study of the effect of different resource levels (local memory and communication overhead) on the stability region of dispatching policies.

This is again studied in the setting of resource constrained dispatching systems, with jobs that consist of a single task that cannot be replicated, but now we allow the servers to have different processing rates, which are not known at the dispatcher.

Universally stable dispatching policy. In Chapter 5 we propose a simple dispatching policy, which requires a memory of size (in bits) logarithmic in the number of servers and a positive (but arbitrarily small) message rate, and we show that it has the largest possible stability region.

Instability of resource constrained policies. Also in Chapter 5, we consider a slightly broader family of decision making architectures and policies than the one defined in Chapter 4, and work towards characterizing the minimum amount of resources required in order to obtain a policy with the largest possible stability region.

In particular, we show that if the average message rate is smaller than of the order of the square of the arrival rate, and the memory size (in bits) is sublogarithmic in the number of servers, then *every* decision making architecture and policy, within the broad class of dispatching policies considered, has a reduced stability region.

1.4 Organization of the thesis

The rest of the thesis is organized as follows. We begin by describing some of our main modeling assumptions and notation in Chapter 2. In Chapter 3 we introduce and analyze a family of resource efficient dispatching policies. In Chapter 4 we present an impossibility result for resource constrained dispatching policies. In Chapter 5 we study the stability region of heterogeneous service systems with limited resources. Finally, we conclude the thesis in Chapter 6, where we also highlight several potential avenues for future research.

Chapter 2

Notation and models

In this chapter we introduce the notation used, and the queueing model studied in the thesis. We refrain from getting into details of mathematical formalism, which will be presented in subsequent chapters. Instead, we focus on highlighting the main features.

2.1 Notation

In this section we collect, for ease of reference, the notation that will be used throughout the thesis. First, we define notation for the asymptotic behavior of positive functions, which is as follows:

$$
\begin{aligned}
f(n) \in o\big(g(n)\big) &\;\Leftrightarrow\; \limsup_{n\to\infty} \frac{f(n)}{g(n)} = 0,\\
f(n) \in O\big(g(n)\big) &\;\Leftrightarrow\; \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty,\\
f(n) \in \Theta\big(g(n)\big) &\;\Leftrightarrow\; 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \le \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty,\\
f(n) \in \Omega\big(g(n)\big) &\;\Leftrightarrow\; \liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0,\\
f(n) \in \omega\big(g(n)\big) &\;\Leftrightarrow\; \liminf_{n\to\infty} \frac{f(n)}{g(n)} = \infty.
\end{aligned}
$$

We let $[\,\cdot\,]^+ \triangleq \max\{\,\cdot\,, 0\}$. We denote by $\mathbb{Z}_+$ and $\mathbb{R}_+$ the sets of non-negative integers and real numbers, respectively. The indicator function is denoted by $\mathbb{1}$, so that $\mathbb{1}_A(x)$ is 1 if $x \in A$, and is 0 otherwise. The Dirac measure δ concentrated at a point x is defined by $\delta_x(A) \triangleq \mathbb{1}_A(x)$. We also define the following sets:

$$\mathcal{S} \triangleq \left\{ s \in [0,1]^{\mathbb{Z}_+} : s_0 = 1;\ s_i \geq s_{i+1},\ \forall\, i \geq 0 \right\},$$

$$\mathcal{S}^1 \triangleq \left\{ s \in \mathcal{S} : \sum_{i=0}^{\infty} s_i < \infty \right\}, \tag{2.1}$$

$$\mathcal{I}_n \triangleq \left\{ x \in [0,1]^{\mathbb{Z}_+} : x_i = \frac{k_i}{n},\ \text{for some } k_i \in \mathbb{Z}_+,\ \forall\, i \geq 0 \right\}.$$

We define the weighted ℓ_2 norm $||\cdot||_w$ on $\mathbb{R}^{\mathbb{Z}_+}$ by

$$||x - y||_w^2 \triangleq \sum_{i=0}^{\infty} \frac{|x_i - y_i|^2}{2^i}.$$

Note that this norm comes from an inner product, so $(\ell_w^2, \|\cdot\|_w)$ is actually a Hilbert space, where

$$\ell_w^2 \triangleq \left\{ s \in \mathbb{R}^{\mathbb{Z}_+} : \|s\|_w < \infty \right\}.$$

We also define partial orders on $\mathcal{S}$ as follows:

$$\begin{aligned} x \geq y \quad &\Leftrightarrow \quad x_i \geq y_i, \quad \forall\, i \geq 1, \\ x > y \quad &\Leftrightarrow \quad x_i > y_i, \quad \forall\, i \geq 1. \end{aligned}$$

We will sometimes work with the Skorokhod spaces of functions

$$D[0,T] \triangleq \left\{ f : [0,T] \to \mathbb{R} : f \text{ is right-continuous with left limits} \right\},$$

endowed with the uniform metric

$$d(x,y) \triangleq \sup_{t \in [0,T]} |x(t) - y(t)|,$$

and

$$D^{\infty}[0,T] \triangleq \left\{ f : [0,T] \to \mathbb{R}^{\mathbb{Z}_+} : f \text{ is right-continuous with left limits} \right\},$$

with the metric

$$d^{\mathbb{Z}_+}(x,y) \triangleq \sup_{t \in [0,T]} ||x(t) - y(t)||_w.$$

Given a set A, its power set, the set of all subsets of A, is denoted by $\mathcal{P}(A)$. Random variables will always be denoted by upper case symbols. Non-random quantities will generally — but not always — be denoted by lower case symbols; exceptions will be pointed out as necessary. We will use boldface fonts to denote vectors. If $\mathbf{v}$ is a vector, we denote its i-th component by $\mathbf{v}_i$. We will denote the (unordered) set of elements of a vector by using the superscript "set"; for example, if $\mathbf{v} = (2,1,3,1)$, then $\mathbf{v}^{set} = \{1,2,3\}$. Furthermore, we will use $|\mathbf{v}|$ to denote the dimension of a vector $\mathbf{v}$. If $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$ is a vector, and $\mathbf{u}$ is a vector with entries in $\{1, \ldots, m\}$, then $\mathbf{v}_{\mathbf{u}}$ is a $|\mathbf{u}|$-dimensional vector whose i-th component is $\mathbf{v}_{\mathbf{u}_i}$; for example, if $\mathbf{u} = (3,1)$, then $\mathbf{v}_{\mathbf{u}} = (\mathbf{v}_3, \mathbf{v}_1)$.

For any positive integer n, we define the sets $\mathcal{N}_n \triangleq \{1, \ldots, n\}$, and

$$\mathcal{R}_n \triangleq \left\{ \mathbf{s} \in \bigcup_{i=0}^{n} (\mathcal{N}_n)^i : \text{there are no repeated coordinates in } \mathbf{s} \right\}, \tag{2.2}$$

where $(\mathcal{N}_n)^0 = \{\emptyset\}$. We say that a permutation $\sigma : \mathcal{N}_n \to \mathcal{N}_n$ **fixes a set** $R \subset \mathcal{N}_n$ if $\sigma(i) = i$, for all $i \in R$. Furthermore, we say that a permutation σ **preserves the ordering** of a subset $A \subset \mathcal{N}_n$ if $\sigma(i) < \sigma(j)$ whenever $i, j \in A$ and $i < j$. If $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$ is a vector in $(\mathcal{N}_n)^m$ and σ is a permutation of $\mathcal{N}_n$, we denote by $\sigma(\mathbf{v})$ the vector $\big(\sigma(\mathbf{v}_1), \ldots, \sigma(\mathbf{v}_m)\big)$. Finally, for any function X of time, and any $t \in \mathbb{R}$, we let $X(t^-) \triangleq \lim_{\tau \to t^-} X(\tau)$, as long as the limit exists.

2.2 General queueing model

In this section, we introduce the basic queueing model that will serve as the basis of the more specific queueing models that are used in subsequent chapters.

We consider a queueing model consisting of n parallel servers, where each server is associated with an infinite capacity First-In-First-Out (FIFO) queue. Jobs arrive to the system as a single renewal process of rate λn, where λ is a positive constant. The sizes of the incoming jobs are independent and identically distributed, independent from the arrival process, and have an arbitrary distribution with unit mean.

Connecting the incoming service requests to the servers there is a central controller (dispatcher), responsible for routing the incoming jobs to suitable queues (see Figure 2-1). This dispatcher makes decisions based on limited information about the state of the queues, which is conveyed through messages from servers to the dispatcher, and stored in a limited local memory.

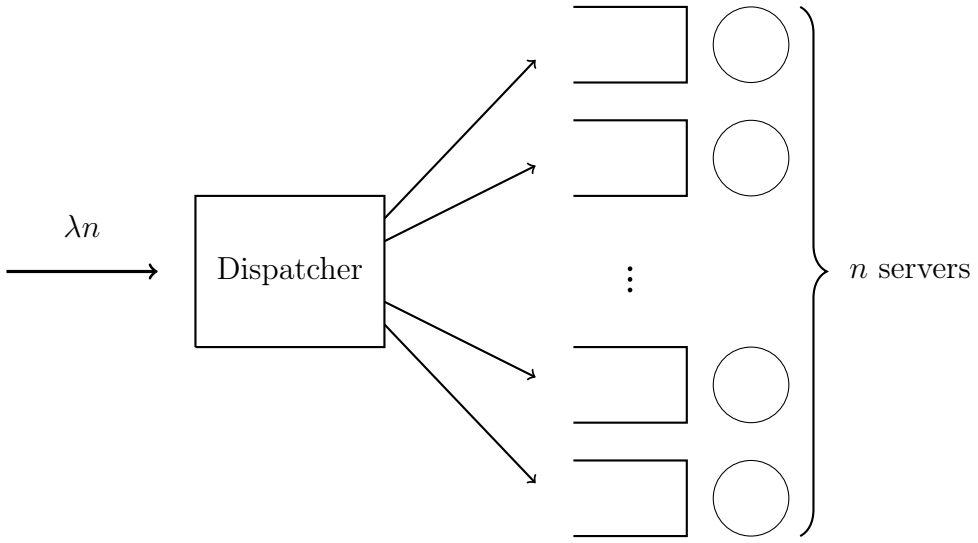

Figure 2-1: General distributed service system.

In particular, we consider a dispatcher with the following limitations:

1. **Finite memory:** The dispatcher has a local finite memory to store any type of information (e.g., information about the current state of the queues).

2. **Bounded message rate:** There is a limit on the average number of messages

exchanged between the dispatcher and the servers.

Note that these two limitations restrict the amount of information available to the dispatcher for decision-making.

Chapter 3

Efficient dispatching policies

In this chapter we focus on homogeneous distributed service systems consisting of a large number of servers with equal service rates, and study the tradeoffs between the expected queueing delay of a typical job, and the resources (local memory and message rate) available to the dispatcher. This is achieved by introducing a family of dispatching policies parameterized by the amount of resources involved. We carry out a thorough analysis in different regimes and show that the queueing delay vanishes as the number of servers increases only if the resources are above a certain level.

More concretely, our development relies on a fluid limit approach, where we take the limit when the number of servers (n) goes to infinity. As is common with fluid-based analyses, we obtain two types of results: (i) qualitative results obtained through a deterministic analysis of a fluid model, and (ii) technical results on the convergence of the actual stochastic system to its fluid counterpart.

On the qualitative end, we establish the following:

a) If the message rate is superlinear in n and the number of memory bits is at least logarithmic in n, then the asymptotic delay is zero.

b) If the message rate is larger than or equal to the arrival rate and the number of memory bits is superlogarithmic in n, then the asymptotic delay is zero.

c) If the message rate is αn and the number of memory bits is $\lceil c \log_2(n) \rceil$, we derive a closed form expression for the (now positive) asymptotic delay in terms of the

arrival rate, α, and c, and which exhibits interesting phase transitions.

On the technical end, and for each one of three regimes corresponding to cases (a), (b), and (c) above, we show the following:

a) The queue length process converges (as $n \to \infty$, and over any finite time interval) almost surely to the unique solution of a certain fluid model.

b) For any initial conditions that correspond to starting with a finite average number of jobs per queue, the fluid solution converges (as time tends to ∞) to a unique invariant state.

c) The steady-state distribution of the finite system converges (as $n \to \infty$) to the invariant state of the fluid model.

The rest of the chapter is organized as follows. In Section 3.1 we present the model and the main results, and also compare a particular regime of our policy to the so-called "power-of-d-choices" policy. In sections 3.2–3.4 we provide the proofs of the main results. Finally, in Section 3.5 we present our conclusions and suggestions for future work.

The results on this chapter first appeared in [19] and [20].

3.1 Model and main results

In this section we present the specific modeling assumptions, the performance metrics of interest, and our main results. In Subsection 3.1.1 we describe the model and our assumptions. In Subsection 3.1.2 we introduce three different regimes of a certain pull-based dispatching policy. In subsections 3.1.3 and 3.1.4 we introduce a fluid model and state the validity of fluid approximations for the transient and the steady-state regimes, respectively. In Subsection 3.1.5, we discuss the asymptotic delay, and show a phase transition in its behavior.

3.1.1 Modeling assumptions and performance metric

We now introduce a refinement of the modeling assumptions for the basic model presented in Section 2.2. In particular, throughout this chapter we assume that the n servers are homogeneous, i.e., they all have the same processing rate. assumed to be equal to 1. Furthermore, jobs arrive to the system as a single Poisson process of rate λn (for some fixed $\lambda \in (0, 1)$), and their sizes are i.i.d., independent from the arrival process, and exponentially distributed with unit mean. Finally, we assume that the central dispatcher has to route each incoming job to a queue immediately upon arrival (i.e., jobs cannot be queued at the dispatcher).

Regarding the performance metric, we will focus on the steady-state expectation of the time between the arrival of a typical job and the time at which it starts receiving service, to be referred to as the **expected queueing delay of a typical job**.

3.1.2 Policy description and high-level overview of the results

In this subsection we introduce our policy and state in a succinct form our results for three of its regimes.

Policy description

For any fixed value of n, the policy that we study operates as follows.

a) **Memory:** The dispatcher maintains a virtual queue comprised of up to c_n server identity numbers (IDs), also referred to as **tokens**, so that the dispatcher's memory size is $\lceil c_n \log_2(n) \rceil$ bits. Since there are only n distinct servers, we will assume throughout the rest of the chapter that $c_n \leq n$.

b) **Spontaneous messages from idle servers:** While a server is idle, it sends messages to the dispatcher as a Poisson process of rate β_n, to inform or remind the dispatcher of its idleness. We assume that β_n is a nondecreasing function of n. Whenever the dispatcher receives a message, it adds the ID of the server

that sent the message to the virtual queue of tokens, unless this ID is already stored or the virtual queue is full, in which cases the new message is discarded.

c) **Dispatching rule:** Whenever a new job arrives, if there is at least one server ID in the virtual queue, the job is sent to the queue of a server whose ID is chosen uniformly at random from the virtual queue, and the corresponding token is deleted. If there are no tokens present, the job is sent to a queue chosen uniformly at random.

Note that under the above described policy, which is also depicted in Figure 3-1, no messages are ever sent from the dispatcher to the servers. Accordingly, following the terminology of [8], we will refer to it as the Resource Constrained Pull-Based (**RCPB**) policy or **Pull-Based** policy for short.

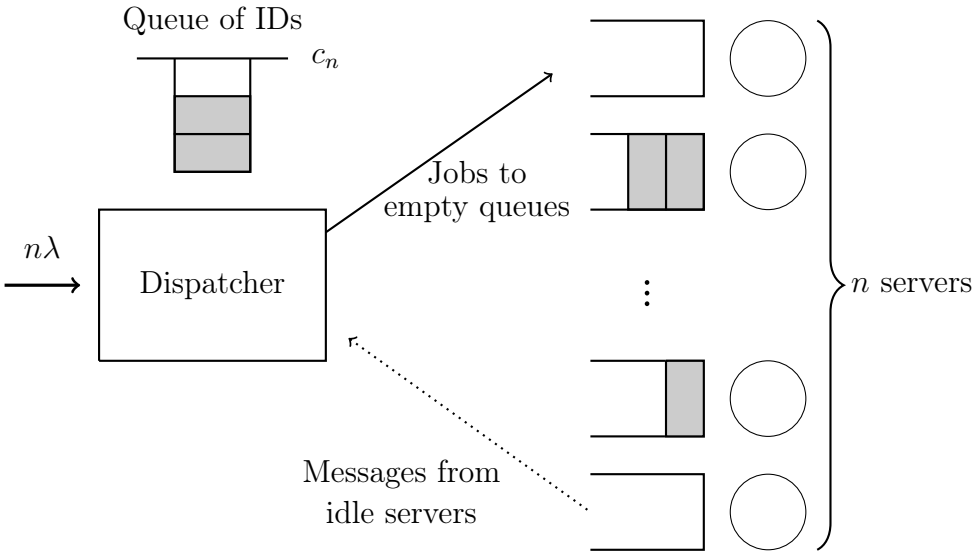

Figure 3-1: Resource Constrained Pull-Based policy. Jobs are sent to queues associated with idle servers, based on tokens in the virtual queue. If no tokens are present, a queue is chosen at random.

High-level summary of the results

We summarize our results for the RCPB policy, for three different regimes, in Table 3.1, where we also introduce some mnemonic terms that we will use to refer to these

regimes. Formal statements of these results are given later in this section. Furthermore, we provide a pictorial representation of the total resource requirements and the corresponding asymptotic queueing delays in Figure 3-2.

Regime	Memory	Idle message rate	Delay
High Memory	$c_n \in \omega(1)$ and $c_n \in o(n)$	$\beta_n = \beta \geq \frac{\lambda}{1-\lambda}$ $\beta_n = \beta < \frac{\lambda}{1-\lambda}$	0 > 0
High Message	$c_n = c \geq 1$	$\beta_n \in \omega(1)$	0
Constrained	$c_n = c \geq 1$	$\beta_n = \beta > 0$	> 0

Table 3.1: The three regimes of our policy, and the resulting asymptotic queueing delays.

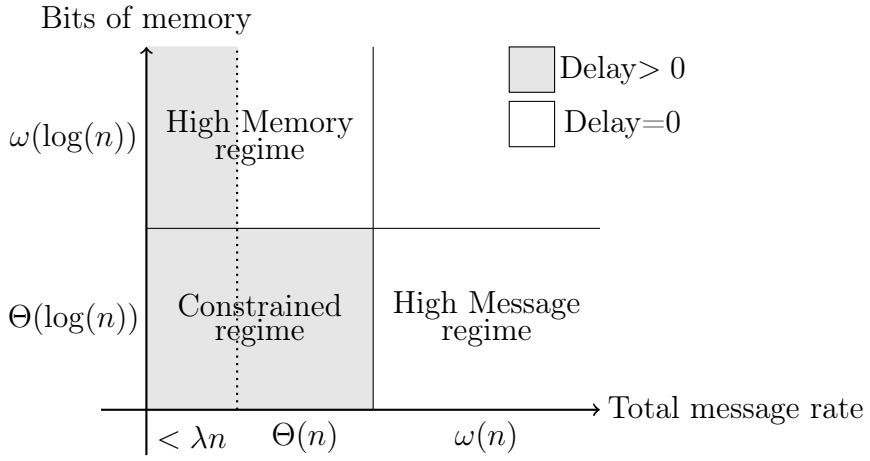

Figure 3-2: Resource requirements of the three regimes, and the resulting asymptotic queueing delays.

The more interesting subcase of the High Memory regime is when $\beta \geq \lambda/(1-\lambda)$, which results in zero asymptotic queueing delay with superlogarithmic memory and linear overall message rate. Note that if we set $\beta = \lambda/(1-\lambda)$, and use the fact that servers are idle a fraction $1-\lambda$ of the time, the resulting time-average message rate becomes exactly λn, i.e., one message per arrival.

3.1.3 Stochastic and fluid descriptions of the system

In this subsection, we define a stochastic process that corresponds to our model under the RCPB policy, as well as an associated fluid model.

Stochastic system representation

Let $\mathbf{Q}_i^n(t)$ be the number of jobs in queue i (including the job currently being served, if any), at time t, in a n-server system. We can model the system as a continuous-time Markov chain whose state at time t is the queue length vector, $\mathbf{Q}^n(t) = \left(\mathbf{Q}_i^n(t)\right)_{i=1}^n \in \mathbb{Z}_+^n$, together with the number of tokens, denoted by $M^n(t) \in \{0, 1, \ldots, c_n\}$. However, as the system is symmetric with respect to the queues, we will use instead the more convenient representation $S^n(t) = \left(S_i^n(t)\right)_{i=0}^\infty$, where

$$S_i^n(t) \triangleq \frac{1}{n} \sum_{j=1}^n \mathbb{1}_{[i,\infty)} \left(\mathbf{Q}_j^n(t)\right), \quad i \in \mathbb{Z}_+,$$

is the fraction of queues with at least i jobs at time t. Once more, the pair $\left(S^n(\cdot), M^n(\cdot)\right)$ is a continuous-time Markov process, with a countable state space.

Finally, another possible state representation involves $V^n(t) = \left(V_i^n(t)\right)_{i=1}^\infty$, where

$$V_i^n(t) \triangleq \sum_{j=i}^\infty S_j^n(t)$$

can be interpreted as the average amount by which a queue length exceeds $i-1$ at time t. In particular, $V_1^n(t)$ is the total number of jobs at time t divided by n, and is finite, with probability 1.

Fluid model

We now introduce the fluid model of $S^n(\cdot)$, associated with our policy. Recall the definition of the set $\mathcal{S}^1$ in Equation (2.1).

Definition 3.1.1 (Fluid model)**.** Given an initial condition $s^0 \in \mathcal{S}^1$, a continuous function $s : [0, \infty) \to \mathcal{S}^1$ is said to be a solution to the fluid model (or fluid solution)

if:

1. $s(0) = s^0$.

2. For all $t \geq 0$, $s_0(t) = 1$.

3. For all $t \geq 0$ outside of a set of Lebesgue measure zero, and for every $i \geq 1$, $s_i(t)$ is differentiable and satisfies

$$\frac{ds_1}{dt}(t) = \lambda\big(1 - P_0(s(t))\big) + \lambda(1 - s_1(t))P_0(s(t)) - \big(s_1(t) - s_2(t)\big), \tag{3.1}$$

$$\frac{ds_i}{dt}(t) = \lambda\big(s_{i-1}(t) - s_i(t)\big)P_0(s(t)) - \big(s_i(t) - s_{i+1}(t)\big) \quad \forall\, i \geq 2, \tag{3.2}$$

   where $P_0(s)$ is given, for the three regimes considered, by:

   (i) High Memory: $P_0(s) = \left[1 - \frac{\beta(1-s_1)}{\lambda}\right]^+$;

   (ii) High Message: $P_0(s) = \left[1 - \frac{1-s_2}{\lambda}\right]^+ \mathbb{1}_{\{1\}}(s_1)$;

   (iii) Constrained: $P_0(s) = \left[\sum_{k=0}^{c} \left(\frac{\beta(1-s_1)}{\lambda}\right)^k\right]^{-1}$.

   We use the convention $0^0 = 1$, so that the case $s_1 = 1$ yields $P_0(s) = 1$.

A solution to the fluid model, $s(\cdot)$, can be thought of as a deterministic approximation to the sample paths of the stochastic process $S^n(\cdot)$, for n large enough. Note that the fluid model does not include a variable associated with the number of tokens. This is because, as we will see, the virtual queue process $M^n(\cdot)$ evolves on a faster time scale than the processes of the queue lengths and does not have a deterministic limit. We thus have a process with two different time scales: on the one hand, the virtual queue evolves on a fast time scale (at least n times faster) and from its perspective the queue process $S^n(\cdot)$ appears static; on the other hand, the queue process $S^n(\cdot)$ evolves on a slower time scale and from its perspective, the virtual queue appears to be at stochastic equilibrium. This latter property is manifested in the drift of the fluid model: $P_0(s(\cdot))$ can be interpreted as the probability that the virtual queue is empty when the rest of the system is fixed at the state $s(\cdot)$. Moreover, the drift of $s_1(\cdot)$ is qualitatively different from the drift of the other components $s_i(\cdot)$, for $i \geq 2$, because our policy treats empty queues differently.

We now provide some intuition for each of the drift terms in Equations (3.1) and (3.2).

(i) $\lambda\big(1 - P_0(s(t))\big)$: This term corresponds to arrivals to an empty queue while there are tokens in the virtual queue, taking into account that the virtual queue is nonempty with probability $1 - P_0(s(t))$, in the limit.

(ii) $\lambda\big(s_{i-1}(t) - s_i(t)\big)P_0(s(t))$: This term corresponds to arrivals to a queue with exactly $i-1$ jobs while there are no tokens in the virtual queue. This occurs when the virtual queue is empty and a queue with $i-1$ jobs is drawn, which happens with probability $P_0(s(t))\big(s_{i-1}(t) - s_i(t)\big)$.

(iii) $-\big(s_i(t) - s_{i+1}(t)\big)$: This term corresponds to departures from queues with exactly i jobs, which after dividing by n, occur at a rate equal to the fraction $s_i(t) - s_{i+1}(t)$ of servers with exactly i jobs.

(iv) Finally, the expressions for $P_0(s)$ are obtained through an explicit calculation of the steady-state distribution of $M^n(t)$ when $S^n(t)$ is fixed at the value s, while also letting $n \to \infty$.

Let us give an informal derivation of the different expressions for $P_0(s)$. Recall that $P_0(s)$ can be interpreted as the probability that the virtual queue is empty when the rest of the system is fixed at the state s. Under this interpretation, for any fixed state s, and for any fixed n, the virtual queue would behave like an M/M/1 queue with capacity c_n, arrival rate $\beta_n n(1 - s_1)$, and departure rate λn. In this M/M/1 queue, the steady-state probability of being empty is

$$P_0^{(n)}(s) = \left[\sum_{k=0}^{c_n} \left(\frac{\beta_n(1-s_1)}{\lambda}\right)^k\right]^{-1}.$$

By taking the limit as $n \to \infty$, we obtain the correct expressions for $P_0(s)$, except in the case of the High Message regime with $s_1 = 1$. In that particular case, this simple interpretation does not work. However, we can go one step further and note that when all servers are busy (i.e., when $s_1 = 1$), servers become idle at rate $1 - s_2$,

which is the proportion of servers with exactly one job left in their queues. Since the high message rate assures that messages are sent almost immediately after the server becomes idle, only a fraction $[\lambda - (1 - s_2)]/\lambda$ of incoming jobs will go to a non-empty queue, which is exactly the probability of finding an empty virtual queue in this case.

3.1.4 Technical results

In this subsection we provide precise statements of our technical results.

Properties of the fluid solutions

The existence of fluid solutions will be established by showing that, almost surely, the limit of every convergent subsequence of sample paths of $S^n(\cdot)$ is a fluid solution (Proposition 3.3.4). In addition, the theorem that follows establishes uniqueness of fluid solutions for all initial conditions $s^0 \in \mathcal{S}^1$, characterizes the unique equilibrium of the fluid model, and states its global asymptotic stability. The regimes mentioned in the statement of the results in this subsection correspond to the different assumptions on memory and message rates described in the 2nd and 3rd columns of Table 3.1, respectively.

Theorem 3.1.1 (Existence, uniqueness, and stability of fluid solutions)**.** *A fluid solution, as described in Definition 3.1.1, exists and is unique for any initial condition $s^0 \in \mathcal{S}^1$. Furthermore, the fluid model has a unique equilibrium s^*, given by*

$$s_i^* = \lambda \left(\lambda P_0^*\right)^{i-1}, \quad \forall\, i \geq 1,$$

where $P_0^* = P_0(s^*)$ is given, for the three regimes considered, by:

(i) High Memory: $P_0^* = \left[1 - \frac{\beta(1-\lambda)}{\lambda}\right]^+$;

(ii) High Message: $P_0^* = 0$;

(iii) Constrained: $P_0^* = \left[\sum_{k=0}^{c} \left(\frac{\beta(1-\lambda)}{\lambda}\right)^k\right]^{-1}$.

This equilibrium is globally asymptotically stable, i.e.,

$$\lim_{t\to\infty} \|s(t) - s^*\|_w = 0,$$

for any initial condition $s^0 \in \mathcal{S}^1$.

The proof is given in sections 3.2 (uniqueness and stability) and 3.3 (existence).

Remark 3.1.1. Note that, if $\beta \geq \lambda/(1-\lambda)$, the High Memory regime also has $P_0^* = 0$ in equilibrium.

Approximation theorems

The three results in this subsection justify the use of the fluid model as an approximation to the finite stochastic system. The first one states that the evolution of the process $S^n(\cdot)$ is almost surely uniformly close, over any finite time horizon $[0, T]$, to the unique fluid solution $s(\cdot)$.

Theorem 3.1.2 (Convergence of sample paths)**.** *Fix* $T > 0$ *and* $s^0 \in \mathcal{S}^1$. *Under each of the three regimes, if*

$$\lim_{n\to\infty} \left\|S^n(0) - s^0\right\|_w = 0, \qquad a.s.,$$

then

$$\lim_{n\to\infty} \sup_{0\leq t\leq T} \|S^n(t) - s(t)\|_w = 0, \qquad a.s.,$$

where $s(\cdot)$ *is the unique fluid solution with initial condition* s^0.

The proof is given in Section 3.3.

Remark 3.1.2. On the technical side, the proof is somewhat involved because the process $\left(S^n(\cdot), M^n(\cdot)\right)$ is not the usual density-dependent Markov process studied by Kurtz [28] and which appears in the study of several dispatching policies (e.g., [34, 40, 47]). This is because $M^n(\cdot)$ is not scaled by n, and consequently evolves in a faster time scale. We are dealing instead with an infinite-level infinite-dimensional jump Markov process, which is a natural generalization of its finite-level

finite-dimensional counterpart studied in Chapter 8 of [38]. The fact that our process may have infinitely many levels (memory states) and is infinite-dimensional prevents us from directly applying known results. Furthermore, even if we truncated $S^n(\cdot)$ to be finite-dimensional as in [33], our process still would not satisfy the more technical hypotheses of the corresponding result in [38] (Theorem 8.15). Finally, the large deviations techniques used to prove Theorem 8.15 in [38] do not directly generalize to infinite dimensions. For all of these reasons, we will prove our fluid limit result directly, by using a coupling approach, as in [12] and [42]. Our results involve a separation of time scales similar to the ones in [46] and [26].

If we combine theorems 3.1.2 and 3.1.1, we obtain that after some time, the state of the finite system $S^n(t)$ can be approximated by the equilibrium of the fluid model s^*, because

$$S^n(t) \xrightarrow{n\to\infty} s(t) \xrightarrow{t\to\infty} s^*,$$

almost surely. If we interchange the order of the limits over n and t, we obtain the limiting behavior of the invariant distribution π_s^n of $S^n(t)$ as n increases. In the next proposition and theorem, we show that the result is the same, i.e., that

$$S^n(t) \xrightarrow{t\to\infty} \pi_s^n \xrightarrow{n\to\infty} s^*,$$

in distribution, so that the interchange of limits is justified.

The first step is to show that for every finite n, the stochastic process of interest is positive recurrent.

Proposition 3.1.3 (Stochastic stability)**.** *For every n, the process $\big(S^n(\cdot), M^n(\cdot)\big)$ is positive recurrent and therefore has a unique invariant distribution π^n.*

The proof is given in Subsection 3.4.1.

Given π^n, the unique invariant distribution of the process $\big(S^n(\cdot), M^n(\cdot)\big)$, let

$$\pi_s^n(\cdot) \triangleq \sum_{m=0}^{c_n} \pi^n(\cdot, m)$$

be the marginal for $S^n(\cdot)$. We have the following result concerning the convergence of this sequence of marginal distributions.

Theorem 3.1.4 (Convergence of invariant distributions)**.** *We have*

$$\lim_{n\to\infty} \pi_s^n = \delta_{s^*}, \quad \textit{in distribution.}$$

The proof is given in Subsection 3.4.2.

Putting everything together, we conclude that when n is large, the fluid model is an accurate approximation to the stochastic system, for both the transient regime (Theorems 3.1.2 and 3.1.1) and the steady-state regime (Theorem 3.1.4). The relationship between the convergence results is depicted in the commutative diagram of Figure 3-3.

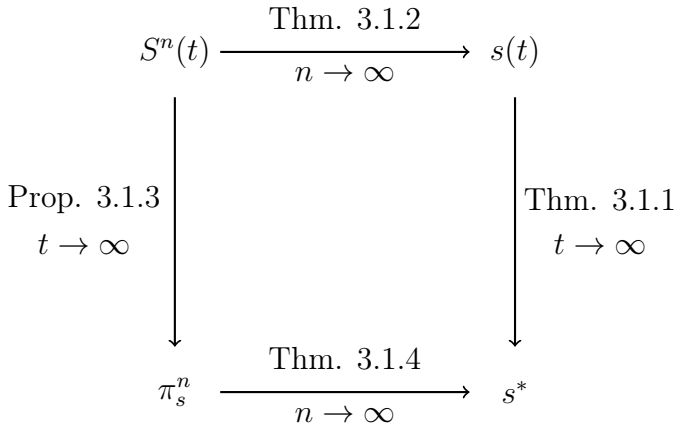

Figure 3-3: Relationship between the stochastic system and the fluid model.

3.1.5 Asymptotic queueing delay and phase transitions

In this subsection we use the preceding results to conclude that in two of the regimes considered, the asymptotic queueing delay is zero. For the third regime, the asymptotic queueing delay is positive and we examine its dependence on various policy parameters.

Queueing delay

Having shown that we can approximate the stochastic system by its fluid model for large n, we can analyze the equilibrium of the latter to approximate the queueing delay under our policy.

For any given n, we define the **queueing delay** of a job, generically denoted by $\mathbb{E}[W^n]$, as the mean time that a job spends in queue until its service starts. Here the expectation is taken with respect to the steady-state distribution, whose existence and uniqueness is guaranteed by Proposition 3.1.3. Then, the **asymptotic queueing delay** is defined as

$$\mathbb{E}[W] \triangleq \limsup_{n\to\infty} \mathbb{E}[W^n].$$

This asymptotic queueing delay can be obtained from the equilibrium s^* of the fluid model as follows. For a fixed n, the expected number of jobs in the system in steady-state is

$$\mathbb{E}\left[\sum_{i=1}^{\infty} nS_i^n\right].$$

Furthermore, the queueing delay of a job is equal to the total time it spends in the system minus the expected service time (which is 1). Using Little's Law, we obtain that the queueing delay is

$$\mathbb{E}[W^n] = \frac{1}{\lambda n}\mathbb{E}\left[\sum_{i=1}^{\infty} nS_i^n\right] - 1 = \frac{1}{\lambda}\mathbb{E}\left[\sum_{i=1}^{\infty} S_i^n\right] - 1.$$

Taking the limit as $n \to \infty$, and interchanging the limit, summation, and expectation, we obtain

$$\mathbb{E}[W] = \frac{1}{\lambda}\left(\sum_{i=1}^{\infty} s_i^*\right) - 1. \tag{3.3}$$

The validity of these interchanges is established the following lemma.

Lemma 3.1.5. *We have*

$$\lim_{n\to\infty} \mathbb{E}\left[\sum_{i=1}^{\infty} S_i^n\right] = \sum_{i=1}^{\infty} s_i^*.$$

Proof. By Fubini's theorem, we have

$$\lim_{n\to\infty} \mathbb{E}\left[\sum_{i=1}^{\infty} S_i^n\right] = \lim_{n\to\infty} \sum_{i=1}^{\infty} \mathbb{E}\left[S_i^n\right].$$

Due to the symmetric nature of the invariant distribution π^n, we have

$$\begin{aligned}
\mathbb{E}\left[S_i^n\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_{[i,\infty)}\left(\mathbf{Q}_j^n\right)\right] \\
&= \mathbb{E}\left[\mathbb{1}_{[i,\infty)}\left(\mathbf{Q}_1^n\right)\right] \\
&= \pi^n\left(\mathbf{Q}_1^n \geq i\right) \\
&\leq \left(\frac{1}{2-\lambda}\right)^{i/2},
\end{aligned}$$

where the inequality is established in Lemma 3.4.1. We can therefore apply the dominated convergence theorem to interchange the limit with the first summation, and obtain

$$\lim_{n\to\infty} \sum_{i=1}^{\infty} \mathbb{E}\left[S_i^n\right] = \sum_{i=1}^{\infty} \lim_{n\to\infty} \mathbb{E}\left[S_i^n\right],$$

We already know that S_i^n converges to s^*, in distribution (Theorem 3.1.4). Then, using a variant of the dominated convergence theorem for convergence in distribution, and the fact that we always have $S_i^n \leq 1$, we can finally interchange the limit and the expectation and obtain

$$\sum_{i=1}^{\infty} \lim_{n\to\infty} \mathbb{E}\left[S_i^n\right] = \sum_{i=1}^{\infty} s_i^*.$$

□

As a corollary, we obtain that if we have a superlinear message rate or a superlogarithmic number of memory bits, the RCPB policy results in zero asymptotic queueing delay.

Corollary 3.1.6. *For the High Memory regime with $\beta \geq \lambda/(1-\lambda)$, and for the High Message regime, the asymptotic queueing delay is zero, i.e., $\mathbb{E}[W] = 0$.*

Proof. From Theorem 3.1.1, we have $P_0^* = 0$ and therefore, $s_1^* = \lambda$ and $s_i^* = 0$, for $i \geq 2$. The result follows from Equation (3.3). □

The asymptotic queueing delay in the Constrained regime

According to Equation (3.3) and Theorem 3.1.1, the asymptotic queueing delay is given by

$$\mathbb{E}[W] = \frac{1}{\lambda} \sum_{i=1}^{\infty} s_i^* - 1 = \sum_{i=1}^{\infty} (\lambda P_0^*)^{i-1} - 1 = \frac{\lambda P_0^*}{1 - \lambda P_0^*}, \tag{3.4}$$

and is positive in the Constrained regime. Nevertheless, the dependence of the queueing delay on the various parameters has some remarkable properties, which we proceed to study.

Suppose that the message rate of each idle server is $\beta = \alpha/(1-\lambda)$ for some constant $\alpha > 0$. Since a server is idle (on average) a fraction $1 - \lambda$ of the time, the resulting average message rate at each server is α, and the overall (system-wide) average message rate is αn. We can rewrite the equilibrium probability P_0^* in Theorem 3.1.1 as

$$P_0^* = \left[\sum_{j=0}^{c} \left(\frac{\alpha}{\lambda} \right)^j \right]^{-1}.$$

This, together with Equation (3.4) and some algebra, implies that

$$\mathbb{E}[W] = \lambda \left[1 - \lambda + \sum_{j=1}^{c} \left(\frac{\alpha}{\lambda} \right)^j \right]^{-1}. \tag{3.5}$$

Phase transition of the queueing delay for $\lambda \uparrow 1$. We have a phase transition between $\alpha = 0$ (which corresponds to uniform random routing) and $\alpha > 0$. In the first case, we have the usual M/M/1 queueing delay: $\lambda/(1-\lambda)$. However, when $\alpha > 0$, the queueing delay is upper bounded uniformly in λ as follows:

$$\mathbb{E}[W] \leq \left(\sum_{k=1}^{c} \alpha^k \right)^{-1}. \tag{3.6}$$

This is established by noting that the expression in Equation (3.5) is monotonically increasing in λ and then setting $\lambda = 1$. Note that when α is fixed, the total message rate is the same, αn, for all $\lambda < 1$. This is a key qualitative improvement over all other resource constrained policies in the literature; see our discussion of the power-of-d-choices policy at the end of this subsection.

Phase transition in the memory-delay tradeoff. When λ and α are held fixed, the asymptotic queueing delay in Equation (3.5) decreases with c. This represents a tradeoff between the asymptotic queueing delay $\mathbb{E}[W]$, and the number of memory bits, which is equal to $\lceil c \log_2(n) \rceil$ for the Constrained regime. However, the rate at which the queueing delay decreases with c depends critically on the value of α, and we have a phase transition when $\alpha = \lambda$.

(i) If $\alpha < \lambda$, then

$$\lim_{c\to\infty} \mathbb{E}[W] = \frac{\lambda(\lambda-\alpha)}{(1-\lambda)(\lambda-\alpha)+1}.$$

Consequently, if $\alpha < \lambda$, it is impossible to drive the queueing delay to 0 by increasing the value of c, i.e., by increasing the amount of memory available.

(ii) If $\alpha = \lambda$, we have

$$\mathbb{E}[W] = \frac{1}{1-\lambda+c} \leq \frac{1}{c},$$

and thus the queueing delay converges to 0 at the rate of $1/c$, as $c \to \infty$.

(iii) If $\alpha > \lambda$, we have

$$\mathbb{E}[W] = \lambda \left[1 - \lambda + \sum_{j=1}^{c} \left(\frac{\alpha}{\lambda}\right)^j\right]^{-1} \leq \left(\frac{\lambda}{\alpha}\right)^c, \tag{3.7}$$

and thus the queueing delay converges exponentially fast to 0, as $c \to \infty$.

This phase transition is due to the fact that the queueing delay depends critically on P_0^*, the probability that there are no tokens left in the dispatcher's virtual queue. In equilibrium, the number of tokens in the virtual queue evolves as a birth-death process with birth rate α, death rate λ, and maximum population c, and has an invariant

distribution which is geometric with ratio α/λ. As a result, as soon as α becomes larger than λ, this birth-death process has an upward drift, and the probability of being at state 0 (no tokens present) decays exponentially with the size of its state space. This argument captures the essence of the phase transition at $\beta = \lambda/(1-\lambda)$ for the High Memory regime.

Comparison with the power-of-d-choices. The power-of-d-choices policy queries d random servers at the time of each arrival and sends the arriving job to the shortest of the queried queues. As such, it involves $2\lambda dn$ messages per unit time. For a fair comparison, we compare this policy to our RCPB policy with $\alpha = 2\lambda d$, so that the two policies have the same average message rate.

The asymptotic queueing delay for the power-of-d-choices policy was shown in [34, 44] to be

$$\mathbb{E}[W_{\text{Pod}}] = \sum_{i=1}^{\infty} \lambda^{\frac{d^i - d}{d-1}} - 1 \geq \lambda^d.$$

Thus, the queueing delay decreases at best exponentially with d, much like the queueing delay decreases exponentially with c in our scheme (cf. Equation (3.7)). However, increasing d increases the number of messages sent, unlike our policy where the average message rate remains fixed at αn.

Furthermore, the asymptotic queueing delay in the power-of-d-choices when $\lambda \uparrow 1$ is shown in [34] to satisfy

$$\lim_{\lambda \uparrow 1} \frac{\mathbb{E}[W_{\text{Pod}}]}{\log\left(\frac{1}{1-\lambda}\right)} = \frac{1}{\log(d)}.$$

For any fixed d, this is an exponential improvement over the queueing delay of randomized routing, but the queueing delay is still unbounded as $\lambda \uparrow 1$. In contrast, the queueing delay of our scheme has a constant upper bound, independent of λ.

In conclusion, if we set $\alpha = 2d\lambda$, so that our policy and the power-of-d policy use the same number of messages per unit of time, our policy results in much better asymptotic queueing delay, especially when $\lambda \uparrow 1$, even if c is as small as 1.

Numerical results. We implemented three policies in Matlab: the power-of-2-choices [34, 44], our RCPB policy, and the PULL policy [40]. We evaluate the algorithms in a system with 500 servers. In our algorithm we used $c = 2$, and $\alpha = \lambda$, so it has the same average message rate as the PULL policy (500λ messages per unit of time), which is 4 times less than what the power-of-2-choices utilizes. In Figure 3-4 we plot the queueing delay as a function of $\log(1/(1-\lambda))$.

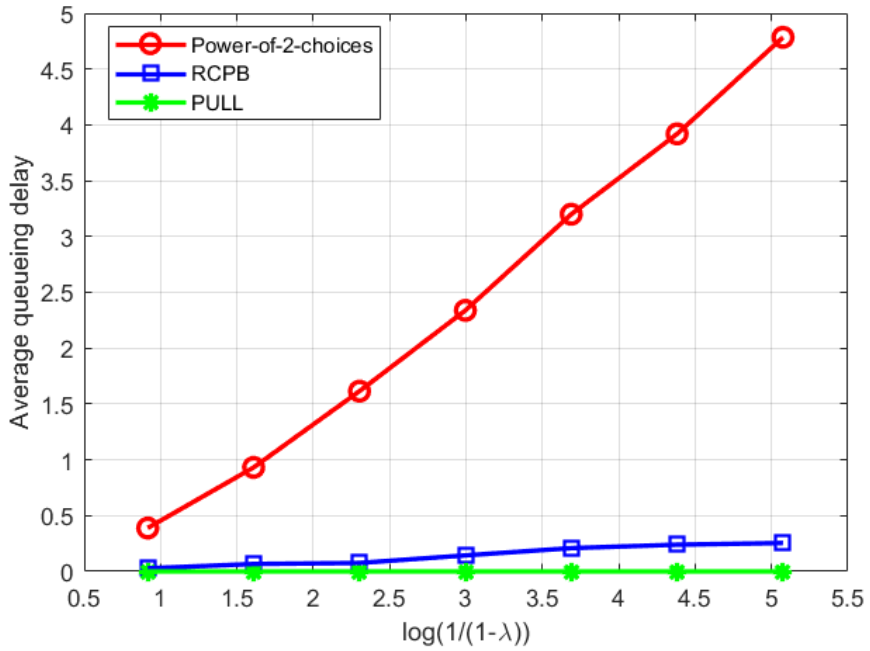

Figure 3-4: Average queueing delay of the power-of-2-choices policy (red circles) vs. our policy (blue squares) vs. PULL (green asterisks).

As expected, the queueing delay remains uniformly bounded under our RCPB policy (blue squares). This is achieved with only $\lceil 2\log_2(500) \rceil = 18$ bits of memory. Furthermore, with this small amount of memory we are also close to the performance of the PULL algorithm, which requires 500 bits of memory.

3.2 Proof of part of Theorem 3.1.1

The proof of Theorem 3.1.1 involves mostly deterministic arguments; these are developed in lemmas 3.2.1 and 3.2.3, and Proposition 3.2.5, which establish uniqueness of fluid solutions, existence and uniqueness of a fluid-model equilibrium, and asymptotic

stability, respectively. The proof of existence of fluid solutions relies on a stochastic argument and is developed in Section 3.3, in parallel with the proof of Theorem 3.1.2.

3.2.1 Uniqueness of solutions

Lemma 3.2.1. *If there exists a fluid solution (cf. Definition 3.1.1) with initial condition $s^0 \in \mathcal{S}^1$, it is unique.*

Proof. The fluid model is of the form $\dot{s}(\cdot) = F\big(s(\cdot)\big)$, where the function $F : \mathcal{S}^1 \to [-1, \lambda]^{\mathbb{Z}_+}$ is defined by

$$\begin{aligned} F_0(s) =& 0, \\ F_1(s) =& \lambda\big(1 - P_0(s)\big) + \lambda(1 - s_1)P_0(s) - (s_1 - s_2), \\ F_i(s) =& \lambda(s_{i-1} - s_i)P_0(s) - (s_i - s_{i+1}), \qquad \forall\, i \geq 2, \end{aligned} \tag{3.8}$$

and where $P_0(s)$ is given for the three regimes by:

(i) High Memory: $P_0(s) = \left[1 - \frac{\beta(1-s_1)}{\lambda}\right]^+$.

(ii) High Message: $P_0(s) = \left[1 - \frac{1-s_2}{\lambda}\right]^+ \mathbb{1}_{\{1\}}(s_1)$.

(iii) Constrained: $P_0(s) = \left[\sum_{k=0}^{c} \left(\frac{\beta(1-s_1)}{\lambda}\right)^k\right]^{-1}$.

The function $P_0(s)$ for the High Memory regime is continuous and piecewise linear in s_1, so it is Lipschitz continuous in s, over the set $\mathcal{S}^1$. Similarly, $P_0(s)$ for the Constrained regime is also Lipschitz continuous in s, because $P_0(s)$ is a rational function of s_1 and the denominator is lower bounded by 1. However, $P_0(s)$ for the High Message regime is only Lipschitz continuous "almost everywhere" in $\mathcal{S}^1$; more precisely, it is Lipschitz continuous everywhere except on the lower dimensional set

$$D \triangleq \left\{ s \in \mathcal{S}^1 : s_1 = 1 \text{ and } s_2 > 1 - \lambda \right\}.$$

Moreover, $P_0(s)$ restricted to D is also Lipschitz continuous.

Suppose that $P_0(s)$ is Lipschitz continuous with constant L on some subset $\mathcal{S}_0$ of $\mathcal{S}^1$. Then, for every $s, s' \in \mathcal{S}_0$ and any $i \geq 1$, we have

$$\begin{aligned}
|F_i(s) - F_i(s')| &= \big|-\lambda P_0(s)\mathbb{1}_{\{1\}}(i) + \lambda(s_{i-1} - s_i)P_0(s) - (s_i - s_{i+1}) \\
&\qquad +\lambda P_0\left(s'\right)\mathbb{1}_{\{1\}}(i) - \lambda\left(s'_{i-1} - s'_i\right)P_0\left(s'\right) + \left(s'_i - s'_{i+1}\right)\big| \\
&\leq |P_0(s) - P_0(s')| + \left|(s_{i-1} - s_i)P_0(s) - \left(s'_{i-1} - s'_i\right)P_0(s')\right| \\
&\qquad + |s_i - s'_i| + \left|s_{i+1} - s'_{i+1}\right| \\
&\leq 2\,|P_0(s) - P_0(s')| + \left|s_{i-1} - s'_{i-1}\right| + 2\,|s_i - s'_i| + \left|s_{i+1} - s'_{i+1}\right| \\
&\leq 2L\|s - s'\|_w + \left|s_{i-1} - s'_{i-1}\right| + 2\,|s_i - s'_i| + \left|s_{i+1} - s'_{i+1}\right|.
\end{aligned}$$

Then,

$$\begin{aligned}
\|F(s) - F(s')\|_w &= \sqrt{\sum_{i=0}^{\infty} \frac{|F_i(s) - F_i(s')|^2}{2^i}} \\
&\leq \sqrt{\sum_{i=1}^{\infty} \frac{\left(2L\|s - s'\|_w + \left|s_{i-1} - s'_{i-1}\right| + 2\,|s_i - s'_i| + \left|s_{i+1} - s'_{i+1}\right|\right)^2}{2^i}} \\
&\leq \sqrt{12\sum_{i=1}^{\infty} \frac{4L^2\|s - s'\|_w^2 + \left|s_{i-1} - s'_{i-1}\right|^2 + 4\,|s_i - s'_i|^2 + \left|s_{i+1} - s'_{i+1}\right|^2}{2^i}} \\
&\leq \|s - s'\|_w \sqrt{12(4L^2 + 2 + 4 + 1)},
\end{aligned}$$

where the second inequality comes from the fact that $(w+x+y+z)^2 \leq 12\big(w^2+x^2+y^2+z^2\big)$, for all $(w, x, y, z) \in \mathbb{R}^4$. This means that F is also Lipschitz continuous on the set $\mathcal{S}_0$.

For the High Memory and Constrained regimes, we can set $\mathcal{S}_0 = \mathcal{S}^1$, and by the preceding discussion, F is Lipschitz continuous on $\mathcal{S}^1$. At this point we cannot immediately guarantee the uniqueness of solutions because F is just Lipschitz continuous on a subset ($\mathcal{S}^1$) of the Hilbert space $(\ell^2_w, \|\cdot\|_w)$. However, we can use Kirszbraun's theorem [27] to extend F to a Lipschitz continuous function $\overline{F}$ on the entire Hilbert space. If we have two different solutions to the equation $\dot{s} = F(s)$ which stay in $\mathcal{S}^1$, we

would also have two different solutions to the equation $\dot{s} = \overline{F}(s)$. Since $\overline{F}$ is Lipschitz continuous, this would contradict the Picard-Lindelöff uniqueness theorem [30]. This establishes the uniqueness of fluid solutions for the High Memory and Constrained regimes.

Note that the preceding argument can also be used to show uniqueness of solutions for any differential equation with a Lipschitz continuous drift in an arbitrary subset of the Hilbert space $(\ell^2_w, \|\cdot\|_w)$, as long as we only consider solutions that stay in that set. This fact will be used in the rest of the proof.

From now on, we concentrate on the High Message regime. In this case, the drift $F(s)$ is Lipschitz continuous only "almost everywhere," and a solution will in general be non-differentiable. In particular, results on the uniqueness of classical (differentiable) solutions do not apply. Our proof will rest on the fact that non-uniqueness issues can only arise when a trajectory hits the closure of the set where the drift $F(s)$ is not Lipschitz continuous, which in our case is just the closure of D:

$$\overline{D} = \left\{ s \in \mathcal{S}^1 : s_1 = 1 \text{ and } s_2 \geq 1 - \lambda \right\}.$$

We now partition the space $\mathcal{S}^1$ into three subsets, $\mathcal{S}^1 \backslash \overline{D}$, D, and $\overline{D} \backslash D$, and characterize the behavior of potential trajectories depending on the initial condition. Note that we only consider fluid solutions, and these always stay in the set $\mathcal{S}^1$, by definition. Therefore, we only need to establish the uniqueness of solutions that stay in $\mathcal{S}^1$.

Claim 3.2.2. *For any fluid solution $s(\cdot)$ in the High Message regime, and with initial condition $s^0 \in \overline{D}$, we have the following.*

i) If $s^0 \in D$, then $s(\cdot)$ either stays in D forever or hits $\overline{D} \backslash D$ at some finite time. In particular, it cannot go directly from D to $\mathcal{S}^1 \backslash \overline{D}$.

ii) If $s^0 \in \overline{D} \backslash D$, then $s(\cdot)$ stays in $\mathcal{S}^1 \backslash D$ forever. In particular, it can never return to D.

Proof.

i) Suppose that $s^0 \in D$, i.e., $s_1^0 = 1$ and $s_2^0 > 1 - \lambda$. Let t_{D^c} be the exit time from D, and suppose that it is finite. Note that, by continuity of solutions, $s_1(t_{D^c}) = 1$. We will show that $s_2(t_{D^c}) = 1 - \lambda$, so that the trajectory hits $\overline{D} \backslash D$. Suppose, in order to derive a contradiction, that this is not the case and, therefore, $s_2(t_{D^c}) > 1 - \lambda$. Then, due to the continuity of solutions, there exists some time $t_1 > t_{D^c}$ such that $s_1(t_1) < 1$ and $s_2(t) > 1 - \lambda$, for all $t \in [t_{D^c}, t_1]$. Let

$$t_0 \triangleq \sup\{t \leq t_1 : s_1(t) = 1\}$$

be the last time before t_1 that $s_1(t)$ is equal to 1. Then we have $s_1(t_0) = 1$, and $s_1(t) < 1$ for all $t \in (t_0, t_1]$. Since the drift F is continuous for all $s_1 < 1$, all times in $(t_0, t_1]$ are regular. On the other hand, for all $t \in (t_0, t_1]$, we have $s_1(t) < 1$ and thus $P_0(s(t)) = 0$, which together with $s_2(t) > 1 - \lambda$ implies that

$$\frac{ds_1(t)}{dt} = \lambda - (s_1(t) - s_2(t)) > 0,$$

for all $t \in (t_0, t_1]$. This contradicts the relations $s_1(t_1) < 1 = s_1(t_0)$, and establishes that $s_1(t_D) = 1$. Therefore the fluid solution s either stays in D forever or it exits D with $s_2 = 1 - \lambda$.

ii) Suppose now that $s^0 \in \overline{D} \backslash D$, i.e., $s_1^0 = 1$ and $s_2^0 = 1 - \lambda$. It is enough to show that $s_2(t) \leq 1 - \lambda$, for all $t \geq 0$. Let

$$\tau_2(\epsilon) \triangleq \min\{t \geq 0 : s_2(t) = 1 - \lambda + \epsilon\}$$

be the first time s_2 reaches $1 - \lambda + \epsilon$. Suppose, in order to derive a contradiction, that there exists $\epsilon^* > 0$ such that $\tau_2(\epsilon^*) < \infty$. Then, due to the continuity of s_2, we also have $\tau_2(\epsilon) < \infty$, for all $\epsilon \leq \epsilon^*$. Since s_2 is differentiable almost everywhere, we can choose ϵ such that $\tau_2(\epsilon)$ is a regular time with $F_2\big(s(\tau_2(\epsilon))\big) >$

0. Using the expression (3.8) for F_2, we obtain

$$\begin{aligned}
0 &< \lambda\Big(s_1(\tau_2(\epsilon)) - s_2(\tau_2(\epsilon))\Big)\left(1 - \frac{1 - s_2(\tau_2(\epsilon))}{\lambda}\right)\mathbb{1}_{\{1\}}(s_1(\tau_2(\epsilon))) \\
&\qquad\qquad - \Big(s_2(\tau_2(\epsilon)) - s_3(\tau_2(\epsilon))\Big) \\
&\leq \lambda\Big(1 - s_2(\tau_2(\epsilon))\Big)\left(1 - \frac{1 - s_2(\tau_2(\epsilon))}{\lambda}\right) - \Big(s_2(\tau_2(\epsilon)) - s_3(\tau_2(\epsilon))\Big) \\
&= \lambda - 1 + s_3(\tau_2(\epsilon)) + s_2(\tau_2(\epsilon))\Big(1 - \lambda - s_2(\tau_2(\epsilon))\Big) \\
&< \lambda - 1 + s_3(\tau_2(\epsilon)),
\end{aligned}$$

or $s_3(\tau_2(\epsilon)) > 1 - \lambda$. On the other hand, we have $s_3(0) \leq s_2(0) = 1 - \lambda$. Combining these two facts, we obtain that $s_3(\tau_2(\epsilon)) > s_3(0)$, i.e., that $s_3(\cdot)$ increased between times 0 and $\tau_2(\epsilon)$. As a result, and since $s_3(\cdot)$ is differentiable almost everywhere, there exists another regular time $\tau_3(\epsilon) \leq \tau_2(\epsilon)$ such that $s_3(\tau_3(\epsilon)) > 1 - \lambda$ and $F_3(s(\tau_3(\epsilon))) > 0$. Proceeding inductively, we can obtain a sequence of nonincreasing regular times $\tau_2(\epsilon) \geq \tau_3(\epsilon) \geq \cdots \geq 0$ such that $s_k(\tau_k(\epsilon)) > 1 - \lambda$, for all $k \geq 2$. Let $\tau_\infty(\epsilon)$ be the limit of this sequence of regular times. Since all coordinates of the fluid solutions are Lipschitz continuous with the same constant L, we have

$$s_k(\tau_\infty) > 1 - \lambda - L\big(\tau_k(\epsilon) - \tau_\infty\big),$$

for all $k \geq 2$. Since $\tau_k(\epsilon) \to \tau_\infty$, there exists some $k^* \geq 2$ such that $s_k(\tau_\infty) > (1 - \lambda)/2 > 0$, for all $k \geq k^*$. But then,

$$\|s(\tau_\infty)\|_1 \geq \sum_{k=k^*}^{\infty} \frac{1 - \lambda}{2} = \infty.$$

This contradicts the fact that $s(\tau_\infty) \in \mathcal{S}^1$, and it follows that we must have $s_2(t) \leq 1 - \lambda$ for all $t \geq 0$.

□

The uniqueness of a solution over the whole time interval $[0, \infty)$ for the High

Message regime can now be obtained by concatenating up to three unique trajectories, depending on the initial condition s^0.

a) Suppose that $s^0 \in \mathcal{S}^1 \backslash \overline{D}$, and let $t_{\overline{D}}$ be the hitting time of $\overline{D}$, i.e.,

$$t_{\overline{D}} \triangleq \inf \left\{ t \geq 0 : s(t) \in \overline{D} \text{ with } s(0) = s^0 \right\}.$$

Since $F|_{\mathcal{S}^1 \backslash \overline{D}}$ (the restriction of the original drift F to the set $\mathcal{S}^1 \backslash \overline{D}$) is Lipschitz continuous, we have the uniqueness of a solution over the time interval $[0, t_{\overline{D}})$, by using the same argument as for the other regimes. If $t_{\overline{D}} = \infty$, then we are done. Otherwise, we have $s(t_{\overline{D}}) \in \overline{D}$; the uniqueness of a solution over the time interval $[t_{\overline{D}}, \infty)$ will immediately follow from the uniqueness of a solution with initial condition in $\overline{D}$.

b) Suppose that $s^0 \in D$. Due to part i) of Claim 3.2.2, a solution can only exit the set D by hitting $\overline{D} \backslash D$, and never by going back directly into $\mathcal{S}^1 \backslash \overline{D}$. Let $t_{\overline{D} \backslash D}$ be the hitting time of $\overline{D} \backslash D$. Since $F|_D$ is Lipschitz continuous, we have uniqueness of a solution over the time interval $[0, t_{\overline{D} \backslash D})$. As in case a), if $t_{\overline{D} \backslash D} = \infty$ we are done. Otherwise, the uniqueness of a solution over the time interval $[t_{\overline{D} \backslash D}, \infty)$ will immediately follow from the uniqueness of a solution with initial condition in $\overline{D} \backslash D$.

c) Suppose that $s^0 \in \overline{D} \backslash D$. Due to part ii) of Claim 3.2.2, a solution stays in $\mathcal{S}^1 \backslash D$ forever. As a result, since $F|_{\mathcal{S}^1 \backslash D}$ is Lipschitz continuous, uniqueness follows.

□

The intuition behind the preceding proof, for the High Message regime, is as follows. A non-differentiable solution may arise if the system starts with a large fraction of the servers having at least two jobs. In that case, the rate $s_1(t) - s_2(t)$ at which the servers become idle is smaller than the rate λ at which idle servers become busy. As a consequence, the fraction $s_1(t)$ of busy servers increases until it possibly reaches its maximum of 1, and stays there until the fraction of servers with exactly

one job, which is now $1 - s_2(t)$, exceeds the total arrival rate λ; after that time servers become idle at a rate faster than the arrival rate. This scenario is illustrated in Figure 3-5.

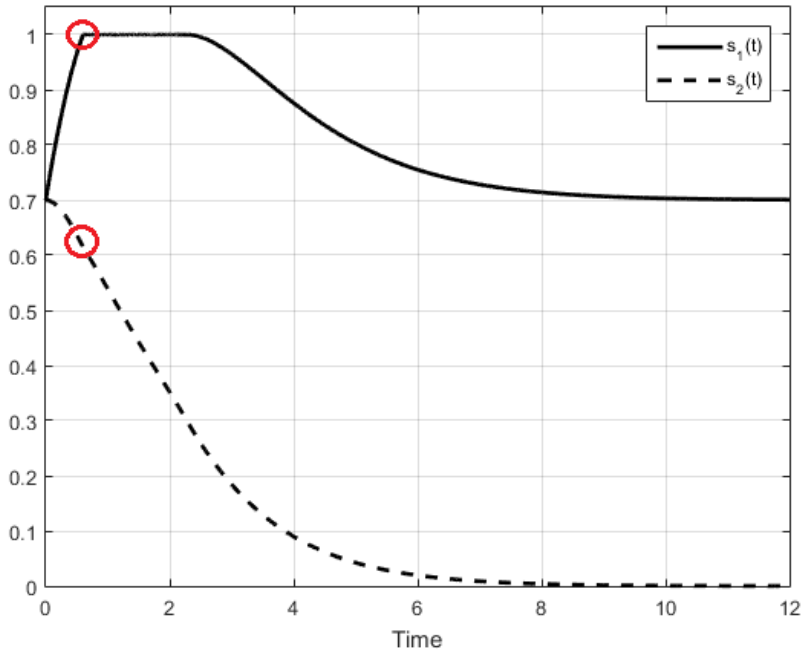

Figure 3-5: An example of a non-differentiable solution for the High Message regime, with $\lambda = 0.9$, $s_1(0) = s_2(0) = s_3(0) = 0.7$, and $s_i(0) = 0$ for all $i \geq 4$. The solution is non-differentiable at the points indicated by the circles.

3.2.2 Existence, uniqueness, and characterization of an equilibrium

Lemma 3.2.3. *The fluid model has a unique equilibrium $s^* \in \mathcal{S}^1$, given by*

$$s_i^* = \lambda(\lambda P_0^*)^{i-1}, \quad \forall\, i \geq 1,$$

where $P_0^* \triangleq P_0(s^*)$ is given by

(i) High Memory: $P_0^* = \left[1 - \frac{\beta(1-\lambda)}{\lambda}\right]^+$.

(ii) High Message: $P_0^* = 0$.

(iii) Constrained: $P_0^* = \left[\sum_{k=0}^{c} \left(\frac{\beta(1-\lambda)}{\lambda}\right)^k\right]^{-1}$.

Proof. A point $s^* \in \mathcal{S}^1$ is an equilibrium if and only if

$$\begin{aligned} 0 =& \lambda\big(1 - P_0(s^*)\big) + \lambda(1 - s_1^*)P_0(s^*) - (s_1^* - s_2^*), \\ 0 =& \lambda(s_{i-1}^* - s_i^*)P_0(s^*) - (s_i^* - s_{i+1}^*), \qquad \forall\, i \geq 2. \end{aligned}$$

Since $s^* \in \mathcal{S}^1$, the sum $\sum_{i=0}^{\infty}(s_i^* - s_{i+1}^*)$ is absolutely convergent, even when we consider all the terms separately, i.e., when we consider s_i^* and $-s_{i+1}^*$ as separate terms, for each $i \geq 0$. Thus, we can obtain equivalent equilibrium conditions by summing these equations over all coordinates $j \geq i$, for any fixed $i \geq 1$. We then obtain that s^* is an equilibrium if and only if

$$0 = \lambda\big(1 - P_0(s^*)\big) + \lambda P_0(s^*) \sum_{j=1}^{\infty}(s_{j-1}^* - s_j^*) - \sum_{j=1}^{\infty}(s_j^* - s_{j+1}^*), \tag{3.9}$$

$$0 = \lambda P_0(s^*) \sum_{j=i}^{\infty}(s_{j-1}^* - s_j^*) - \sum_{j=i}^{\infty}(s_j^* - s_{j+1}^*), \qquad \forall\, i \geq 2. \tag{3.10}$$

Since the sums are absolutely convergent, we can rearrange the terms in equations (3.9) and (3.10) to obtain that $s^* \in \mathcal{S}^1$ is an equilibrium if and only if

$$\begin{aligned} 0 &= \lambda - s_1^*, \\ 0 &= \lambda P_0(s^*)s_{i-1}^* - s_i^*, \quad \forall\, i \geq 2. \end{aligned}$$

These conditions yield $s_1^* = \lambda < 1$, and

$$s_i^* = \lambda(\lambda P_0(s^*))^{i-1}, \quad \forall\, i \geq 1,$$

which concludes the proof. □

3.2.3 Asymptotic stability of the equilibrium

We will establish global asymptotic stability by sandwiching a fluid solution between two solutions that converge to s^*, similar to the argument in [44]. Towards this purpose, we first establish a monotonicity result.

Lemma 3.2.4. *Suppose that s^1 and s^2 are two fluid solutions with $s^1(0) \geq s^2(0)$. Then $s^1(t) \geq s^2(t)$, for all $t \geq 0$.*

Proof. It is known that uniqueness of solutions implies their continuous dependence on initial conditions, not only for the classical solutions in the High Memory and Constrained regimes, but also for the non-differentiable solutions of the High Message regime (see Chapter 8 of [16]). Using this fact, it can be seen that it is enough to verify that $s^1(t) \geq s^2(t)$ when $s^1(0) > s^2(0)$, which we henceforth assume, under our particular definition of ">" in Section 2.1. Let us define

$$t_1 \triangleq \inf\left\{t \geq 0 : s_k^1(t) < s_k^2(t), \text{ for some } k \geq 1\right\}.$$

If $t_1 = \infty$, then $s^1(t) \geq s^2(t)$ for all $t \geq 0$, and the result holds. It remains to consider the case where $t_1 < \infty$, which we assume from now on.

By the definition of t_1, we have $s_i^1(t) \geq s_i^2(t)$ for all $i \geq 1$, and for all $t \leq t_1$. Since $P_0(s)$ is nondecreasing in s, this implies that $P_0(s^1(t)) \geq P_0(s^2(t))$, for all $t \leq t_1$. Then, for all regular times $t \leq t_1$ and any $i \geq 2$, and also using the fact that s_i is nonincreasing in i, we have

$$\begin{aligned}
F_i\big(s^1(t)\big) - F_i\big(s^2(t)\big) &= \lambda\big[s_{i-1}^1(t) - s_i^1(t)\big]P_0(s^1(t)) + \big[s_{i+1}^1(t) - s_{i+1}^2(t)\big] \\
&\qquad - \lambda\big[s_{i-1}^2(t) - s_i^2(t)\big]P_0\big(s^2(t)\big) - \big[s_i^1(t) - s_i^2(t)\big] \\
&\geq \lambda\big[s_{i-1}^1(t) - s_i^1(t)\big]P_0\big(s^2(t)\big) \\
&\qquad - \lambda\big[s_{i-1}^2(t) - s_i^2(t)\big]P_0\big(s^2(t)\big) - \big[s_i^1(t) - s_i^2(t)\big] \\
&\geq -\lambda P_0\big(s^2(t)\big)\big[s_i^1(t) - s_i^2(t)\big] - \big[s_i^1(t) - s_i^2(t)\big] \\
&\geq -2\big[s_i^1(t) - s_i^2(t)\big].
\end{aligned}$$

Then, by Grönwall's inequality we have

$$s_i^1(t) - s_i^2(t) \geq e^{-2t}\big[s_i^1(0) - s_i^2(0)\big], \quad \forall i \geq 2, \tag{3.11}$$

for all $t \leq t_1$. This implies that $s_i^1(t) - s_i^2(t) > 0$, for all $i \geq 2$ and for all $t \leq t_1$. It

follows that, at time t_1, we must have $s_1^1(t_1) = s_1^2(t_1)$. The rest of the proof considers separately two different cases.

Case 1: Suppose that we are dealing with the High Memory or the Constrained regime, or with the High Message regime with $s_1^1(t_1) = s_1^2(t_1) < 1$. Since $s_1^1(t_1) = s_1^2(t_1)$, we have $P_0\big(s^1(t_1)\big) = P_0\big(s^2(t_1)\big)$. Then, due to the continuity of s^1, s^2, and of P_0 (local continuity for the High Message regime), there exists $\epsilon > 0$ such that

$$\lambda s_1^2(t) P_0\big(s^2(t)\big) - \lambda s_1^1(t) P_0\big(s^1(t)\big) - \big[s_1^1(t) - s_1^2(t)\big] > -\epsilon,$$

and (using Equation (3.11)) $s_2^1(t) - s_2^2(t) > \epsilon$, for all $t \leq t_1$ sufficiently close to t_1. As a result, we have

$$F_1\big(s^1(t)\big) - F_1\big(s^2(t)\big) = \lambda s_1^2(t) P_0\big(s^2(t)\big) - \lambda s_1^1(t) P_0\big(s^1(t)\big) \\ - \big[s_1^1(t) - s_1^2(t)\big] + \big[s_2^1(t) - s_2^2(t)\big] > 0, \quad (3.12)$$

for all $t < t_1$ sufficiently close to t_1. Therefore, $s_1^1 - s_1^2$ was increasing just before t_1. On the other hand, from the definition of t_1, we have $s_1^1(t_1) = s_1^2(t_1)$ and $s_1^1(t) \geq s_1^2(t)$ for all $t < t_1$. This is a contradiction, and therefore this case cannot arise.

Case 2: Suppose now that we are dealing with the High Message regime, and that $s_1^1(t_1) = s_1^2(t_1) = 1$. Since $t_1 < \infty$, we can pick a time $t_2 > t_1$, arbitrarily close to t_1, such that $s_1^1(t_2) < s_1^2(t_2)$. Let us define

$$t_1' \triangleq \sup\left\{t \leq t_2 : s_1^1(t) = s_1^2(t)\right\}.$$

Due to the continuity of s^1 and s^2, and since $s_1^1(t_1') = s_1^2(t_1')$ and $s_2^1(t_1) > s_2^2(t_1)$, there exists $\epsilon > 0$ such that $s_1^2(t) - s_1^1(t) < \epsilon$ and $s_2^1(t) - s_2^2(t) > \epsilon$, for all $t \in [t_1', t_2]$ (we can always take a smaller t_2, if necessary, so that this holds). Furthermore, since $s_1^1(t) < 1$ for all $t \in [t_1', t_2]$, we have $P_0\big(s^1(t)\big) = 0$, for all $t \in [t_1', t_2]$. Using these facts in Equation (3.12), we obtain $F_1\big(s^1(t)\big) - F_1\big(s^2(t)\big) \geq 0$, for all $t \in [t_1', t_2]$. Therefore, $s_1^1 - s_1^2$ is nondecreasing in that interval. This is a contradiction, because we have $s_1^1(t_1') = s_1^2(t_1')$ and $s_1^1(t_2) < s_1^2(t_2)$. Therefore, this case cannot arise either. $\square$

We will now show that we can "sandwich" any given trajectory $s(\cdot)$ between a smaller one $s^l(\cdot)$ and a larger one $s^u(\cdot)$ (according to our partial order $\geq$) and prove that both $s^l(t)$ and $s^u(t)$ converge to s^* as $t \to \infty$, to conclude that $s(t)$ converges to s^* as $t \to \infty$.

Proposition 3.2.5. *The equilibrium s^* of the fluid model is globally asymptotically stable, i.e.,*

$$\lim_{t\to\infty} \|s(t) - s^*\|_w = 0,$$

for all fluid solutions $s(\cdot)$.

Proof. Suppose that $s(0) = s^0 \in \mathcal{S}^1$. We define initial conditions $s^u(0)$ and $s^l(0)$ by letting

$$s_i^u(0) \triangleq \max\{s_i(0), s_i^*\}, \quad \text{and} \quad s_i^l(0) \triangleq \min\{s_i(0), s_i^*\},$$

for all i. We then have $s^u(0) \geq s^0 \geq s^l(0)$, $s^u(0) \geq s^* \geq s^l(0)$, and $s^u(0), s^l(0) \in \mathcal{S}^1$. Due to monotonicity (Lemma 3.2.4), we obtain that $s^u(t) \geq s(t) \geq s^l(t)$ and $s^u(t) \geq s^* \geq s^l(t)$ for all $t \geq 0$. Thus it suffices to prove that $\|s^u(t) - s^*\|_w$ and $\left\|s^l(t) - s^*\right\|_w$ converge to 0 as $t \to \infty$.

For any $s \in \mathcal{S}^1$, we introduce an equivalent representation in terms of a vector v with components v_i defined by

$$v_i \triangleq \sum_{j=i}^{\infty} s_j, \qquad i \geq 1.$$

Note that any $s \in \mathcal{S}^1$ can be fully recovered from v. Therefore, we can work with a representation $v^u(t)$, $v^l(t)$, and v^*, of the vectors $s^u(t)$, $s^l(t)$, and s^*, respectively.

From the proof of Lemma 3.2.1, we know that a trajectory can be non-differentiable at most at a single point in time. This can occur only for the High Message regime, and only if the trajectory hits the set

$$D = \left\{s \in \mathcal{S}^1 : s_1 = 1 \text{ and } s_2 > 1 - \lambda\right\},$$

where the drift is discontinuous. In all other cases, the trajectories are not only differentiable, but also Lipschitz continuous (in time), with the same Lipschitz constant for all coordinates. Therefore, in order to prove the asymptotic stability of the solutions, which is a property of the limiting behavior as $t \to \infty$, we can assume that the trajectories are everywhere differentiable and Lipschitz continuous.

Our first step is to derive a differential equation for v_i. This requires the interchange of summation and differentiation, which we proceed to justify. For any $i \geq 1$, we define a sequence of functions $\{f_k^{(i)}(\cdot)\}_{k=1}^{\infty}$, as follows:

$$f_k^{(i)}(t) \triangleq \sum_{j=i}^{k} \frac{ds_j^u}{dt}(t), \qquad \forall\, t \geq 0.$$

Using equations (3.1) and (3.2), we obtain

$$\begin{aligned} f_k^{(1)}(t) &= \lambda - s_1^u(t) + \Big[s_{n+1}^u(t) - \lambda s_k^u(t) P_0\big(s^u(t)\big)\Big], \\ f_k^{(i)}(t) &= \lambda s_{i-1}^u(t) P_0\big(s^u(t)\big) - s_i^u(t) + \Big[s_{k+1}^u(t) - \lambda s_k^u(t) P_0\big(s^u(t)\big)\Big], \quad \forall\, i \geq 2. \end{aligned}$$

Since $s^u(t) \in \mathcal{S}^1$, for all t, we have the pointwise limits

$$\begin{aligned} \lim_{k\to\infty} f_k^{(1)}(t) &= \lambda - s_1^u(t), \\ \lim_{k\to\infty} f_k^{(i)}(t) &= \lambda s_{i-1}^u(t) P_0\big(s^u(t)\big) - s_i^u(t), \quad \forall\, i \geq 2. \end{aligned}$$

On the other hand, since all components of $s^u(\cdot)$ are Lipschitz continuous with the same constant, and since $P_0(s)$ is also Lipschitz-continuous, the functions in the sequence $\{f_k^{(i)}(\cdot)\}_{k=1}^{\infty}$ are equicontinuous, for any given i. Then, the Arzelà-Ascoli theorem allows us to conclude that $f_k^{(i)}(\cdot)$ also converges uniformly, over any compact interval of time, to their pointwise limits. Using the uniform convergence, and the fact that $s^u(0) \in \mathcal{S}^1$, we can interchange summation and differentiation (Theorem

7.17 in [37]) to obtain

$$\frac{dv_1^u}{dt}(t) = \frac{d}{dt}\sum_{j=1}^{\infty} s_j^u(t) = \sum_{j=1}^{\infty}\frac{ds_j^u}{dt}(t) = \lambda - s_1^u(t)$$
$$\frac{dv_i^u}{dt}(t) = \frac{d}{dt}\sum_{j=i}^{\infty} s_j^u(t) = \sum_{j=i}^{\infty}\frac{ds_j^u}{dt}(t) = \lambda s_{i-1}^u(t)P_0(s^u(t)) - s_i^u(t), \quad \forall\, i \geq 2.$$

Turning the above differential equations into integral equations, and using the facts $s_1^* = \lambda$ and $\lambda s_{i-1}^* P_0^* - s_i^* = 0$, we have

$$v_1^u(t) - v_1^u(0) = \int_0^t (s_1^* - s_1^u(\tau))\, d\tau,$$
$$v_i^u(t) - v_i^u(0) = \int_0^t \Big(\lambda\Big(s_{i-1}^u(\tau)P_0(s^u(\tau)) - s_{i-1}^* P_0^*\Big) - \big(s_i^u(\tau) - s_i^*\big)\Big)d\tau.$$

Note that from the definition of v_i, we have $v_1^u(t) \geq v_i^u(t)$. Furthermore, from Lemma 3.2.4, we have $s_1^u(t) \geq s_1^*$, so that $\dot{v}_1^u(t) \leq 0$, for all $t \geq 0$. It follows that

$$v_1^u(0) \geq v_1^u(t) \geq v_i^u(t) \geq v_i^u(t) - v_i^u(0) \geq -v_i^u(0),$$

for all t.

We will now use induction on i to prove coordinate-wise convergence, i.e., that $|s_i^u(t) - s_i^*|$ converges to 0, as $t \to \infty$, for all $i \geq 1$. We start with the base case, $i = 1$. We have $s_1^u(\tau) - s_1^* \geq 0$, for all $\tau \geq 0$. Using the fact $\dot{v}_1^u(t) \leq 0$, we see that $v_1^u(t)$ converges to some limit, which we denote by $v_1^u(\infty)$. Then,

$$0 \leq \int_0^{\infty} \big(s_1^u(\tau) - s_1^*\big)d\tau = v_1^u(0) - v_1^u(\infty) \leq v_1^u(0) < \infty,$$

which, together with the fact that s_1 is Lipschitz continuous, implies that $\big(s_1^u(\tau) - s_1^*\big)$ converges to zero, as $\tau \to \infty$.

We now consider some $i \geq 2$ and make the induction hypothesis that

$$\int_0^\infty \left(s_k^u(\tau) - s_k^*\right) d\tau < \infty, \quad \forall\, k \leq i-1. \tag{3.13}$$

Then,

$$-v_i^u(0) \leq v_i^u(t) - v_i^u(0) = \int_0^t \left(\lambda\Big(s_{i-1}^u(\tau) P_0(s^u(\tau)) - s_{i-1}^* P_0^*\Big) - \left(s_i^u(\tau) - s_i^*\right)\right) d\tau. \tag{3.14}$$

Adding and subtracting $\lambda s_{i-1}^* P_0(s^u(\tau))$ inside the integral, we obtain

$$\begin{aligned} -\, v_1^u(0) \leq \int_0^t \Big(&\lambda \left[s_{i-1}^u(\tau) - s_{i-1}^*\right] P_0(s^u(\tau)) \\ &+ \lambda \left[P_0(s^u(\tau)) - P_0^*\right] s_{i-1}^* - \left(s_i^u(\tau) - s_i^*\right)\Big) d\tau. \end{aligned} \tag{3.15}$$

Using Lemma 3.2.4, we have $s_{i-1}^u(\tau) \geq s_{i-1}^*$ for all $i \geq 1$, and for all $\tau \geq 0$, which also implies that $P_0(s^u(\tau)) \geq P_0^*$ for all $\tau \geq 0$. Therefore, the two terms inside brackets are nonnegative. Using the facts $\lambda < 1$, $s_{i-1}^* \leq 1$, and $P_0(s^u(\tau)) \leq 1$, Equation (3.15) implies that

$$-v_i^u(0) \leq \int_0^t \Big(\left[s_{i-1}^u(\tau) - s_{i-1}^*\right] + \left[P_0(s^u(\tau)) - P_0^*\right] - \left[s_i^u(\tau) - s_i^*\right] \Big) d\tau,$$

or

$$\int_0^t \left(s_i^u(\tau) - s_i^*\right) d\tau \leq v_i(0) + \int_0^t \left(s_{i-1}^u(\tau) - s_{i-1}^*\right) d\tau + \int_0^t \Big(P_0(s^u(\tau)) - P_0^*\Big) d\tau. \tag{3.16}$$

The first integral on the right-hand side of Equation (3.16) is upper-bounded uniformly in t, by the induction hypothesis (Equation (3.13)). We now derive an upper bound on the last integral, for each one of the three regimes.

(i) **High Memory regime**: By inspecting the expression for $P_0(s)$ for the High-

Memory variant, we observe that it is monotonically nondecreasing and Lipschitz continuous in s_1. Therefore, there exists a constant L such that

$$\int_0^t \big(P_0(s^u(\tau)) - P_0^*\big)d\tau \leq \int_0^t L\big(s_1^u(\tau) - s_1^*\big)d\tau.$$

Using the induction hypothesis for $k = 1$, we conclude that the last integral on the right-hand side of Equation (3.16) is upper bounded, uniformly in t.

(ii) **Constrained regime**: For the Constrained regime, the function $P_0(s)$ is again monotonically nondecreasing and, as remarked at the beginning of the proof of Lemma 3.2.1, it is also Lipschitz continuous in s_1. Thus, the argument is identical to the previous case.

(iii) **High Message regime**: We have an initial condition $s^0 \in \mathcal{S}^1$, and therefore $0 \leq v_1^0 < \infty$. As already remarked, we have $\dot{v}_1^u(t) = \lambda - s_1^u(t) \leq 0$. It follows that s_1^u can be equal to 1 for at most $v_1^0/(1-\lambda)$ units of time. Therefore, $P_0(s^u(t)) = [1 - (1 - s_2^u(t))/\lambda]^+ \mathbb{1}_{\{1\}}\big(s_1^u(t)\big)$ can be positive only on a set of times of Lebesgue measure at most $v_1^0/(1-\lambda)$. This implies the uniform (in t) upper bound

$$\int_0^t \Big(P_0(s^u(\tau)) - P_0^*\Big)d\tau = \int_0^t P_0(s^u(\tau))d\tau \leq \frac{v_1^0}{1-\lambda}.$$

For all three cases, we have shown that the last integral in Equation (3.16) is upper bounded, uniformly in t. It follows from Equation (3.16) and the induction hypothesis that

$$\int_0^\infty \left(s_i^u(\tau) - s_i^*\right) d\tau < \infty.$$

This completes the proof of the induction step. Using the Lipschitz-continuity of $s_i^u(\cdot)$, it follows that $s_i^u(t)$ converges to s_i^*, as $t \to \infty$, for all $i \geq 1$. It is straightforward to check that this coordinate-wise convergence, together with boundedness ($s_i^u(t) \leq 1$,

for all i and t), implies that also

$$\lim_{t\to\infty} \|s^u(t) - s^*\|_w = 0.$$

An analogous argument gives us the convergence

$$\lim_{t\to\infty} \|s^l(t) - s^*\|_w = 0,$$

which concludes the proof. □

3.3 Proof of Theorem 3.1.2 and of the rest of Theorem 3.1.1

We will now prove the convergence of the stochastic system to the fluid solution. The proof involves three steps. We first define the process using a coupled sample path approach, as in [42]. We then show the existence of limiting trajectories under the fluid scaling (Proposition 3.3.3). We finally show that any such limit trajectory must satisfy the differential equations in the definition of the fluid model (Proposition 3.3.4).

3.3.1 Probability space and coupling

We will first define a common probability space for all n. We will then define a coupled sequence of processes $\{(S^n(\cdot), M^n(\cdot))\}_{n=1}^{\infty}$. This approach will allow us to obtain almost sure convergence in the common probability space.

Fundamental processes and initial conditions

All processes of interest (for all n) will be driven by certain common fundamental processes.

a) **Driving Poisson processes:** Independent Poisson counting processes $\mathcal{N}_\lambda(\cdot)$ (process of arrivals, with rate λ), and $\mathcal{N}_1(\cdot)$ (process of potential departures,

with rate 1). A coupled sequence $\{\mathcal{N}_{\beta_n}(\cdot)\}_{n=1}^{\infty}$ (processes of potential messages, with nondecreasing rates β_n), independent of $\mathcal{N}_\lambda(\cdot)$ and $\mathcal{N}_1(\cdot)$, such that the events in $\mathcal{N}_{\beta_n}(\cdot)$ are a subset of the events in $\mathcal{N}_{\beta(n+1)}(\cdot)$ almost surely, for all $n \geq 1$. These processes are defined on a common probability space $(\Omega_D, \mathcal{A}_D, \mathbb{P}_D)$.

b) **Selection variables:** Three independent and individually i.i.d. sequences of random variables $\{U_k\}_{k=1}^{\infty}$, $\{V_k\}_{k=1}^{\infty}$, and $\{W_k\}_{k=1}^{\infty}$, uniform on $[0,1]$, defined on a common probability space $(\Omega_S, \mathcal{A}_S, \mathbb{P}_S)$.

c) **Initial conditions:** A sequence of random variables $\left\{\left(S^{(0,n)}, M^{(0,n)}\right)\right\}_{n=1}^{\infty}$ defined on a common probability space $(\Omega_0, \mathcal{A}_0, \mathbb{P}_0)$ and taking values in $(\mathcal{S}^1 \cap \mathcal{I}_n) \times \{0, 1, \ldots, c_n\}$.

The whole system will be defined on the probability space

$$(\Omega, \mathcal{A}, \mathbb{P}) = (\Omega_D \times \Omega_S \times \Omega_0, \mathcal{A}_D \times \mathcal{A}_S \times \mathcal{A}_0, \mathbb{P}_D \times \mathbb{P}_S \times \mathbb{P}_0).$$

All of the randomness in the system (for any n) will be specified by these fundamental processes, and everything else will be a deterministic function of them.

A coupled construction of sample paths

Recall that our policy results in a Markov process $\left(S^n(\cdot), M^n(\cdot)\right)$, taking values in the set $\left(\mathcal{S}^1 \cap \mathcal{I}_n\right) \times \{0, 1, \ldots, c_n\}$, where $S_i^n(t)$ is the fraction of servers with at least i jobs and $M^n(t)$ is the number of tokens stored in memory, at time t. We now describe a particular construction of the process, as a deterministic function of the fundamental processes. We decompose the process $S^n(\cdot)$ as the sum of two non-negative and non-decreasing processes, $A^n(\cdot)$ and $D^n(\cdot)$, that represent the (scaled by n) total cumulative arrivals to and departures from the queues, respectively, so that

$$S^n(\cdot) = S^{(0,n)} + A^n(\cdot) - D^n(\cdot).$$

Let $t_j^{\lambda,n}$, $t_j^{1,n}$, and $t_j^{\beta,n}$ be the time of the j-th arrival of the processes $\mathcal{N}_\lambda(n\,\cdot)$, $\mathcal{N}_1(n\,\cdot)$, and $\mathcal{N}_{\beta_n}(n\,\cdot)$, respectively. In order to simplify notation, we will omit the superscripts

λ, 1, and β, when the corresponding process is clear. For every $t \geq 1$, the first component of $A^n(t)$ is

$$A_1^n(t) \triangleq \frac{1}{n} \sum_{j=1}^{\mathcal{N}_\lambda(nt)} \left[\mathbb{1}_{[1,c_n]} \left(M^n \left(t_j^{n-} \right) \right) + \mathbb{1}_{\{0\}} \left(M^n \left(t_j^{n-} \right) \right) \mathbb{1}_{\left[0, 1 - S_1^n \left(t_j^{n-} \right) \right)} (U_j) \right]. \quad (3.17)$$

The above expression is interpreted as follows. We have an upward jump of size $1/n$ in $A_1^n(\cdot)$ every time that a job joins an empty queue, which happens every time that there is an arrival and either (i) there are tokens in the virtual queue (i.e., $M^n > 0$) or, (ii) there are no tokens and an empty queue is drawn uniformly at random, which happens with probability $1 - S_1^n$. Similarly, for $i \geq 2$,

$$A_i^n(t) \triangleq \frac{1}{n} \sum_{j=1}^{\mathcal{N}_\lambda(nt)} \mathbb{1}_{\{0\}} \left(M^n \left(t_j^{n-} \right) \right) \mathbb{1}_{\left[1 - S_{i-1}^n \left(t_j^{n-} \right), 1 - S_i^n \left(t_j^{n-} \right) \right)} (U_j).$$

In this case we have an upward jump in $A_i^n(\cdot)$ of size $1/n$ every time that there is an arrival, there are no tokens in the virtual queue (i.e., $M^n = 0$), and a queue with exactly $i - 1$ jobs is drawn uniformly at random, which happens with probability $S_{i-1}^n - S_i^n$. Moreover, for all $i \geq 1$,

$$D_i^n(t) \triangleq \frac{1}{n} \sum_{j=1}^{\mathcal{N}_1(nt)} \mathbb{1}_{\left[1 - S_i^n \left(t_j^{n-} \right), 1 - S_{i+1}^n \left(t_j^{n-} \right) \right)} (W_j).$$

We have an upward jump in $D_i^n(\cdot)$ of size $1/n$ when there is a departure from a queue with exactly i jobs, which happens with rate $\left(S_i^n - S_{i+1}^n \right) n$.

Recall that β_n is the message rate of an empty server. In the High Memory and Constrained regimes, we have $\beta_n = \beta$, while in the High Message regime β_n is a nondecreasing and unbounded sequence. Potential messages are generated according to the process $\mathcal{N}_{\beta_n}(n \cdot)$, but an actual message is generated only if a randomly selected

queue is empty. Thus, the number of tokens in the virtual queue evolves as follows:

$$M^n(t) \triangleq M^{(0,n)} - \sum_{j=1}^{\mathcal{N}_\lambda(nt)} \mathbb{1}_{[1,c_n]}\left(M^n\left(t_j^{n-}\right)\right) + \sum_{j=1}^{\mathcal{N}_{\beta_n}(nt)} \mathbb{1}_{[0,c_n-1]}\left(M^n\left(t_j^{n-}\right)\right) \mathbb{1}_{\left[0,1-S_1^n\left(t_j^{n-}\right)-\frac{M^n\left(t_j^{n-}\right)}{n}\right)}(V_j). \tag{3.18}$$

To see this, if the virtual queue is not empty, a token is removed from the virtual queue each time there is an arrival. Furthermore, if the virtual queue is not full, a new token is added each time a new message arrives from one of the $n(1-S_1^n)-M^n$ queues that do not have corresponding tokens in the virtual queue.

As mentioned earlier, the proof involves the following two steps:

1. We show that there exists a measurable set $\mathcal{C} \subset \Omega$ with $\mathbb{P}(\mathcal{C}) = 1$ such that for all $\omega \in \mathcal{C}$, any sequence of sample paths $S^n(\omega, \cdot)$ contains a further subsequence that converges to a Lipschitz continuous trajectory $s(\cdot)$, as $n \to \infty$.

2. We characterize the derivative of $s(\cdot)$ at any regular point and show that it is identical to the drift of our fluid model. Hence $s(\cdot)$ must be a fluid solution for some initial condition s^0, yielding also, as a corollary, the existence of fluid solutions.

3.3.2 Tightness of sample paths

We start by finding a set of "nice" sample paths ω for which any subsequence of the sequence $\{S^n(\omega, \cdot)\}_{n=1}^{\infty}$ contains a further subsequence $\{S^{n_k}(\omega, \cdot)\}_{k=1}^{\infty}$ that converges to some Lipschitz continuous function $s(\cdot)$. The arguments involved here are fairly straightforward and routine.

Lemma 3.3.1. *Fix $T > 0$. There exists a measurable set $\mathcal{C} \subset \Omega$ such that $\mathbb{P}(\mathcal{C}) = 1$*

and for all $\omega \in \mathcal{C}$,

$$\lim_{n\to\infty} \sup_{t\in[0,T]} \left| \frac{1}{n}\mathcal{N}_\lambda(\omega, nt) - \lambda t \right| = 0, \tag{3.19}$$

$$\lim_{n\to\infty} \sup_{t\in[0,T]} \left| \frac{1}{n}\mathcal{N}_1(\omega, nt) - t \right| = 0, \tag{3.20}$$

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[a,b)}(U_i(\omega)) = b - a, \quad \textit{for all } [a,b) \subset [0,1], \tag{3.21}$$

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[c,d)}(W_i(\omega)) = d - c, \quad \textit{for all } [c,d) \subset [0,1]. \tag{3.22}$$

Proof. Using the Functional Strong Law of Large Numbers for Poisson processes, we obtain a subset $\mathcal{C}_D \subset \Omega_D$ such that $\mathbb{P}_D(\mathcal{C}_D) = 1$ on which Equations (3.19) and (3.20) hold. Furthermore, the Glivenko-Cantelli lemma gives us another subset $\mathcal{C}_S \subset \Omega_S$ such that $\mathbb{P}_S(\mathcal{C}_S) = 1$ and on which equations (3.21) and (3.22) hold. Taking $\mathcal{C} = \mathcal{C}_D \times \mathcal{C}_S \times \Omega_0$ concludes the proof. □

Let us fix an arbitrary $s^0 \in [0,1]$, sequences $R_n \downarrow 0$ and $\gamma_n \downarrow 0$, and a constant $L > 0$. For $n \geq 1$, we define the following subsets of $D[0,T]$:

$$E_n(R_n, \gamma_n) \triangleq \Big\{ s \in D[0,T] : |s(0) - s^0| \leq R_n, \text{ and } \\ |s(a) - s(b)| \leq L|a-b| + \gamma_n, \ \forall\, a, b \in [0,T] \Big\}. \tag{3.23}$$

We also define

$$E_c \triangleq \Big\{ s \in D[0,T] : \ s(0) = s^0, \ |s(a) - s(b)| \leq L|a-b|, \ \forall\, a, b \in [0,T] \Big\},$$

which is the set of L-Lipschitz continuous functions with fixed initial conditions, and which is known to be sequentially compact, by the Arzelà-Ascoli theorem.

Lemma 3.3.2. *Fix* $T > 0$, $\omega \in \mathcal{C}$, *and some* $s^0 \in \mathcal{S}^1$. *Suppose that*

$$\left\| S^n(\omega, 0) - s^0 \right\|_w \leq \tilde{R}_n,$$

for some sequence $\tilde{R}_n \downarrow 0$. *Then, there exist sequences* $\left\{R_n^{(i)} \downarrow 0\right\}_{i=0}^{\infty}$ *and* $\gamma_n \downarrow 0$ *such that*

$$S_i^n(\omega, \cdot) \in E_n\left(R_n^{(i)}, \gamma_n\right), \quad \forall\, i \in \mathbb{Z}_+,\ \forall\ n \geq 1,$$

with the constant L *in the definition of* E_n *equal to* $1 + \lambda$.

Proof. Fix some $\omega \in \mathcal{C}$. Based on our coupled construction, each coordinate of $A^n(\cdot)$ (the process of cumulative arrivals) and $D^n(\cdot)$ (the process of cumulative departures) is non-decreasing, and can have a positive jump, of size $1/n$, only when there is an event in $\mathcal{N}_\lambda(n\,\cdot)$ or $\mathcal{N}_1(n\,\cdot)$, respectively. As a result, for every i and n, we have

$$|A_i^n(\omega, a) - A_i^n(\omega, b)| \leq \frac{1}{n}\left|\mathcal{N}_\lambda(\omega, na) - \mathcal{N}_\lambda(\omega, nb)\right|, \quad \forall\, a, b \in [0, T],$$

and

$$|D_i^n(\omega, a) - D_i^n(\omega, b)| \leq \frac{1}{n}\left|\mathcal{N}_1(\omega, na) - \mathcal{N}_1(\omega, nb)\right|, \quad \forall\, a, b \in [0, T].$$

Therefore,

$$|S_i^n(\omega, a) - S_i^n(\omega, b)| \leq \frac{1}{n}\left|\mathcal{N}_\lambda(\omega, na) - \mathcal{N}_\lambda(\omega, nb)\right| + \frac{1}{n}\left|\mathcal{N}_1(\omega, na) - \mathcal{N}_1(\omega, nb)\right|.$$

Since $\omega \in \mathcal{C}$, Lemma 3.3.1 implies that $\frac{1}{n}\mathcal{N}_\lambda(\omega, nt)$ and $\frac{1}{n}\mathcal{N}_1(\omega, nt)$ converge uniformly on $[0, T]$ to λt and to t, respectively. Thus, there exists a pair of sequences $\gamma_n^1 \downarrow 0$ and $\gamma_n^2 \downarrow 0$ (which depend on ω) such that for all $n \geq 1$,

$$\frac{1}{n}|\mathcal{N}_\lambda(\omega, na) - \mathcal{N}_\lambda(\omega, nb)| \leq \lambda|a - b| + \gamma_n^1,$$

and

$$\frac{1}{n}|\mathcal{N}_1(\omega, na) - \mathcal{N}_1(\omega, nb)| \leq |a - b| + \gamma_n^2,$$

which imply that

$$|S_i^n(\omega, a) - S_i^n(\omega, b)| \leq (1 + \lambda)|a - b| + (\gamma_n^1 + \gamma_n^2).$$

The proof is completed by setting $R_n^{(i)} = 2^i \tilde{R}_n$, $\gamma_n = \gamma_n^1 + \gamma_n^2$, and $L = 1 + \lambda$. $\square$

We are now ready to prove the existence of convergent subsequences of the process of interest.

Proposition 3.3.3. *Fix $T > 0$, $\omega \in \mathcal{C}$, and some $s^0 \in \mathcal{S}^1$. Suppose (as in Lemma 3.3.2) that $\|S^n(\omega, 0) - s^0\|_w \leq \tilde{R}_n$, where $\tilde{R}_n \downarrow 0$. Then, every subsequence of $\{S^n(\omega, \cdot)\}_{n=1}^{\infty}$ contains a further subsequence $\{S^{n_k}(\omega, \cdot)\}_{k=1}^{\infty}$ which converges to a coordinate-wise Lipschitz continuous function $s(\cdot)$ with $s(0) = s^0$ and*

$$|s_i(a) - s_i(b)| \leq L|a - b|, \qquad \forall\, a, b \in [0, T], i \in \mathbb{Z},$$

where L is independent of T, ω, and $s(\cdot)$.

Proof. As in Lemma 3.3.2, let $L = 1 + \lambda$. A standard argument, similar to the one in [12] and [42], based on the sequential compactness of E_c and the "closeness" of $E_n\big(R_n^{(i)}, \gamma_n\big)$ to E_c establishes the following. For any $i \geq 1$, every subsequence of $\{S_i^n(\omega, \cdot)\}_{n=1}^{\infty}$ contains a further subsequence that converges to a Lipschitz continuous function $y_i(\cdot)$ with $y_i(0) = s_i^0$.

Starting with the existence of coordinate-wise limit points, we now argue the existence of a limit point of $S^n(\cdot)$ in $D^{\infty}[0, T]$. Let $s_1(\cdot)$ be a Lipschitz continuous limit point of $\{S_1^n(\omega, \cdot)\}_{n=1}^{\infty}$, so that there is a subsequence such that

$$\lim_{k \to \infty} d\left(S_1^{n_k^1}(\omega, \cdot), s_1(\cdot)\right) = 0.$$

We then proceed inductively and let $s_{i+1}(\cdot)$ be a limit point of a further subsequence of $\left\{S_{i+1}^{n_k^i}(\omega, \cdot)\right\}_{k=1}^{\infty}$, where $\{n_k^i\}_{k=1}^{\infty}$ are the indices of the subsequence of $S_i^n(\cdot)$.

We now argue that $s(\cdot)$ is indeed a limit point of $S^n(\cdot)$ in $D^{\infty}[0, T]$. Fix a positive integer i. Because of the construction of $s(\cdot)$, $S_j^{n_k^i}(\omega, \cdot)$ converges to $s_j(\cdot)$, as $k \to \infty$, for $j = 1, \ldots, i$. In particular, there exists some $n^i > i$, for which

$$d\big(S_j^{n^i}(\omega, \cdot), s_j(\cdot)\big) \leq \frac{1}{i}, \qquad j = 1, \ldots, i.$$

We then have

$$d^{\mathbb{Z}_+}\left(S^{n^i}(\omega,\cdot),s(\cdot)\right) = \sup_{t\in[0,T]}\sqrt{\sum_{j=1}^{\infty}2^{-j}\left|S_j^{n^i}(\omega,t)-s_j(t)\right|^2}$$
$$\leq \frac{1}{i}+\sqrt{\sum_{j=n^i+1}^{\infty}2^{-j+2}}.$$

We now let i increase to infinity (in which case n^i also increases to infinity), and we conclude that $d^{\mathbb{Z}_+}\left(S^{n^i}(\omega,\cdot),s(\cdot)\right)\to 0$.

□

This concludes the proof of the tightness of the sample paths. It remains to prove that any possible limit point is a fluid solution.

3.3.3 Derivatives of the fluid limits

Proposition 3.3.4. *Fix $\omega\in\mathcal{C}$ and $T>0$. Let $s(\cdot)$ be a limit point of some subsequence of $\{S^n(\omega,\cdot)\}_{n=1}^{\infty}$. As long as ω does not belong to a certain zero-measure subset of $\mathcal{C}$, $s(\cdot)$ satisfies the differential equations that define a fluid solution (cf. Definition 3.1.1).*

Proof. We fix some $\omega\in\mathcal{C}$ and for the rest of this proof we suppress the dependence on ω in our notation. Let $\{S^{n_k}(\cdot)\}_{k=1}^{\infty}$ be a subsequence that converges to $s(\cdot)$, i.e.,

$$\lim_{k\to\infty}\sup_{0\leq t\leq T}\left\|S^{n_k}(t)-s(t)\right\|_w=0.$$

After possibly restricting, if necessary, to a further subsequence, we can define Lipschitz continuous functions $a_i(\cdot)$ and $d_i(\cdot)$ as the limits of the subsequences of cumulative arrivals and departures processes $\{A_i^{n_k}(\cdot)\}_{k=1}^{\infty}$ and $\{D_i^{n_k}(\cdot)\}_{k=1}^{\infty}$ respectively. Because of the relation $S_i^n(\cdot)=S^{(0,n)}+A_i^n(\cdot)-D_i^n(\cdot)$, it is enough to prove the

following relations, for almost all t:

$$\begin{aligned}
\frac{da_1}{dt}(t) =& \lambda[1 - P_0(s(t))] + \lambda[1 - s_1(t)]P_0(s(t)), \\
\frac{da_i}{dt}(t) =& \lambda[s_{i-1}(t) - s_i(t)]P_0(s(t)), \qquad \forall\, i \geq 2, \\
\frac{dd_i}{dt}(t) =& s_i(t) - s_{i+1}(t), \qquad \forall\, i \geq 1.
\end{aligned}$$

We will provide a proof only for the first one, as the other proofs are similar. The main idea in the argument that follows is to replace the token process $M^n(\cdot)$ by simpler, time-homogeneous birth-death processes that are easy to analyze.

Let us fix some time $t \in (0, T)$, which is a regular time for both $a_1(\cdot)$ and $d_1(\cdot)$. Let $\epsilon > 0$ be small enough so that $t + \epsilon \leq T$ and so that it also satisfies a condition to be introduced later. Equation (3.17) yields

$$\begin{aligned}
A_1^{n_k}(t+\epsilon) - A_1^{n_k}(t) =& \frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \Big[\mathbb{1}_{[1,c_{n_k}]} \left(M^{n_k}\left(t_j^{n_k -}\right)\right) \\
& + \mathbb{1}_{\{0\}} \left(M^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{\left[0, 1 - S_1^{n_k}\left(t_j^{n_k -}\right)\right)}(U_j) \Big]. \qquad (3.24)
\end{aligned}$$

By Lemma 3.3.2, there exists a sequence $\gamma_{n_k} \downarrow 0$ and a constant L such that

$$S_1^{n_k}(u) \in \left[s_1(t) - (\epsilon L + \gamma_{n_k}),\ s_1(t) + (\epsilon L + \gamma_{n_k}) \right), \quad \forall\, u \in [t, t+\epsilon].$$

Then, for all sufficiently large k, we have

$$S_1^{n_k}(u) \in \left[s_1(t) - 2\epsilon L,\ s_1(t) + 2\epsilon L \right), \quad \forall\, u \in [t, t+\epsilon]. \qquad (3.25)$$

In particular, for k sufficiently large and for every event time $t_j^{n_k -} \in (t, t+\epsilon]$ of the driving process $\mathcal{N}_\lambda(n\,\cdot)$, we have

$$\left[0, 1 - S_1^{n_k}\left(t_j^{n_k -}\right) \right) \subset \Big[0, 1 - s_1(t) + 2\epsilon L \Big).$$

This implies that

$$A_1^{n_k}(t+\epsilon) - A_1^{n_k}(t) \leq \frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \Big[\mathbb{1}_{[1,c_{n_k}]} \left(M^{n_k}\left(t_j^{n_k -}\right)\right) + \mathbb{1}_{\{0\}} \left(M^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(U_j) \Big].$$

We wish to analyze this upper bound on $A_1^{n_k}(t+\epsilon) - A_1^{n_k}(t)$, which will then lead to an upper bound on $(da_i/dt)(t)$. Towards this purpose, we will focus on the empirical distribution of $\mathbb{1}_{\{0\}} \left(M^{n_k}\left(t_j^{n_k -}\right)\right)$, which depends on the birth-death process $M^{n_k}(\cdot)$, and which is in turn modulated by $S^{n_k}(\cdot)$. In particular, we will define two coupled time-homogeneous birth-death processes: $M_+^{n_k}(\cdot)$, which is dominated by $M^{n_k}(\cdot)$; and $M_-^{n_k}(\cdot)$, which dominates $M^{n_k}(\cdot)$ over $(t, t+\epsilon]$, i.e.,

$$M_+^{n_k}(u) \leq M^{n_k}(u) \leq M_-^{n_k}(u), \qquad \forall\, u \in (t, t+\epsilon]. \tag{3.26}$$

This is accomplished as follows. Using again Equation (3.25), when n_k is sufficiently large, we get the set inclusion

$$\left[0, 1 - S_1^{n_k}\left(t_j^{n_k -}\right) - \frac{M^{n_k}\left(t_j^{n_k -}\right)}{n_k} \right) \subset \Big[0, 1 - s_1(t) + 2\epsilon L \Big),$$

for all event times $t_j^{n_k} \in [t, t+\epsilon)$. Furthermore, our assumptions on c_{n_k} imply that $M^{n_k}(t)/n_k \leq c_{n_k}/n_k$ goes to zero as $k \to \infty$. Thus, when n_k is sufficiently large,

$$\left[0, 1 - S_1^{n_k}\left(t_j^{n_k -}\right) - \frac{M^{n_k}\left(t_j^{n_k -}\right)}{n_k} \right) \supset \Big[0, 1 - s_1(t) - 3\epsilon L \Big),$$

for all event times $t_j^{n_k} \in [t, t+\epsilon)$. We now define intermediate coupled processes $\tilde{M}_+^{n_k}(\cdot)$ and $\tilde{M}_-^{n_k}(\cdot)$ by replacing the last indicator set in the evolution equation for the process $M^n(\cdot)$ (cf. Equation (3.18)), by the deterministic sets introduced above. Furthermore, we set $\tilde{M}_+^{n_k}(t) = 0 \leq M^{n_k}(t)$ and $\tilde{M}_-^{n_k}(t) = c_{n_k} \geq M^{n_k}(t)$.

More concretely, for all $u \in [t, t+\epsilon]$, we let

$$\tilde{M}_-^{n_k}(u) \triangleq c_{n_k} - \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k u)} \mathbb{1}_{[1,c_{n_k}]}\left(\tilde{M}_-^{n_k}\left(t_j^{n_k-}\right)\right) + \sum_{j=\mathcal{N}_{\beta_{n_k}}(n_k t)+1}^{\mathcal{N}_{\beta_{n_k}}(n_k u)} \mathbb{1}_{[0,c_{n_k}-1]}\left(\tilde{M}_-^{n_k}\left(t_j^{n_k-}\right)\right)\mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(V_j)$$

and

$$\tilde{M}_+^{n_k}(u) \triangleq 0 - \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k u)} \mathbb{1}_{[1,c_{n_k}]}\left(\tilde{M}_+^{n_k}\left(t_j^{n_k-}\right)\right) + \sum_{j=\mathcal{N}_{\beta_{n_k}}(n_k t)+1}^{\mathcal{N}_{\beta_{n_k}}(n_k u)} \mathbb{1}_{[0,c_{n_k}-1]}\left(\tilde{M}_+^{n_k}\left(t_j^{n_k-}\right)\right)\mathbb{1}_{[0,1-s_1(t)-3\epsilon L)}(V_j).$$

We note that the processes $\tilde{M}_-^{n_k}(\cdot)$ and $\tilde{M}_+^{n_k}(\cdot)$ are plain, time-homogenous birth-death Markov processes, no longer modulated by $S^{n_k}(\cdot)$, and therefore easy to analyze. It can now be argued, by induction on the event times, that $\tilde{M}_-^{n_k}(u) \geq M^{n_k}(u)$ for all $u \in [t, t+\epsilon]$. We omit the details but simply note that (i) this inequality holds at time t; (ii) whenever the process $M^{n_k}(\cdot)$ has an upward jump, the same is true for $\tilde{M}_-^{n_k}(\cdot)$, unless $\tilde{M}_-^{n_k}(\cdot)$ is already at its largest possible value, c_{n_k}, in which case the desired inequality is preserved; (iii) as long as the desired inequality holds, whenever the process $\tilde{M}_-^{n_k}(\cdot)$ has a downward jump, the same is true for $M^{n_k}(\cdot)$, unless $M^{n_k}(\cdot)$ is already at its smallest possible value, 0, in which case the desired inequality is again preserved. Using also a symmetrical argument for $\tilde{M}_+^{n_k}(\cdot)$, we obtain the domination relationship

$$\tilde{M}_+^{n_k}(u) \leq M^{n_k}(u) \leq \tilde{M}_-^{n_k}(u), \quad \forall\, u \in (t, t+\epsilon]. \tag{3.27}$$

Even though $\tilde{M}_+^{n_k}(\cdot)$ and $\tilde{M}_-^{n_k}(\cdot)$ are simple birth-death processes, it is convenient to simplify them even further. We thus proceed to define the coupled processes $M_+^{n_k}(\cdot)$ and $M_-^{n_k}(\cdot)$ by modifying the intermediate processes $\tilde{M}_+^{n_k}(\cdot)$ and $\tilde{M}_-^{n_k}(\cdot)$ in a different way for each regime.

(i) **High Memory regime**: Recall that in this regime we have $\beta_{n_k} = \beta$ for all k. Let us fix some l, independently from k, and let $c_l = c(n_l)$. For every k, we define $M_+^{n_k}(\cdot)$ and $M_-^{n_k}(\cdot)$ by replacing the upper bound c_{n_k} on the number of tokens in $\tilde{M}_+^{n_k}(\cdot)$ and $\tilde{M}_-^{n_k}(\cdot)$, by c_l and ∞ respectively. More concretely, for $u \in [t, t+\epsilon]$ we let

$$M_-^{n_k}(u) \triangleq c_{n_k} - \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k u)} \mathbb{1}_{[1,\infty)}\left(M_-^{n_k}\left(t_j^{n_k -}\right)\right) + \sum_{j=\mathcal{N}_\beta(n_k t)+1}^{\mathcal{N}_\beta(n_k u)} \mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(V_j)$$

and

$$M_+^{n_k}(u) \triangleq 0 - \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k u)} \mathbb{1}_{[1,c_l]}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) + \sum_{j=\mathcal{N}_\beta(n_k t)+1}^{\mathcal{N}_\beta(n_k u)} \mathbb{1}_{[0,c_l-1]}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{[0,1-s_1(t)-3\epsilon L)}(V_j).$$

When k is large enough, we have $c_{n_k} \geq c_l$, and as we are replacing c_{n_k} by c_l in $\tilde{M}_+^{n_k}(u)$, we are reducing the state space of the homogeneous birth-death process $\tilde{M}_+^{n_k}(\cdot)$. It is easily checked (by induction on the events of the processes) that we have the stochastic dominance $\tilde{M}_+^{n_k}(u) \geq M_+^{n_k}(u)$, for all $u \in [t, t+\epsilon]$. Using a similar argument, we obtain $\tilde{M}_-^{n_k}(u) \leq M_-^{n_k}(u)$, for all $u \in [t, t+\epsilon]$. These facts, together with Equation (3.27), imply the desired dominance relation in Equation (3.26).

(ii) **High Message regime**: Recall that in this regime we have $c_{n_k} = c$, for all k. Let us fix some l, independently from k, and let $\beta_l = \beta(n_l)$. We define $M_+^{n_k}(\cdot)$ by replacing the process $\mathcal{N}_{\beta_{n_k}}(\cdot)$ that generates the spontaneous messages in

$\tilde{M}_+^{n_k}(\cdot)$, by $\mathcal{N}_{\beta_l}(\cdot)$. More concretely, for $u \in [t, t+\epsilon]$ we let

$$M_+^{n_k}(u) \triangleq 0 - \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k u)} \mathbb{1}_{[1,c]}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) + \sum_{j=\mathcal{N}_{\beta_l}(n_k t)+1}^{\mathcal{N}_{\beta_l}(n_k u)} \mathbb{1}_{[0,c-1]}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{[0,1-s_1(t)-3\epsilon L)}(V_j).$$

Recall that we assumed that the event times in the Poisson process $\mathcal{N}_{\beta_{n_k}}(\cdot)$ are a subset of the event times of $\mathcal{N}_{\beta(n_{k+1})}(\cdot)$, for all k. As a result, when $k \geq l$, the process $M_+^{n_k}(\cdot)$ only has a subset of the upward jumps in $\tilde{M}_+^{n_k}(\cdot)$, and thus (using again a simple inductive argument) satisfies $\tilde{M}_+^{n_k}(u) \geq M_+^{n_k}(u)$, for all $u \in [t, t+\epsilon]$. Furthermore, we define $M_-^{n_k}(u) \triangleq c$, for all $u \in [t, t+\epsilon]$, which clearly satisfies $\tilde{M}_-^{n_k}(u) \leq M_-^{n_k}(u)$, for all $u \in [t, t+\epsilon]$. Combining these facts with Equation (3.27), we have again the desired dominance relation in Equation (3.26).

(iii) **Constrained regime**: Recall that in this regime we have $c_{n_k} = c$ and $\beta_{n_k} = \beta$, for all $k \geq 1$. For this case, we define $M_-^{n_k}(u) \triangleq \tilde{M}_-^{n_k}(u)$ and $M_+^{n_k}(u) \triangleq \tilde{M}_+^{n_k}(u)$, for all $u \in [t, t+\epsilon]$, which already satisfy the desired dominance relation in Equation (3.26).

For all three regimes, and having fixed l, the dominance relation in Equation (3.26) implies that when k is large enough ($k \geq l$), we have

$$\mathbb{1}_{\{0\}}\left(M_-^{n_k}\left(t_j^{n_k -}\right)\right) \leq \mathbb{1}_{\{0\}}\left(M^{n_k}\left(t_j^{n_k -}\right)\right) \leq \mathbb{1}_{\{0\}}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right)$$

for all $t_j^{n_k -} \in (t, t+\epsilon]$. Consequently,

$$A_1^{n_k}(t+\epsilon) - A_1^{n_k}(t) \leq \frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \Big[1 - \mathbb{1}_{\{0\}}\left(M_-^{n_k}\left(t_j^{n_k -}\right)\right) + \mathbb{1}_{\{0\}}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(U_j)\Big]. \quad (3.28)$$

Note that the transition rates of the birth-death processes $M_-^{n_k}(\cdot)$ and $M_+^{n_k}(\cdot)$, for

different n_k, involve n_k only as a scaling factor. As a consequence, the corresponding steady-state distributions are the same for all n_k.

Let $P_0^-(s(t))$ and $P_0^+(s(t))$ be the steady-state probabilities of state 0 for $M_-^{n_k}(\cdot)$ and $M_+^{n_k}(\cdot)$, respectively. Then, using the PASTA property, we have that as $n_k \to \infty$, the empirical averages

$$\frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \mathbb{1}_{\{0\}} \left(M_-^{n_k} \left(t_j^{n_k -} \right) \right) \tag{3.29}$$

and

$$\frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \mathbb{1}_{\{0\}} \left(M_+^{n_k} \left(t_j^{n_k -} \right) \right)$$

converge almost surely to $\epsilon\lambda P_0^-(s(t))$ and $\epsilon\lambda P_0^+(s(t))$, respectively.

We now continue with the explicit calculation of $P_0^-(s(t))$ and $P_0^+(s(t))$.

(i) **High Memory regime**:

$$P_0^-(s(t)) = \left[1 - \frac{\beta \cdot \min\{1 - s_1(t) + 2\epsilon L, 1\}}{\lambda} \right]^+,$$

and

$$P_0^+(s(t)) = \left[\sum_{k=0}^{c_l} \left(\frac{\beta \left(1 - s_1(t) - 3\epsilon L\right)^+}{\lambda} \right)^k \right]^{-1},$$

(ii) **High Message regime**: If $s_1(t) < 1$, then we assume that ϵ has been chosen small enough so that $1 - s_1(t) - 3\epsilon L > 0$. We then obtain

$$P_0^-(s(t)) = 0$$

and

$$P_0^+(s(t)) = \left[\sum_{k=0}^{c} \left(\frac{\beta_l [1 - s_1(t) - 3\epsilon L]^+}{\lambda} \right)^k \right]^{-1}.$$

Suppose now that $s_1(t) = 1$. In this case, the approach based on the processes $M_-^{n_k}$ and $M_+^{n_k}$ is not useful, because it yields $P_0^-(s(t)) = 0$ and $P_0^+(s(t)) = 1$,

for all $\epsilon > 0$ and for all β_l. This case will be considered separately later.

(iii) **Constrained regime**:

$$P_0^-(s(t)) = \left[\sum_{k=0}^{c} \left(\frac{\beta \cdot \min\{1 - s_1(t) + 2\epsilon L, 1\}}{\lambda}\right)^k\right]^{-1}$$

and

$$P_0^+(s(t)) = \left[\sum_{k=0}^{c} \left(\frac{\beta\big(1 - s_1(t) - 3\epsilon L\big)^+}{\lambda}\right)^k\right]^{-1},$$

We now continue by considering all three regimes, with the exception of the High Message regime with $s_1(t) = 1$, which will be dealt with separately. We use the fact that the random variables U_j are independent from the process $M_+^{n_k}$. Using an elementary argument, which is omitted, it can be seen that

$$\frac{1}{n_k} \sum_{j=\mathcal{N}_\lambda(n_k t)+1}^{\mathcal{N}_\lambda(n_k(t+\epsilon))} \mathbb{1}_{\{0\}}\left(M_+^{n_k}\left(t_j^{n_k -}\right)\right) \mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(U_j)$$

converges to the limit of the empirical average in Equation (3.29), which is the product of $\epsilon\lambda P_0^+(s(t))$ times the expected value of $\mathbb{1}_{[0,1-s_1(t)+2\epsilon L)}(U_j)$. That is, it converges to $\epsilon P_0^+(s(t)) \min\{1 - s_1(t) + 2\epsilon L, 1\}$, $\mathbb{P}$-almost surely.

Recall that we have fixed some $\epsilon > 0$ and some l and, furthermore, that P_0^- and P_0^+ depend on l for the High Memory and High Message regimes, and on ϵ for all regimes. We will first take limits, as $k \to \infty$, while holding ϵ and l fixed. Using the inequality in Equation (3.28), and the fact that the left-hand side converges to the fluid limit $a(t + \epsilon) - a(t)$ as $k \to \infty$, we obtain

$$a_1(t + \epsilon) - a_1(t) \leq \epsilon\lambda[1 - P_0^-(s(t))] + \epsilon\lambda P_0^+(s(t)) \min\{1 - s_1(t) + 2\epsilon L, 1\}.$$

An analogous argument yields

$$a_1(t + \epsilon) - a_1(t) \geq \epsilon\lambda[1 - P_0^+(s(t))] + \epsilon\lambda P_0^-(s(t))[1 - s_1(t) - 2\epsilon L]^+.$$

We now take the limit as $l \to \infty$, so that $c_l \to \infty$ for the High Memory regime and $\beta_l \to \infty$ for the High Message regime, and then take the limit as $\epsilon \to 0$. Some elementary algebra shows that in all cases, $P_0^+(s(t))$ and $P_0^-(s(t))$ both converge to $P_0(s(t))$, as defined in the statement of the proposition. We thus obtain

$$\frac{da_1(t)}{dt} = \lambda[1 - P_0(s(t))] + \lambda[1 - s_1(t)]P_0(s(t)), \tag{3.30}$$

as desired.

We now return to the exceptional case of the High Message regime with $s_1(t) = 1$, and find the derivative of $a_1(t)$ using a different argument. Recall that we have the hard bound $S_1^n(t) \leq 1$, for all t and for all n. This leads to the same bound for the fluid solutions, i.e., $s_1(t) \leq 1$ for all t. As a result, since $t > 0$ is a regular time, we must have $\dot{s}_1(t) = 0$. Furthermore, we also have the formula

$$\frac{dd_1}{dt}(t) = s_1(t) - s_2(t) = 1 - s_2(t),$$

which is established by an independent argument, using the same proof technique as for $\dot{a}_1$, but without the inconvenience of having to deal with M^{n_k}. Then, since $\dot{s}_1(t) = \dot{a}_1(t) - \dot{d}_1(t)$, we must also have

$$\frac{da_1}{dt}(t) = 1 - s_2(t). \tag{3.31}$$

On the other hand, it can be easily checked that $\dot{a}_1(t) \leq \lambda$ for all regular t, and thus we must have $s_2(t) \geq 1 - \lambda$. We have thus established that at all regular times $t > 0$ with $s_1(t) = 1$, $s_2(t)$ must be at least $1 - \lambda$. Then it follows (cf. Definition 3.1.1) that at time t, we have

$$P_0(s(t)) = \left[1 - \frac{1 - s_2(t)}{\lambda}\right]^+ \mathbb{1}_{\{1\}}\big(s_1(t)\big) = 1 - \frac{1 - s_2(t)}{\lambda}.$$

It is then easily checked that Equation (3.31) is of the form

$$\frac{da_1}{dt}(t) = \lambda(1 - P_0(s(t))) + \lambda(1 - s_1(t))P_0(s(t)),$$

exactly as in Equation (3.30), where the last equality used the property $s_1(t) = 1$.

The derivatives of $a_i(\cdot)$, for $i > 1$, and of $d_i(\cdot)$, for $i \geq 1$, are obtained using similar arguments, which are omitted. □

For every sample path outside a zero-measure set, we have established the following. Proposition 3.3.3 implies the existence of limit points of the process $S^n(\cdot)$. Furthermore, according to Proposition 3.3.4 these limit points verify the differential equations of the fluid model. Since all stochastic trajectories $S^n(\cdot)$ take values in $\mathcal{S}$ (which is a closed set), their limits are functions taking values in $\mathcal{S}$ as well. We will now show that the limit $s(\cdot)$ actually takes values in the smaller set $\mathcal{S}^1$, which is a requirement in our definition of fluid solutions. Using the same argument as in the proof of Proposition 3.2.5, it can be shown that

$$\frac{d}{dt}\|s(t)\|_1 \leq \lambda,$$

for all regular times t. Since the trajectories $s(\cdot)$ are continuous with respect to our weighted norm $\|\cdot\|_w$, but not necessarily with respect to the 1-norm, it now remains to be checked that the 1-norm cannot become infinite at a non-regular time.

Suppose that t_1 is a non-regular time. Recall, from the proof of Proposition 3.2.5, that such a time may occur only once, and only in the High Message regime, if trajectory hits the set

$$D = \{s \in \mathcal{S} : s_1 = 1,\ s_2 > 1 - \lambda\}.$$

For all $t < t_1$, we have $P_0(s(t)) = 0$, and thus $\dot{s}_i(t) \leq 0$, for all $t < t_1$ and all $i \geq 2$. Combining this with the continuity of the coordinates, we obtain $s_i(t_1) \leq s_i(0)$, for

all $i \geq 2$. It follows that

$$\|s(t_1)\|_1 \leq 1 + s_1(t_1) + \sum_{i=2}^{\infty} s_i(0) \leq 2 + \|s(0)\|_1.$$

Combining this with the fact that $\|s(0)\|_1 < \infty$, we get that $\|s(t)\|_1 < \infty$, for all $t \geq 0$, and thus $s(t) \in \mathcal{S}^1$, for all $t \geq 0$. This implies the existence of fluid solutions, thus completing the proof of Theorem 3.1.1.

Moreover, we have already established a uniqueness result in Theorem 3.1.1: for any initial condition $s^0 \in \mathcal{S}^1$, we have at most one fluid solution. We also have (Proposition 3.3.3) that every subsequence of $S^n(\cdot)$ has a further subsequence that converges — by necessity to the same (unique) fluid solution. It can be seen that this implies the convergence of $S^n(\cdot)$ to the fluid solution, thus proving Theorem 3.1.2.

3.4 Proofs of Proposition 3.1.3 and Theorem 3.1.4

In this section, we prove Proposition 3.1.3 and Theorem 3.1.4, which assert that for any finite n, the stochastic system is positive recurrent with some invariant distribution π^n and that the sequence of the marginals of the invariant distributions, $\{\pi_s^n\}_{n=1}^{\infty}$, converges in distribution to a measure concentrated on the unique equilibrium of the fluid model. These results guarantee that the properties derived from the equilibrium s^* of the fluid model, and specifically for the asymptotic queueing delay, are an accurate approximation of the steady state of the stochastic system for n large enough.

3.4.1 Stochastic stability of the n-th system

We will use the Foster-Lyapunov criterion to show that for any fixed n, the continuous-time Markov process $\big(S^n(\cdot), M^n(\cdot)\big)$ is positive recurrent.

Our argument is developed by first considering a detailed description of the system:

$$\Big(\mathbf{Q}_1^n(\cdot), \ldots, \mathbf{Q}_n^n(\cdot), M^n(\cdot)\Big),$$

which keeps track of the size of each queue, but without keeping track of the identities of the servers with associated tokens in the virtual queue. As hinted in Subsection 3.1.3, this is a continuous-time Markov process, on the state space

$$Z_n \triangleq \left\{ (\mathbf{q}_1, \ldots, \mathbf{q}_n, m) \in \mathbb{Z}_+^n \times \{0, 1, \ldots, c_n\} : \sum_{i=1}^{n} \mathbb{1}_{\{0\}}(\mathbf{q}_i) \geq m \right\}.$$

The transition rates, denoted by $r^n_{\cdot \to \cdot}$ are as follows, where we use $\mathbf{e}_i$ to denote the i-th unit vector in $\mathbb{Z}_+^n$.

1. When there are no tokens available ($m = 0$), each queue sees arrivals with rate λ:

$$r^n_{(\mathbf{q},0)\to(\mathbf{q}+\mathbf{e}_i,0)} = \lambda, \qquad i = 1, \ldots, n.$$

2. When there are tokens available ($m > 0$), the arrival stream, which has rate $n\lambda$, is divided equally between all empty queues:

$$r^n_{(\mathbf{q},m)\to(\mathbf{q}+\mathbf{e}_i,m-1)} = \frac{n\lambda \mathbb{1}_{\{0\}}(\mathbf{q}_i)}{\sum_{j=1}^{n} \mathbb{1}_{\{0\}}(\mathbf{q}_j)} \mathbb{1}_{[1,c_n]}(m), \quad i = 1, \ldots, n.$$

3. Transitions due to service completions occur at a uniform rate of 1 at each queue, and they do not affect the token queue:

$$r^n_{(\mathbf{q},m)\to(\mathbf{q}-\mathbf{e}_i,m)} = \mathbb{1}_{[1,\infty)}(\mathbf{q}_i), \qquad i = 1, \ldots, n.$$

4. Messages from idling servers are sent to the dispatcher (hence resulting in additional tokens) at a rate equal to β_n times the number of empty servers that do not already have associated tokens in the virtual queue:

$$r^n_{(\mathbf{q},m)\to(\mathbf{q},m+1)} = \beta_n \left(\sum_{i=1}^{n} \mathbb{1}_{\{0\}}(\mathbf{q}_i) - m \right) \mathbb{1}_{[0,c_n-1]}(m).$$

Note that, for all t, the state at time t of the Markov process of interest, $(S^n(t), M^n(t))$, is a deterministic function of $\left(\mathbf{Q}^n(t), M^n(t)\right)$. Therefore, to establish positive recur-

rence of the process $(S^n(\cdot), M^n(\cdot))$, it suffices to establish positive recurrence of the process $(\mathbf{Q}^n(\cdot), M^n(\cdot))$, as in the proof that follows.

Proof of Proposition 3.1.3. The Markov process $\big(\mathbf{Q}^n(\cdot), M^n(\cdot)\big)$ on the state space Z_n is clearly irreducible, with all states reachable from each other. To show positive recurrence, we define the quadratic Lyapunov function

$$\Phi(\mathbf{q}, m) \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbf{q}_i^2, \tag{3.32}$$

and note that

$$\sum_{(\mathbf{q}', m') \neq (\mathbf{q}, m)} \Phi(\mathbf{q}', m') r^n_{(\mathbf{q},m)\to(\mathbf{q}',m')} < \infty, \qquad \forall\, (\mathbf{q}, m) \in Z_n.$$

We also define the finite set

$$F_n \triangleq \left\{ (\mathbf{q}, m) \in Z_n : \sum_{i=1}^{n} \mathbf{q}_i < \frac{n(\lambda+2)}{2(1-\lambda)} \right\}.$$

As $\mathbf{q}_i$ can change but at most 1 during a transition, we use the relations $(\mathbf{q}_i+1)^2 - \mathbf{q}_i^2 = 2\mathbf{q}_i + 1$ and $(\mathbf{q}_i - 1)^2 - \mathbf{q}_i^2 = -2\mathbf{q}_i + 1$. For any $(\mathbf{q}, m)$ outside the set F_n, we have

$$\begin{aligned}
&\sum_{(\mathbf{q}', m') \in Z_n} \left[\Phi(\mathbf{q}', m') - \Phi(\mathbf{q}, m)\right] r^n_{(\mathbf{q},m)\to(\mathbf{q}',m')} \\
&\quad = \frac{1}{n} \sum_{i=1}^{n} \left[(2\mathbf{q}_i + 1)\lambda \left(\frac{n \mathbb{1}_{\{0\}}(\mathbf{q}_i)}{\sum_{j=1}^{n} \mathbb{1}_{\{0\}}(\mathbf{q}_j)} \mathbb{1}_{[1,c_n]}(m) + \mathbb{1}_{\{0\}}(m) \right) - (2\mathbf{q}_i - 1)\mathbb{1}_{[1,\infty)}(\mathbf{q}_i) \right] \\
&\quad = \lambda + \frac{1}{n} \sum_{i=1}^{n} \left[\mathbb{1}_{[1,\infty)}(\mathbf{q}_i) - 2\mathbf{q}_i \left(1 - \lambda \mathbb{1}_{\{0\}}(m)\right) \right] \\
&\quad \leq \lambda + 1 - \frac{2(1-\lambda)}{n} \sum_{i=1}^{n} \mathbf{q}_i \leq -1, \quad \forall\, (\mathbf{q}, m) \in Z_n \backslash F_n.
\end{aligned}$$

The last equality is obtained through a careful rearrangement of terms; the first inequality is obtained by replacing each indicator function by unity. Then, the Foster-Lyapunov criterion [18] implies positive recurrence. □

3.4.2 Convergence of the invariant distributions

As a first step towards establishing the interchange of limits result, we start by establishing some tightness properties, in the form of uniform (over all n) upper bounds for $\mathbb{E}_{\pi^n}\left[\|S^n\|_1\right]$ and for $\pi^n\left(\mathbf{Q}_1^n \geq k\right)$. One possible approach to obtaining such bounds is to use an appropriate coupling and show that our system is stochastically dominated by a system consisting of n independent parallel M/M/1 queues. However, we follow an easier approach based on a simple linear Lyapunov function and the results in [23] and [10].

Lemma 3.4.1. *Let π^n be the unique invariant distribution of the process $(\mathbf{Q}^n(\cdot), M^n(\cdot))$. We then have the uniform upper bounds*

$$\pi^n\left(\mathbf{Q}_1^n \geq k\right) \leq \left(\frac{1}{2-\lambda}\right)^{k/2}, \quad \forall\, n,\ \forall\, k,$$

and

$$\mathbb{E}_{\pi^n}\left[\|S^n\|_1\right] \leq 2 + \frac{2}{1-\lambda}, \qquad \forall\, n.$$

Proof. Consider the linear Lyapunov function

$$\Psi(\mathbf{q}, m) \triangleq \mathbf{q}_1.$$

Under the terminology in [10], this Lyapunov function has exception parameter $B = 1$, drift $\gamma = 1-\lambda$, maximum jump $\nu_{\max} = 1$, and maximum rate $p_{\max} \leq 1$. Note that this function is not a witness of stability because the set $\{(\mathbf{q}, m) \in Z_n : \Psi(\mathbf{q}, m) < 1\}$ is not finite. However, the boundedness of the upward jumps allows us to use Theorem 2.3 from [23] to obtain that $\mathbb{E}_{\pi^n}\left[\mathbf{Q}_1^n\right] < \infty$. Thus, all conditions in Theorem 1 in [10] are satisfied, yielding the upper bounds

$$\pi^n\left(\mathbf{Q}_1^n \geq 1 + 2m\right) \leq \left(\frac{1}{2-\lambda}\right)^{m+1}, \qquad \forall\ m \geq 1,$$

and

$$\mathbb{E}_{\pi^n}\left[\mathbf{Q}_1^n\right] \leq 1 + \frac{2}{1-\lambda}.$$

The first part of the result is obtained by letting $m = (k-1)/2$ if k is odd or $m = k/2 - 1$ if k is even. Finally, using the definition $\|S^n\|_1 = 1 + \frac{1}{n}\sum_{i=1}^{n} \mathbf{Q}_i^n$, which, together with symmetry yields

$$\mathbb{E}\left[\|S^n\|_1\right] = 1 + \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\mathbf{Q}_i^n\right] = \mathbb{E}\left[\mathbf{Q}_1^n\right],$$

and concludes the proof. □

We now prove our final result on the interchange of limits.

Proof of Theorem 3.1.4. Consider the set $\mathbb{Z}_+ \cup \{\infty\}$ endowed with the topology of the Alexandroff compactification, which is known to be metrizable. Moreover, it can be seen that the topology defined by the norm $\|\cdot\|_w$ on $[0,1]^{\mathbb{Z}_+}$ is equivalent to the product topology, which makes $[0,1]^{\mathbb{Z}_+}$ compact. As a result, the product $\{s \in \mathcal{S}^1 : \|s\|_1 \leq M\} \times (\mathbb{Z}_+ \cup \{\infty\})$ is closed, and thus compact, for all M. Note that, for each n, the invariant distribution π^n is defined over the set $(\mathcal{S}^1 \cap \mathcal{I}_n) \times \{0, 1, \ldots, c_n\}$. This is a subset of $\mathcal{S}^1 \times (\mathbb{Z}_+ \cup \{\infty\})$, so we can extend the measures π^n to the latter, larger set.

Let $\{S^n(0)\}_{n=1}^{\infty}$ be a sequence of random variables distributed according to the marginals $\{\pi_s^n\}_{n=1}^{\infty}$. From Lemma 3.4.1, we have

$$\mathbb{E}_{\pi^n}\left[\|S^n(0)\|_1\right] \leq 2 + \frac{2}{1-\lambda}, \quad \forall\, n. \tag{3.33}$$

Using Markov's inequality, it follows that for every $\epsilon > 0$, there exists a constant M such that

$$\pi_s^n\big(\{s \in \mathcal{S}^1 : \|s\|_1 \leq M\}\big) \geq 1 - \epsilon, \quad \forall\, n,$$

which implies that

$$\pi^n\big(\{s \in \mathcal{S}^1 : \|s\|_1 \leq M\} \times (\mathbb{Z}_+ \cup \{\infty\})\big) \geq 1 - \epsilon, \quad \forall\, n.$$

Thus, the sequence $\{\pi^n\}_{n=1}^{\infty}$ is tight and, by Prohorov's theorem [11], it is also relatively compact in the weak topology on the set of probability measures. It follows that

any subsequence has a weakly convergent subsequence whose limit is a probability measure over $\mathcal{S}^1 \times (\mathbb{Z}_+ \cup \{\infty\})$.

Let $\{\pi^{n_k}\}_{k=1}^{\infty}$ be a weakly convergent subsequence, and let π be its limit. Let $S(0)$ be a random variable distributed according to π_s, where π_s is the marginal of π. Since $\mathcal{S}^1 \times (\mathbb{Z}_+ \cup \{\infty\})$ is separable, we invoke Skorokhod's representation theorem to construct a probability space $(\Omega_0, \mathcal{A}_0, \mathbb{P}_0)$ and a sequence of random variables $(S^{n_k}(0), M^{n_k}(0))$ distributed according to π^{n_k}, such that

$$\lim_{k\to\infty} \|S^{n_k}(0) - S(0)\|_w = 0 \qquad \mathbb{P}_0 - a.s. \tag{3.34}$$

We use the random variables $(S^{n_k}(0), M^{n_k}(0))$ as the initial conditions for a sequence of processes $\{(S^{n_k}(\cdot), M^{n_k}(\cdot))\}_{k=1}^{\infty}$, so that each one of these processes is stationary. Note that the initial conditions, distributed as π^{n_k}, do not necessarily converge to a deterministic initial condition (this is actually part of what we are trying to prove), so we cannot use Theorem 3.1.2 directly to find the limit of the sequence of processes $\{S^{n_k}(\cdot)\}_{k=1}^{\infty}$. However, given any $\omega \in \Omega_0$ outside a zero $\mathbb{P}_0$-measure set, we can restrict this sequence of stochastic processes to the probability space

$$(\Omega_\omega, \mathcal{A}_\omega, \mathbb{P}_\omega) = (\Omega_D \times \Omega_S \times \{\omega\}, \mathcal{A}_D \times \mathcal{A}_S \times \{\emptyset, \{\omega\}\}, \mathbb{P}_D \times \mathbb{P}_S \times \delta_\omega)$$

and apply Theorem 3.1.2 to this new space, to obtain

$$\lim_{k\to\infty} \sup_{0\le t\le T} \|S^{n_k}(t,\omega) - S(t,\omega)\|_w = 0, \quad \mathbb{P}_\omega - a.s.,$$

where $S(t,\omega)$ is the fluid solution with initial condition $S(0,\omega)$. Since this is true for all $\omega \in \Omega_0$ except for a set of zero P_0-measure, it follows that

$$\lim_{k\to\infty} \sup_{0\le t\le T} \|S^{n_k}(t) - S(t)\|_w = 0, \qquad \mathbb{P} - a.s.,$$

where $\mathbb{P} = \mathbb{P}_D \times \mathbb{P}_S \times \mathbb{P}_0$ and where $S(\cdot)$ is another stochastic process whose randomness is only in the initial condition $S(0)$ (its trajectory is the deterministic fluid solution

for that specific initial condition).

We use Lemma 3.4.1 once again to interchange limit, expectation, and infinite summation in Equation (3.33) (using the same argument as in Lemma 3.1.5) to obtain

$$\mathbb{E}_{\pi_s}\big[\|S(0)\|_1\big] \leq 2 + \frac{2}{1-\lambda}.$$

Using Markov's inequality now in the limit, it follows that for every $\epsilon > 0$, there exists a constant M such that

$$\pi_s\big(\|S(0)\|_1 \leq M\big) \geq 1 - \epsilon. \tag{3.35}$$

Recall that the uniqueness of fluid solutions (Theorem 3.1.1) implies the continuous dependence of solutions on initial conditions [16]. Moreover, Theorem 3.1.1 implies that any solution $s(\cdot)$ with initial conditions $s(0) \in \mathcal{S}^1$ converges to s^* in time. As a result, there exists $T_\epsilon > 0$ such that

$$\sup_{s(0):\, \|s(0)\|_1 \leq M} \big\|s(T_\epsilon) - s^*\big\|_w < \epsilon.$$

Combining this with Equation (3.35), we obtain

$$\begin{aligned}
\mathbb{E}_{\pi_s}\big[\|S(T_\epsilon) - s^*\|_w\big] &= \mathbb{E}_{\pi_s}\Big[\|S(T_\epsilon) - s^*\|_w \;\Big|\; \|S(0)\|_1 \leq M\Big]\, \pi_s\big(\|S(0)\|_1 \leq M\big) \\
&\quad + \mathbb{E}_{\pi_s}\Big[\|S(T_\epsilon) - s^*\|_w \;\Big|\; \|S(0)\|_1 > M\Big]\, \pi_s\big(\|S(0)\|_1 > M\big) \\
&< \epsilon + \big(\sup_{s\in\mathcal{S}} \|s - s^*\|_w\big)\epsilon \\
&\leq 2\epsilon,
\end{aligned} \tag{3.36}$$

where the expectations $\mathbb{E}_{\pi_s}$ are with respect to the random variable $S(0)$, distributed according to π_s, even though the dependence on $S(0)$ is suppressed from our notation and is left implicit. On the other hand, due to the stationarity of $S^{n_k}(\cdot)$, the random variables $S^{n_k}(0)$ and $S^{n_k}(T_\epsilon)$ have the same distribution, for any k. Taking the limit as $k \to \infty$, we see that $S(0)$ and $S(T_\epsilon)$ have the same distribution. Combining this

with Equation (3.36), we obtain

$$\mathbb{E}_{\pi_s}\left[\|S(0) - s^*\|_w\right] \leq 2\epsilon.$$

Since ϵ was arbitrary, it follows that $S(0) = s^*$, π_s-almost surely, i.e., the distribution π_s of $S(0)$ is concentrated on s^*. We have shown that the limit π_s of a convergent subsequence of π^n is the Dirac measure δ_{s^*}. Since this is true for every convergent subsequence and π^n is tight, this implies that π^n converges to δ_{s^*}, as claimed. □

3.5 Conclusions and future work

The main objective of this chapter was to study the tradeoff between the amount of resources (messages and memory) available to a central dispatcher, and the expected queueing delay of a typical job, as the system size increases. We introduced a parametric family of dispatching policies and, using a fluid model and associated convergence theorems, we showed that with enough resources, we can drive the queueing delay to zero as the system size increases.

We also analyzed a resource constrained regime of our pull-based policies that, although it does not have vanishing queueing delay, it has some remarkable properties. We showed that by wisely exploiting an arbitrarily small message rate (but still proportional to the arrival rate), we obtain a queueing delay which is finite and uniformly upper bounded for all $\lambda < 1$, a significant qualitative improvement over the queueing delay of the M/M/1 queue (obtained when we use no messages). Furthermore, we compared it with the popular power-of-d-choices and found that, while using the same number of messages, our policy achieves a much lower expected queueing delay, especially when λ is close to 1.

There are several interesting directions for future research. For example:

(i) It would be interesting to extend these results to the case of different service disciplines such as processor sharing or LIFO, or to the case of general service time distributions, as these are prevalent in many applications.

(ii) Another interesting line of work is to consider a reverse problem, in which we have decentralized arrivals to several queues, a central server, and a scheduler that needs to decide which queue to serve. In this context we expect to again find a similar tradeoff between the resources used and the queueing delay.

Chapter 4

Universal delay lower bound for dispatching policies

As in Chapter 3, we focus again on distributed service systems consisting of a large number of servers with homogeneous service rates. However, instead of studying yet another policy or decision making architecture, we step back and address a more fundamental question: what is the minimum amount of resources required to obtain the best possible delay performance, as the number of servers increases? Regarding performance, we focus on the expected queueing delay of a typical job. Regarding resources, we focus on the average number of messages exchanged between the dispatcher and the servers per unit of time, and on the number of bits of "long term" memory that the dispatcher has at its disposal.

More concretely, we introduce a unified framework for dispatching policies and show that if the memory size (in bits) is logarithmic in the number of servers and the average message rate is proportional to the arrival rate, then the expected queueing delay of a typical job cannot vanish as the system size increases. This complements the results in Chapter 3 where we showed that if we relax either one of those restrictions, there exist dispatching policies where the expected queueing delay of a typical job vanishes as the system size increases.

In order to establish the impossibility results described above, we develop a novel combinatorial approach to handle the constraint on memory size, which involves es-

tablishing the impact of limited memory on the different decisions that a dispatcher can make. Using these results, we obtain a universal lower bound for the expected queueing delay of a typical job, which implies the desired result.

The rest of the chapter is organized as follows. The model and the main result are presented in Section 4.1. In Section 4.2 we discuss our result in the context of some concrete dispatching policies from the earlier literature. In Section 4.3 we provide the proof of our main result. Finally, in Section 4.5 we present our conclusions and suggestions for future work.

The results on this chapter first appeared in [19] and [21].

4.1 Model and main results

In this section we present the specific modeling assumptions, the performance metrics of interest, and our main results. In Subsection 4.1.1 we describe the model and our assumptions. In Subsection 4.1.2 we present a unified framework that defines a broad set of dispatching policies, which includes most of the policies studied in previous literature. In Subsection 4.1.3 we present our negative result on the expected queueing delay under resource constrained policies within this set of policies. Finally, in Subsection 4.1.4 we combine the results in this chapter with the ones in Chapter 3 to better understand the tradeoff between resources and queueing delay.

4.1.1 Modeling assumptions and performance metric

We now introduce a refinement of the modeling assumptions for the basic model introduced in Section 2.2. In particular, throughout this chapter we assume that the n servers are homogeneous, and have a constant processing rate equal to 1. Furthermore, jobs arrive to the system as a single renewal process of rate λn (for some fixed $\lambda \in (0, 1)$), and are i.i.d., independent from the arrival process, and have a general distribution with unit mean. Finally, the central dispatcher has to route each

incoming job to a queue immediately upon arrival (i.e., jobs cannot be queued at the dispatcher).

The dispatcher has limited information on the state of the queues; it can only rely on a limited amount of local memory and on messages that provide partial information about the state of the system. These messages (which are assumed to be instantaneous) can be sent from a server to the dispatcher at any time, or from the dispatcher to a server (in the form of queries) at the time of an arrival. Messages from a server can only contain information about the state of its own queue (number of remaining jobs and the remaining workload of each one). Within this context, a system designer has the freedom to choose a messaging policy, as well as the rules for updating the memory and for selecting the destination of an incoming job.

Regarding the performance metric, we will focus on the steady-state expectation of the time between the arrival of a typical job and the time at which it starts receiving service, to be referred to as the **expected queueing delay of a typical job**. We will formalize this definition in Subsection 4.1.3.

4.1.2 Unified framework for dispatching policies

In this subsection we present a unified framework that describes memory-based dispatching policies. In order to do this, we fix n and we introduce a sample path construction of the evolution of the system under an arbitrary policy.

Let c_n be the number of memory bits available to the dispatcher. We define the corresponding set of memory states to be $\mathcal{M}_n \triangleq \{1, \ldots, 2^{c_n}\}$. Furthermore, we define the set of possible states at a server as the set of nonnegative sequences $\mathcal{Q} \triangleq \mathbb{R}_+^{\mathbb{Z}_+}$, where a sequence specifies the remaining workload of each job in that queue, including the one that is being served. (In particular, an idle server is represented by the zero sequence.) As long as a queue has a finite number of jobs, the queue state is a sequence that has only a finite number of non-zero entries. The reason that we include the workload of the jobs in the state is that we wish to allow for a broad class of policies, that can take into account the remaining workload in the queues. In particular, we

allow for information-rich messages that describe the full workload sequence at the server that sends the message. We are interested in the process

$$\mathbf{Q}(\cdot) = \big(\mathbf{Q}_1(\cdot), \ldots, \mathbf{Q}_n(\cdot)\big) = \Big(\big(\mathbf{Q}_{1,j}(\cdot)\big)_{j=1}^{\infty}, \ldots, \big(\mathbf{Q}_{n,j}(\cdot)\big)_{j=1}^{\infty}\Big),$$

which takes values in the set $\mathcal{Q}^n$, and describes the evolution of the workload of each job in each queue. Here $\mathbf{Q}_{i,j}(t)$ is the remaining workload of the j-th job in the i-th queue, at time t, which for $j \geq 2$ is simply the job's service time. We are also interested in the process $M(\cdot)$ that describes the evolution of the memory state, and in the process $Z(\cdot)$ that describes the remaining time until the next arrival of a job.

Fundamental processes and initial conditions

The processes of interest will be driven by certain common fundamental processes.

1. **Arrival process:** A delayed renewal counting process $A_n(\cdot)$ with rate λn, and event times $\{T_k\}_{k=1}^{\infty}$, defined on a probability space $(\Omega_A, \mathcal{A}_A, \mathbb{P}_A)$.

2. **Spontaneous messages process:** A Poisson counting process $R_n(\cdot)$ with rate βn, and event times $\{T_k^s\}_{k=1}^{\infty}$, defined on a probability space $(\Omega_R, \mathcal{A}_R, \mathbb{P}_R)$.

3. **Job sizes:** A sequence of i.i.d. random variables $\{W_k\}_{k=1}^{\infty}$ with mean one, defined on a probability space $(\Omega_W, \mathcal{A}_W, \mathbb{P}_W)$.

4. **Randomization variables:** Four independent and individually i.i.d. sequences of random variables $\{U_k\}_{k=1}^{\infty}$, $\{V_k\}_{k=1}^{\infty}$, $\{X_k\}_{k=1}^{\infty}$, and $\{Y_k\}_{k=1}^{\infty}$, uniform on $[0,1]$, defined on a common probability space $(\Omega_U, \mathcal{A}_U, \mathbb{P}_U)$.

5. **Initial conditions:** Random variables $\mathbf{Q}(0)$, $M(0)$, and $Z(0)$, defined on a common probability space $(\Omega_0, \mathcal{A}_0, \mathbb{P}_0)$.

The whole system will be defined on the associated product probability space

$$\big(\Omega_A \times \Omega_R \times \Omega_W \times \Omega_U \times \Omega_0, \mathcal{A}_A \times \mathcal{A}_R \times \mathcal{A}_W \times \mathcal{A}_U \times \mathcal{A}_0, \mathbb{P}_A \times \mathbb{P}_R \times \mathbb{P}_W \times \mathbb{P}_U \times \mathbb{P}_0\big),$$

to be denoted by $(\Omega, \mathcal{A}, \mathbb{P})$. All of the randomness in the system originates from these fundamental processes, and everything else is a deterministic function of them.

A construction of sample paths

We now consider some fixed n, and provide a construction of a Markov process $\big(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot)\big)$, that takes values in the set $\mathcal{Q}^n \times \mathcal{M}_n \times \mathbb{R}_+$. The memory process $M(\cdot)$ is piecewise constant, and can only jump at the time of an event. All processes to be considered will have the càdlàg property (right-continuous with left limits) either by assumption (e.g., the underlying fundamental processes) or by construction.

There are three types of events: job arrivals, spontaneous messages, and service completions. We now describe the sources of these events, and what happens when they occur.

Job arrivals: At the time of the k-th event of the arrival process A_n, which occurs at time T_k and involves a job with size W_k, the following transitions happen sequentially but instantaneously.

1. First, the dispatcher chooses a vector of distinct servers $\mathbf{S}_k$, from which it solicits information about their state, according to

$$\mathbf{S}_k = f_1\Big(M\left(T_k^-\right), W_k, U_k\Big),$$

   where $f_1 : \mathcal{M}_n \times \mathbb{R}_+ \times [0,1] \to \mathcal{R}_n$ is a measurable function defined by the policy. Note that the set of servers that are sampled only depends on the current memory state and on the size of the incoming job, but it is chosen in a randomized way, thanks to the independent random variable U_k. Thus, we allow for randomized policies; for example, the dispatcher might choose to sample a fixed number of servers uniformly at random.

2. Then, messages are sent to the servers in the vector $\mathbf{S}_k$, and the servers respond with messages containing their queue states; thus, the information received by

the dispatcher is the vector $\mathbf{Q}_{\mathbf{S}_k}$. This results in $2|\mathbf{S}_k|$ messages exchanged. Using this information, the destination of the incoming job is chosen to be

$$D_k = f_2\Big(M\big(T_k^-\big), W_k, \mathbf{S}_k, \mathbf{Q}_{\mathbf{S}_k}\big(T_k^-\big), V_k\Big),$$

where $f_2 : \mathcal{M}_n \times \mathbb{R}_+ \times \mathcal{R}_n \times \big(\cup_{i=0}^n \mathcal{Q}^i\big) \times [0,1] \to \mathcal{N}_n$ is a measurable function defined by the policy. Note that the destination of a job can only depend on the current memory state, the job size, as well as the vector of queried servers and the state of their queues, but it is chosen in a randomized way, thanks to the independent random variable V_k. Once again, we allow for randomized policies that, for example, dispatch jobs uniformly at random.

3. Finally, the memory state is updated according to

$$M(T_k) = f_3\Big(M\big(T_k^-\big), W_k, \mathbf{S}_k, \mathbf{Q}_{\mathbf{S}_k}\big(T_k^-\big), D_k\Big),$$

where $f_3 : \mathcal{M}_n \times \mathbb{R}_+ \times \mathcal{R}_n \times \big(\cup_{i=0}^n \mathcal{Q}^i\big) \times \mathcal{N}_n \to \mathcal{M}_n$ is a measurable function defined by the policy. Note that the new memory state is obtained using the same information as for selecting the destination, plus the destination of the job, but without randomization.

Spontaneous messages: At the time of the k-th event of the spontaneous message process R_n, which occurs at time T_k^s, the i-th server sends a spontaneous message to the dispatcher if and only if

$$g_1\Big(\mathbf{Q}\big(T_k^s\big), X_k\Big) = i,$$

where $g_1 : \mathcal{Q}^n \times [0,1] \to \{0\} \cup \mathcal{N}_n$ is a measurable function defined by the policy. In that case, the memory is updated to the new memory state

$$M(T_k^s) = g_2\Big(M\big(T_k^{s-}\big), i, \mathbf{Q}_i\big(T_k^s\big)\Big),$$

where $g_2 : \mathcal{M}_n \times \mathcal{N}_n \times \mathcal{Q} \to \mathcal{M}_n$ is a measurable function defined by the policy, and

which prescribes the server who sends a message. On the other hand, no message is sent when $g_1\big(\mathbf{Q}(T_k^s), X_k\big) = 0$. Note that the dependence of g_1 on $\mathbf{Q}$ allows the message rate at each server to depend on the server's current workload. For example, we could let idle servers send repeated spontaneous messages (as a Poisson process) to inform the dispatcher of their idleness.

Service completions: As time progresses, the remaining workload of each job that is at the head of line in a queue decreases at a constant, unit rate. When a job's workload reaches zero, the job leaves the system and every other job advances one slot. Let $\{T_k^d(i)\}_{k=1}^{\infty}$ be the sequence of departure times at the i-th server. At those times, the i-th server sends a message to the dispatcher if and only if

$$h_1\Big(\mathbf{Q}_i\big(T_k^d(i)\big), Y_k\Big) = 1,$$

where $h_1 : \mathcal{Q} \times [0,1] \to \{0,1\}$ is a measurable function defined by the policy. In that case, the memory is updated to the new memory state

$$M\Big(T_k^d(i)\Big) = h_2\Big(M\big(T_k^d(i)^-\big), i, \mathbf{Q}_i\big(T_k^d(i)\big)\Big),$$

where $h_2 : \mathcal{M}_n \times \mathcal{N}_n \times \mathcal{Q} \to \mathcal{M}_n$ is a measurable function defined by the policy. On the other hand, no message is sent when $h_1\big(\mathbf{Q}_i(T_k^d(i)), Y_k\big) = 0$.

Remark 4.1.1. We have chosen to describe the collection of queried servers by a vector, implying an ordering of the servers in that collection. We could have described this collection as an (unordered) set. These two options are essentially equivalent but it turns out that the ordering provided by the vector description allows for a simpler presentation of the proof.

Remark 4.1.2. For any given n, a policy is completely determined by the spontaneous message rate β, and the functions f_1, f_2, f_3, g_1, g_2, h_1, and h_2. Furthermore, many policies in the literature that are described without explicit mention of memory or messages can be cast within our framework, as we will see in Section 4.2.

Remark 4.1.3. The memory update functions f_3, g_2, and h_2 do not involve randomization, even though our main result could be extended in that direction. We made this choice because none of the policies introduced in earlier literature require such randomization, and because it simplifies notation and the proofs.

Remark 4.1.4. We only consider the memory used to store information in between arrivals or messages. Thus, when counting the memory resources used by a policy, we do not take into account information that is used in zero time (e.g., the responses from the queries at the time of an arrival) or the memory required to evaluate the various functions that describe the policy. If that additional memory were to be accounted for, then any memory constraints would be more severe, and therefore our negative result would still hold.

The dispatching policies that we have introduced obey certain constraints:

(i) The dispatcher can only send messages to the servers at the time of an arrival, and in a single round of communication. This eliminates the possibility of policies that sequentially poll the servers uniformly at random until they find an idle one. Indeed, it can be shown that such sequential polling policies may lead to asymptotically vanishing delays, without contradicting our lower bounds. On the other hand, in practice, queries involve some processing and travel time ϵ. Were we to consider a more realistic model with $\epsilon > 0$, sequential polling would also incur positive delay.

(ii) We assume that the dispatcher must immediately send an incoming job to a server upon arrival. This prevents the dispatcher from maintaining a centralized queue and operating the system as a G/G/n queue.

We now introduce a symmetry assumption on the policies. In essence it states that at the time of a job arrival, and given the current memory state, if certain sampling and dispatching decisions and a certain memory update are possible, then a permuted version of these decisions and updates is also possible (and equally likely), starting with a suitably permuted memory state.

Assumption 4.1.1. (Symmetric policies.) We assume that the dispatching policy is symmetric, in the following sense. For any given permutation of the servers σ, there exists a corresponding (not necessarily unique) permutation σ_M of the memory states $\mathcal{M}_n$ that satisfies all of the following properties.

1. For every $m \in \mathcal{M}_n$ and $w \in \mathbb{R}_+$, and if U is a uniform random variable on $[0,1]$, then
$$\sigma\Big(f_1(m,w,U)\Big) \stackrel{d}{=} f_1\big(\sigma_M(m),w,U\big),$$
where $\stackrel{d}{=}$ stands for equality in distribution.

2. For every $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, $\mathbf{s} \in \mathcal{R}_n$, and $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, and if V is a uniform random variable on $[0,1]$, then[1]
$$\sigma\Big(f_2\big(m,w,\mathbf{s},\mathbf{q},V\big)\Big) \stackrel{d}{=} f_2\big(\sigma_M(m),w,\sigma(\mathbf{s}),\mathbf{q},V\big).$$

3. For every $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, $\mathbf{s} \in \mathcal{R}_n$, and $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, and $d \in \mathcal{N}_n$, we have
$$\sigma_M\Big(f_3\big(m,w,\mathbf{s},\mathbf{q},d\big)\Big) = f_3\big(\sigma_M(m),w,\sigma(\mathbf{s}),\mathbf{q},\sigma(d)\big).$$

As a concrete illustration, our symmetry assumption implies the following. If a certain memory state mandates that the vector $(2,4,5)$ of servers must be sampled (with probability 1), independently from the incoming job size, then there exists some other memory state which mandates that the vector $(1,5,7)$ will be sampled, independently from the incoming job size, and the same holds for every 3-element vector with distinct entries. Since there are $n(n-1)(n-2)$ different vectors, there must be at least so many different memory states. This suggests that if we have too few memory states, the number of "distinguished" servers, i.e., servers that are treated in a special manner is severely limited. This is a key element of the proof of the delay

[1]Note that the argument on the right-hand side of the relation below involves $\mathbf{q}$ rather than a permuted version of $\mathbf{q}$, even though the vector $\mathbf{s}$ gets permuted. We are essentially comparing a situation where the dispatcher queries a vector $\mathbf{s}$ and receives certain numerical values $\mathbf{q}$ with the situation where the dispatcher queries a vector $\sigma(\mathbf{s})$ and receives the **same** numerical values $\mathbf{q}$.

lower bound that we present in the next subsection.

One may contemplate a different (stronger) definition of symmetry. For example, in the first part, we could have required that

$$\sigma\big(f_1(m,w,u)\big) = f_1\big(\sigma_M(m),w,u\big), \qquad \forall u \in [0,1]. \tag{4.1}$$

While this would lead to a simpler proof, this stronger definition would be too restrictive. This is shown in the following example.

Example 4.1.1. Consider a policy that samples a fixed number d of servers, uniformly at random (regardless of the memory state and of the incoming job size), and that satisfies this stronger symmetry assumption. Then, $f_1(m,w,u)$ is a vector of dimension d, for all $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, and $u \in [0,1]$. Let σ, τ be a pair of permutations such that $\sigma(f_1(m,w,u)) \neq \tau(f_1(m,w,u))$. The stronger symmetry assumption in Equation (4.1) implies that there exists a pair of associated permutations σ_M, τ_M of the memory states such that

$$f_1\big(\sigma_M(m),w,u\big) = \sigma\big(f_1(m,w,u)\big) \neq \tau\big(f_1(m,w,u)\big) = f_1\big(\tau_M(m),w,u\big).$$

It follows that $\sigma_M(m) \neq \tau_M(m)$, and thus there must be at least as many memory states as the number of different vectors of dimension d with different entries. There are $\binom{n}{d}d!$ such vectors, and therefore a large memory would be required to implement such a uniform sampling policy if Equation (4.1) were to be enforced.

On the other hand, the symmetry assumption that we have adopted in this chapter only requires equality in distribution, and uniform sampling can be achieved with only one memory state (i.e., with no bits of memory). Indeed, since the sampling of servers is done uniformly at random, we have

$$f_1(m,w,U) \overset{d}{=} \sigma\big(f_1(m,w,U)\big),$$

for all permutations σ.

This example shows that the symmetry assumption that we have adopted can be substantially weaker (and thus easier to satisfy), and allows small-memory implementation of simple natural policies.

Remark 4.1.5. Note that a symmetry assumption is imposed on the memory update function f_3 at the time that a job is dispatched. However, we do not need to impose a similar assumption on the memory update functions g_2 and h_2 at the times that the dispatcher receives a message. Similarly, there is no symmetry assumption on the functions g_1 and h_1 that govern the generation of server messages. In particular, we allow each server to send spontaneous messages at its own identity-dependent, and hence asymmetric, rate.

4.1.3 Delay lower bound for resource constrained policies

Before stating the main result of this chapter, we introduce formal definitions for the average message rate between the dispatcher and the servers, and for our performance metric for the delay. Furthermore, we introduce an assumption on the arrival process.

First, given a policy of the form specified in the previous subsection, we define the **average message rate** between the dispatcher and the servers as

$$\limsup_{t\to\infty} \frac{1}{t} \left[\sum_{k=1}^{A_n(t)} 2|\mathbf{S}_k| + \sum_{k=1}^{R_n(t)} \mathbb{1}_{\mathcal{N}_n}\Big(g_1\big(\mathbf{Q}\big(T_k^s\big), X_k\big)\Big) + \sum_{i=1}^{n} \sum_{k:\, T_k^d(i)<t} \mathbb{1}_{\{1\}}\Big(h_1\big(\mathbf{Q}_i\big(T_k^d(i)\big), Y_k\big)\Big) \right]. \quad (4.2)$$

Second, we provide a formal definition of our performance metric for the delay. We assume that the process $\big(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot)\big)$ is stationary, with invariant probability measure π. Since the destinations of jobs (and their queueing delays) are deterministic functions of the state and i.i.d. randomization variables, the point process of arrivals with the queueing delays as marks, is also stationary. Using this, we define the **expected queueing delay in steady-state π of a typical job**, denoted by $\mathbb{E}_\pi^0[L_0]$, as follows. If L_k is the queueing delay of the k-th job under the stationary process

$\big(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot)\big)$, then

$$\mathbb{E}_\pi^0 [L_0] \triangleq \mathbb{E}_\pi \left[\frac{1}{\lambda n t} \sum_{k=1}^{A_n(t)} L_k \right], \tag{4.3}$$

where the right-hand side is independent from t due to the stationarity of the processes involved (see [7]). Furthermore, if the stationary process $\big(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot)\big)$ is ergodic (in the sense that every invariant set has measure either 0 or 1 under π), we have

$$\mathbb{E}_\pi^0 [L_0] = \lim_{t\to\infty} \frac{1}{A_n(t)} \sum_{k=1}^{A_n(t)} L_k, \qquad a.s.$$

Finally, we introduce an assumption on the arrival process.

Assumption 4.1.2. Let I_n be distributed as the typical inter-arrival times of the delayed renewal process $A_n(\cdot)$. We assume that there exists a constant $\bar{\epsilon} > 0$, independent from n, such that the following holds. For every $\epsilon \in (0, \bar{\epsilon}]$, there exists a positive constant δ_ϵ such that

$$\delta_\epsilon \leq \mathbb{P}\left(I_n \leq \frac{\epsilon}{n}\right) \leq 1 - \delta_\epsilon,$$

for all n.

This assumption implies that arbitrarily small inter-arrival times of order $\Theta(1/n)$ occur with a probability that is bounded away from 0, and from 1, for all n. In particular, this excludes deterministic inter-arrival times, and inter-arrival times that can take values of order $o(1/n)$ with probability of order $1 - o(1)$. On the other hand, if $A(\cdot)$ is a delayed renewal process, where the typical inter-arrival times are continuous random variables with positive density around 0, then the process $A_n(\cdot)$, defined as $A_n(t) \triangleq A(nt)$ for all $t \geq 0$, satisfies Assumption 4.1.2.

We are now ready to state the main result. It asserts that within the class of symmetric policies that we consider, and under some upper bounds on the mem-

ory size (logarithmic) and the message rate (linear), the expected queueing delay in steady-state of a typical job is bounded below by a positive constant.

Theorem 4.1.1 (Positive delay for resource constrained policies)**.** *For any constants $\lambda \in (0,1)$, c, $\alpha > 0$, and for every arrival process that satisfies Assumption 4.1.2, there exists a constant $\zeta(\lambda, c, \alpha) > 0$ with the following property. For any fixed n, consider a symmetric memory-based dispatching policy, i.e., that satisfies Assumption 4.1.1, with at most $c\log_2(n)$ bits of memory, with an average message rate (cf. Equation (4.2)) upper bounded by αn in expectation, and under which the process $\big(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot)\big)$ admits at least one invariant probability measure π_n. Then, for all n large enough, we have*

$$\mathbb{E}^0_{\pi_n}[L_0] \geq \zeta(\lambda, c, \alpha),$$

where $\mathbb{E}^0_{\pi_n}[L_0]$ is the expected queueing delay in steady-state π_n of a typical job.

The proof is given in Section 4.3.

4.1.4 Queueing delay vs resources tradeoff

In this subsection we summarize the results obtained in Chapter 3 and in this chapter. First, recall that Corollary 3.1.6 in Chapter 3 implies that with either (i) a memory of size (in bits) of order $\Omega(\log(n))$ and a message rate of order $\omega(n)$, or (ii) a memory of size (in bits) of order $\omega(\log(n))$ and a message rate greater than or equal to λn, there exists a policy with vanishing queueing delay. Second, note that Theorem 4.1.1 states that symmetric policies with $O(\log(n))$ bits of memory and a message rate of order $O(n)$ cannot have vanishing queueing delays. Finally, there is a third regime (policies with $\Omega(\log(n))$ bits of memory and a message rate smaller than λn) for which there are no known results. The three regimes are depicted in Figure 4-1.

4.2 Literature review

In this section we put our results in perspective by showing that various dispatching policies considered earlier in the literature are special cases of the class of symmetric

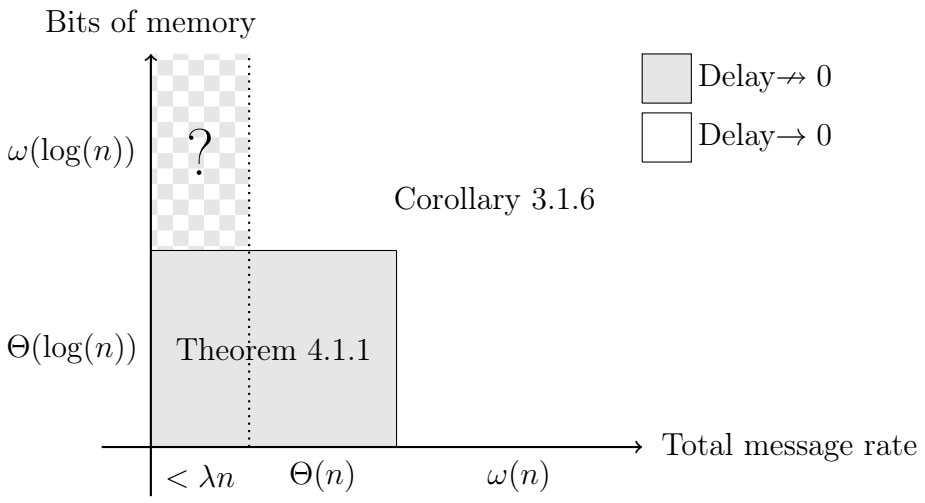

Figure 4-1: Resource requirements for vanishing queueing delays.

dispatching policies described above. Most policies have only been studied for the case of Poisson arrivals and exponential service times, so this review is restricted to that case unless stated otherwise.

Open-loop policies

Random routing. The simplest policy is to dispatch each arriving job to a random queue, with each queue being equally likely to be selected. In this case, the system behaves as n independent parallel M/M/1 queues. This policy needs no messages or memory, and has a positive queueing delay independent of n.

Round-Robin (RR). When the dispatcher has no access to the workload of incoming jobs and no messages are allowed, it is optimal to dispatch arriving jobs to the queues in a Round-Robin fashion [39]. This policy does not require messages but needs $\lceil \log_2(n) \rceil$ bits of memory to store the ID of the next queue to receive a job. In the limit, each queue behaves like a D/M/1 queue (see [39]). While random routing is a symmetric policy, Round-Robin is not. To see this, note that a memory state i must be followed by state $i+1$, and such a transition is not permutation-invariant; in particular, the memory update function f_3 does not satisfy the symmetry assumption. Round-Robin can be made symmetric by using an additional $\lceil n \log_2(n) \rceil$ bits

of memory to specify the order with which the different servers are selected. But in any case, this policy also has a positive queueing delay, that does not vanish as n increases.

Policies based on queue lengths

Join a shortest queue (SQ). If we wish to minimize the queueing delay and have access to the queue lengths but not to the job sizes, an optimal policy is to have each incoming job join a shortest queue, breaking any ties uniformly at random [45]. When n goes to infinity, the queueing delay vanishes, but this policy requires a message rate of $2\lambda n^2$ (n queries and n responses for each arrival), and no memory. This policy is symmetric and achieves vanishing delay, but uses a superlinear number of messages.

Join a shortest of d random queues (SQ(d)). In order to sharply decrease the number of messages sent, Mitzenmacher [34] and Vvedenskaya et al. [44] introduced the power-of-d-choices policy. When there is an arrival, d servers are chosen uniformly at random, and the job is sent to a shortest queue among those d servers. This policy fits our framework, and in particular is symmetric; it uses $2\lambda dn$ messages per unit of time, and zero memory. This policy was also analyzed in the case of heavy-tailed service times by Bramson et al. [14], yielding similar results. In any case, this policy has positive delay, which is consistent with Theorem 4.1.1.

Join a shortest of d_n random queues (SQ(d_n)). More recently, Mukherjee et al. [35] analyzed a variation of the SQ(d) policy, which lets d be a function of the system size n. This policy is symmetric, uses $2\lambda d_n n$ messages per unit of time and zero memory, and has zero delay as long as $d_n \to \infty$, which is consistent with Theorem 4.1.1.

Join a shortest of d queues, with memory (SQ(d,b)). Another improvement over the power-of-d-choices, proposed by Mitzenmacher et al. in [33], is obtained by using extra memory to store the IDs of the b (with $b \leq d$) least loaded queues known at the time of the previous arrival. When a new job arrives, d queues are sampled

uniformly at random and the job is sent to a least loaded queue among the d sampled and the b stored queues. This policy is symmetric, needs $2\lambda dn$ messages per unit of time and $\lceil b \log_2(n) \rceil$ bits of memory, and has positive delay, consistent with Theorem 4.1.1.

SQ(d) for divisible jobs. Recently, Ying et al. [47] considered the case of jobs of size m_n (with $m_n \in \omega(1)$ and $m_n/n \to 0$) arriving as a Poisson process of rate $n\lambda/m_n$, where each job can be divided into m_n tasks with mean size 1. Then, the dispatcher samples dm_n queues and does a water-filling of those queues with the m_n tasks. In this case, the number of messages sent per unit of time is $2\lambda dn$ and no memory is used. Even though this was not mentioned in [47], this policy can be shown to drive the queueing delay to 0 if $d \geq 1/(1-\lambda)$. However, this model does not fall into our framework because it involves divisible jobs.

Policies based on remaining workload

Join a least loaded queue (LL). An appealing policy is the one that sends incoming jobs to a queue with the least remaining workload, in which case the whole system behaves as an M/M/n queue. This policy is symmetric and achieves a vanishing delay as $n \to \infty$, but it has the same quadratic messaging requirements as SQ.

Join a least loaded of d queues (LL(d)). A counterpart of SQ(d) is LL(d), in which the dispatcher upon arrival chooses d queues uniformly at random and sends the job to one of those queues with the least remaining workload, breaking any ties uniformly at random. This setting was studied in [25], and it does not result in asymptotically vanishing delay, consistent with Theorem 4.1.1.

Policies based on job size

The previous policies dispatched the incoming jobs based on information about the state of the queues, obtained by dynamically exchanging messages with the servers.

Such information could include the remaining workload at the different queues. On the other hand, if the dispatcher only knows the size of an incoming job (which might be difficult in practice [15]), it could use a static and memoryless policy that selects a target server based solely on the job size. Harchol-Balter et al. [24] showed that delay is minimized over all such static policies by a non-symmetric policy that partitions the set of possible job sizes into consecutive intervals and assigns each interval to a different server. This is especially effective when the jobs have highly variable sizes (e.g., heavy-tailed), yet the resulting delay can be no better than that of an M/D/1 queue, and does not vanish as $n \to \infty$. This scheme does not require any message exchanges, and could be made symmetric by using the memory to store a list of the n intervals of job sizes corresponding to each of the n servers.

Pull-based load balancing

Join-Idle-Queue (JIQ). In order to reduce the message rate, Badonnel and Burgess [8], Lu et al. [31], and Stolyar [40] propose a scheme where messages are sent from a server to the dispatcher whenever the server becomes idle, so that the dispatcher can keep track of the set of idle servers in real time. Then, an arriving job is to be sent to an empty queue (if there is one) or to a queue chosen uniformly at random (if all queues are non-empty). This policy requires at most λn messages per unit of time and exactly n bits of memory (one bit for each queue, indicating whether it is empty or not). Stolyar [40] has shown that when n goes to infinity, the expected queueing delay vanishes. This policy is symmetric. It has a vanishing delay and a linear message rate, but uses superlogarithmic memory, consistent with Theorem 4.1.1.

Persistent Idle (PI). In order to ensure a full capacity region, even for small values of n, Atar et al. [5] propose a variation of JIQ where, when there are no idle servers available, jobs are sent to the last server that was idle. In [5] the authors showed that, by avoiding the dispatching of jobs uniformly at random, this policy is always stable as long as the combined processing power of all servers can handle the incoming workload (regardless of the disparity in the processing rates of the individual servers).

As with the JIQ policy, a symmetric implementation of PI would require at least n bits of memory, and a message rate of the order of n.

Idle-One-First (I1F). In order to obtain good delay performance, even in a heavy-traffic regime such as the Non-Degenerate Slowdown Regime (NDS), Gupta and Walton [22] propose another variation of JIQ where the dispatcher not only keeps track of which server is idle, but also of which server has only one job in its queue. Note that, even though it has to keep track of servers with both zero and one jobs in their queues, this policy requires at most λn messages per unit of time (at most one per departure of a job). Furthermore, a memory with at least $\lceil n \log_2(3) \rceil$ bits is required, since each queue can have three states: empty, has one job, or has two or more jobs, which leads to 3^n different states that need to be stored in memory. This policy is symmetric, it has a vanishing delay and a linear message rate, but uses superlogarithmic memory, consistent with Theorem 4.1.1.

Resource Constrained Pull-Based (RCPB). In order to reduce the message rate and the memory usage, we proposed in Chapter 3 a family of dispatching policies, similar to Join-Idle-Queue, where the dispatcher keeps a small list of up to c_n idle servers, and where messages are sent from each idle server to the dispatcher as a Poisson process of rate ν_n. Then, an arriving job is sent to an empty queue (if the dispatcher has the ID of one on its list) or to a queue chosen uniformly at random (if the dispatcher's list is empty). This policy requires $(1-\lambda)\nu_n n$ messages per unit of time and $\lceil c_n \log_2(n) \rceil$ bits of memory (the size of the list, c_n, times the number of bits required to store one ID, $\log_2(n)$ bits). In Chapter 3 we showed that, if we either have a high message rate regime (RCPB-HMess) with $c_n = 1$ and $\nu_n \to \infty$, or a high memory regime (RCPB-HMem) with $c_n \to \infty$ and $\nu_n = \lambda/(1-\lambda)$, the expected queueing delay vanishes as $n \to \infty$. This policy is symmetric, and it only has a vanishing delay when either the message rate is superlinear, or the memory is superlogarithmic. This is consistent with Theorem 4.1.1.

4.2.1 Memory, messages, and queueing delay

We now summarize the resource requirements (memory and message rate) and the asymptotic delay of the policies reviewed in this section that fall within our framework.

Policy	Memory (bits)	Message rate	Limiting delay
Random	0	0	> 0
RR [39]	$\lceil \log_2(n) \rceil$	0	> 0
SQ [45]	0	$2\lambda n^2$	0
SQ(d) [34]	0	$2d\lambda n$	> 0
SQ(d_n) [35]	0	$\omega(n)$	0
SQ(d, b) [33]	$\lceil b \log_2(n) \rceil$	$2d\lambda n$	> 0
LL	0	$2\lambda n^2$	0
LL(d) [25]	0	$2d\lambda n$	> 0
JIQ [40]	n	λn	0
RCPB-HMess	$\lceil \log_2(n) \rceil$	$\omega(n)$	0
RCPB-HMem	$\omega(\log(n))$	λn	0

Note that any one of the above listed policies that achieves vanishing queueing delay falls into one (or both) of the following two categories:

a) Those requiring $\omega(n)$ message rate, namely, SQ, SQ(d_n), and LL.

b) Those requiring $\omega(\log(n))$ bits of memory (JIQ, and RCPB-HMem).

The main result of this chapter effectively establishes this fundamental limitation of symmetric policies.

4.3 Proof of Theorem 4.1.1

Let us fix some n. In the sequel, we will assume that n is large enough whenever needed for certain inequalities to hold. We fix a memory-based policy that satisfies Assumption 4.1.1 (symmetry), with at most n^c memory states, and which results in the process $(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot))$ having at least one invariant probability measure.

Let us fix such an invariant probability measure π_n. We consider the process in steady-state; that is, we assume that $(\mathbf{Q}(0), M(0), Z(0))$ is distributed according to π_n. Accordingly, probabilities $\mathbb{P}(\cdot)$ and expectations $\mathbb{E}[\,\cdot\,]$ encountered in the sequel will always refer to the process in steady-state.

The high-level outline of the proof is as follows. In Subsection 4.3.1 we show that under our symmetry assumption, the dispatcher can give special treatment to at most c servers, which we call *distinguished servers*. The treatment of all other servers, is symmetric, in some appropriate sense.

In Subsection 4.3.2 we consider a sequence of bad events under which, over a certain time interval, there are $c + 1$ consecutive arrivals, no service completions or messages from the servers, and all sampled servers are "busy" with a substantial workload. Then, in Subsection 4.3.3, we show that this sequence of bad events has non-negligible probability.

In Subsection 4.3.4, we develop some further consequences of the symmetry assumption, which we use to constrain the information available to the dispatcher at the time of the $(c+1)$st arrival. Loosely speaking, the idea is that during the interval of interest, the server only has information on c distinguished servers together with (useless) information on some busy servers. This in turn implies (subsection 4.3.5) that at least one of the first $c+1$ arrivals must be dispatched to a server on which no useful information is available, and which therefore has a non-negligible probability of inducing a non-negligible delay, thus completing the proof.

4.3.1 Local limitations of finite memory

We consider the (typical) case where a relatively small number of servers are sampled. We will use the symmetry assumption to show that except for a small set of distinguished servers, of size at most c, all other servers must be treated as indistinguishable.

Proposition 4.3.1. *Let U be a uniform random variable over $[0, 1]$. For all n large enough, for every memory state $m \in \mathcal{M}_n$ and every possible job size $w \in \mathbb{R}_+$, the*

following holds. Consider any vector of servers $\mathbf{s} \in \mathcal{R}_n$ *(and its associated set of servers* $\mathbf{s}^{set}$*) with* $|\mathbf{s}| \in o(n)$*, and any integer* ℓ *with* $|\mathbf{s}| + 1 \leq \ell \leq n$*. Consider the event* $B(m, w; \mathbf{s}, \ell)$ *that exactly* ℓ *servers are sampled and that the first* $|\mathbf{s}|$ *of them are the same as the vector* $\mathbf{s}$*, i.e.,*

$$B(m, w; \mathbf{s}, \ell) \triangleq \big\{|f_1(m, w, U)| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{f_1(m, w, U)_i = \mathbf{s}_i\big\},$$

and assume that the conditional probability measure

$$\mathbb{P}\left(\cdot \;\middle|\; B(m, w; \mathbf{s}, \ell)\right)$$

is well-defined. Then, there exists a unique set $R(m, w, \mathbf{s}, \ell) \subset \mathcal{N}_n \backslash \mathbf{s}^{set}$ *of minimal cardinality such that*

$$\mathbb{P}\left(f_1(m, w, U)_{|\mathbf{s}|+1} = j \;\middle|\; B(m, w; \mathbf{s}, \ell)\right) \tag{4.4}$$

is the same for all $j \notin R(m, w, \mathbf{s}, \ell) \cup \mathbf{s}^{set}$*. Furthermore,* $|R(m, w, \mathbf{s}, \ell)| \leq c$*.*

Remark 4.3.1. With some notational abuse, the measure $\mathbb{P}$ in Proposition 4.3.1 need not correspond to the measure $\mathbb{P}$ that describes the process. We are simply considering probabilities associated with a deterministic function of the uniform random variable U.

Proof. Throughout the proof, we fix a particular memory state m, job size w, vector of servers $\mathbf{s}$ with $|\mathbf{s}| \in o(n)$, and an integer ℓ in the range $|\mathbf{s}| + 1 \leq \ell \leq n$. To simplify notation, we will suppress the dependence on w.

Consider the random vector $\mathbf{S}(m) \triangleq f_1(m, U)$. Let $\mathbf{v}$ be the vector whose components are indexed by j ranging in the set $\left(\mathbf{s}^{set}\right)^c = \mathcal{N}_n \backslash \mathbf{s}^{set}$, and defined for any such j, by

$$\mathbf{v}_j \triangleq \mathbb{P}\left(\mathbf{S}(m)_{|\mathbf{s}|+1} = j \;\middle|\; B(m; \mathbf{s}, \ell)\right).$$

We need to show that for j outside a "small" set, all of the components of $\mathbf{v}$ are equal. Let $z_1, \ldots, z_d$ be the distinct values of $\mathbf{v}_j$, as j ranges over $\left(\mathbf{s}^{set}\right)^c$, and let

$A_\alpha = \{j \in \left(\mathbf{s}^{set}\right)^c \mid \mathbf{v}_j = z_\alpha\}$. The sequence of sets $(A_1, \ldots, A_d)$ provides a partition of $\left(\mathbf{s}^{set}\right)^c$ into equivalence classes, with $\mathbf{v}_j = \mathbf{v}_{j'} = z_\alpha$, for all j, j' in the α-th equivalence class A_α. Let $k_1, \ldots, k_d$ be the cardinalities of the equivalence classes $A_1, \ldots, A_d$. Without the loss of generality, assume that k_d is a largest such cardinality. We define

$$R \triangleq \left\{j \in \left(\mathbf{s}^{set}\right)^c \mid \mathbf{v}_j \neq \mathbf{v}_d\right\} = A_1 \cup \cdots \cup A_{d-1},$$

so that $R^c \cap \left(\mathbf{s}^{set}\right)^c = A_d$. For every $j, j' \in A_d$, we have $\mathbf{v}_j = \mathbf{v}_{j'} = \mathbf{v}_d$, and therefore the condition (4.4) is satisfied by R. Note that by choosing $\mathbf{v}_d$ to be the most common value, we are making the cardinality of the set $R^c \cap \left(\mathbf{s}^{set}\right)^c = \{j \notin \mathbf{s}^{set} \mid \mathbf{v}_j = \mathbf{v}_d\}$ as large as possible, from which it follows that the set $R \cap \left(\mathbf{s}^{set}\right)^c$ is as small as possible, and therefore R, as defined, is indeed a minimal cardinality subset of $\left(\mathbf{s}^{set}\right)^c$ that satisfies (4.4).

We now establish the desired upper bound on the cardinality of R. Let $\Sigma_{\mathbf{s}^{set}}$ be the set of permutations that fix the set $\mathbf{s}^{set}$. Consider an arbitrary permutation $\sigma \in \Sigma_{\mathbf{s}^{set}}$ and let σ_M be a corresponding permutation of the memory states, as defined by Assumption 4.1.1. We let $\mathbf{v}_{\sigma^{-1}}$ be the vector with components $(\mathbf{v}_{\sigma^{-1}})_j = \mathbf{v}_{\sigma^{-1}(j)}$, for $j \notin \mathbf{s}^{set}$. Note that as we vary σ over the set $\Sigma_{\mathbf{s}^{set}}$, $\mathbf{v}_{\sigma^{-1}}$ ranges over all possible permutations of the vector $\mathbf{v}$. We also have, for $j \notin \mathbf{s}^{set}$,

$$\begin{aligned}
(\mathbf{v}_{\sigma^{-1}})_j &= \mathbf{v}_{\sigma^{-1}(j)} \\
&= \mathbb{P}\Big(\mathbf{S}(m)_{|\mathbf{s}|+1} = \sigma^{-1}(j) \;\Big|\; B(m; \mathbf{s}, \ell)\Big) \\
&= \mathbb{P}\left(\mathbf{S}(m)_{|\mathbf{s}|+1} = \sigma^{-1}(j) \;\middle|\; \big\{|\mathbf{S}(m)| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}(m)_i = \mathbf{s}_i\big\}\right) \\
&= \mathbb{P}\left(\sigma\big(\mathbf{S}(m)_{|\mathbf{s}|+1}\big) = j \;\middle|\; \big\{|\mathbf{S}(m)| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}(m)_i = \mathbf{s}_i\big\}\right) \\
&= \mathbb{P}\left(\sigma\big(\mathbf{S}(m)_{|\mathbf{s}|+1}\big) = j \;\middle|\; \big\{|\sigma(\mathbf{S}(m))| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\sigma(\mathbf{S}(m)_i) = \sigma(\mathbf{s}_i)\big\}\right)
\end{aligned}$$

$$= \mathbb{P}\left(\sigma\big(\mathbf{S}(m)_{|\mathbf{s}|+1}\big) = j \;\middle|\; \big\{|\sigma(\mathbf{S}(m))| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\sigma(\mathbf{S}(m)_i) = \mathbf{s}_i\big\} \right)$$
$$= \mathbb{P}\left(\mathbf{S}\big(\sigma_M(m)\big)_{|\mathbf{s}|+1} = j \;\middle|\; \big\{|\mathbf{S}(\sigma_M(m))| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}(\sigma_M(m))_i = \mathbf{s}_i\big\} \right).$$

Note that in the above expressions, the only random variables are $\mathbf{S}(m)$ and $\mathbf{S}\big(\sigma_M(m)\big)$, while $\mathbf{s}$ is a fixed vector. The next to last equality above holds because σ fixes the elements in the vector $\mathbf{s}$; the last equality follows because the random variables $\sigma\big(\mathbf{S}(m)\big)$ and $\mathbf{S}\big(\sigma_M(m)\big)$ are identically distributed, according to Part 1 of the symmetry Assumption 4.1.1. The equality that was established above implies that $\sigma_M(m)$ completely determines the vector $\mathbf{v}_{\sigma^{-1}}$. As $\sigma \in \Sigma_{\mathbf{s}^{set}}$ changes, $\sigma_M(m)$ can take at most n^c distinct values, due to the assumed bound on the memory size, and this leads to a bound on the number of possible permutations of the vector $\mathbf{v}$:

$$\big|\{\mathbf{v}_{\sigma^{-1}} : \sigma \in \Sigma_{\mathbf{s}^{set}}\}\big| \leq n^c.$$

We now argue that since $\mathbf{v}$ has relatively few distinct permutations, most of its entries $\mathbf{v}_i$ must be equal. Recall the partition of the set $\big(\mathbf{s}^{set}\big)^c$ of indices into equivalence classes, of sizes $k_1, \ldots, k_d$, with k_d being the largest cardinality. Note that there is a one-to-one correspondence between distinct permutations $\mathbf{v}_{\sigma^{-1}}$ of the vector $\mathbf{v}$ and distinct partitions of $\big(\mathbf{s}^{set}\big)^c$ into a sequence of subsets of cardinalities $k_1, \ldots, k_d$, with the value z_α being taken on the α-th subset. It follows that the number of different partitions of S^c into sets with the given cardinalities, which is given by the multinomial coefficient, satisfies

$$\binom{n-|\mathbf{s}|}{k_1!\ k_2!\ \cdots k_d!} = \big|\{\mathbf{v}_{\sigma^{-1}} : \sigma \in \Sigma_{\mathbf{s}^{set}}\}\big| \leq n^c.$$

The number of choices of a k_d-element subset is no larger than the number of partitions. Therefore,

$$\binom{n-|\mathbf{s}|}{k_d} \leq n^c.$$

An elementary calculation (cf. Lemma 4.4.1) implies that when n is large enough, we must have either (i) $k_d \geq n - |\mathbf{s}| - c$ or (ii) $k_d \leq c$. We argue that the second possibility cannot occur. Indeed, if $k_d \leq c$, and since k_d is the largest cardinality, it follows that $k_\alpha \leq c$ for every α. Since $k_1 + \cdots + k_d = n - |\mathbf{s}|$, we obtain that the number of classes, d, is at least $\lceil (n - |\mathbf{s}|)/c \rceil$. When dealing with d different classes, the number of possible partitions is at least $d!$; this can be seen by focusing on the least-indexed entry in each of the d classes and noting that these d entries may appear in an arbitrary order. Since $|\mathbf{s}| \in o(n)$, we have $n - |\mathbf{s}| \geq n/2$ for all n large enough, and putting everything together, we obtain

$$\lceil (n/2c) \rceil ! \leq \left\lceil \frac{n - |\mathbf{s}|}{c} \right\rceil ! \leq n^c.$$

This is clearly impossible when n is large enough, and case (ii) can therefore be eliminated. We conclude that $|A_d| = k_d \geq n - |\mathbf{s}| - c$. Since $|A_1 \cup \cdots \cup A_d| = |\left(\mathbf{s}^{set}\right)^c| = n - |\mathbf{s}|$, it follows that $|R| = |A_1 \cup \cdots \cup A_{d-1}| \leq c$, which is the desired cardinality bound on R.

It should be apparent that any minimal cardinality set R that satisfies (4.4) must be constructed exactly as our set A_d. Thus, non-uniqueness of the set R with the desired properties will arise if and only if there is another subset A_α, with $\alpha \neq d$, with the same maximal cardinality k_d. On the other hand, since $|\mathbf{s}| \in o(n)$, we have $k_d \geq n - |\mathbf{s}| - c > n/2$, when n is large enough. But having two disjoint subsets, A_d and A_α, each of cardinality larger than $n/2$ is impossible, which proves uniqueness. $\square$

Using a similar argument, we can also show that the distribution of the destination of the incoming job is uniform (or zero) outside the set of sampled servers and a set of at most c distinguished servers.

Proposition 4.3.2. *Let V be a uniform random variable over $[0, 1]$. For all n large enough, for every memory state $m \in \mathcal{M}_n$, every vector of indices $\mathbf{s} \in \mathcal{R}_n$ with $|\mathbf{s}| \in o(n)$, every queue vector state $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, and every job size $w \in \mathbb{R}_+$, the following holds. There exists a unique set $R'\big(m, w, \mathbf{s}, \mathbf{q}\big) \subset \mathcal{N}_n \backslash \mathbf{s}^{set}$ of minimal cardinality such*

that

$$\mathbb{P}\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, V\big) = j\Big) = \mathbb{P}\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, V\big) = k\Big),$$

for all $j, k \notin R'\big(m, w, \mathbf{s}, \mathbf{q}\big) \cup \mathbf{s}^{set}$. *Furthermore,* $|R'(m, w, \mathbf{s}, \mathbf{q})| \leq c$.

Proof. The proof is analogous to the proof of the previous proposition. We start by defining a vector $\mathbf{v}$, whose components are again indexed by j ranging in the set $\mathcal{N}_n \backslash \mathbf{s}^{set}$, by

$$\mathbf{v}_j \triangleq \mathbb{P}\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, V\big) = j\Big).$$

Other than this new definition of the vector $\mathbf{v}$, the rest of the proof follows verbatim the one for Proposition 4.3.1. □

4.3.2 A sequence of "bad" events

In this subsection we introduce a sequence of "bad" events that we will be focusing on in order to establish a positive lower bound on the delay.

Recall that T_1^s is the time of the first event of the underlying Poisson process of rate βn that generates the spontaneous messages from the servers. Recall also that we denote by $\mathbf{Q}_{i,1}(t)$ the remaining workload of the job being serviced in server i, at time t, with $\mathbf{Q}_{i,1}(t) = 0$ if no job is present at server i. Let

$$B \triangleq \left\{ i : \sum_{j=1}^{\infty} \mathbf{Q}_{i,j}(0) \geq 2\gamma \right\}, \tag{4.5}$$

which is the set of servers with at least 2γ remaining workload in their queues, and let $N_b = |B|$, where $\gamma \leq 1$ is a small positive constant, independent of n, to be specified later.

Consider the following events:

(i) the first $c+1$ jobs after time 0 are all of size at least 2γ,

$$\mathcal{A}_w \triangleq \{W_1, \ldots, W_{c+1} \geq 2\gamma\};$$

(ii) the first potential spontaneous message occurs after time γ/n, and the $(c+1)$-st arrival occurs before time γ/n,

$$\mathcal{A}_a \triangleq \left\{T_1^s > \frac{\gamma}{n}\right\} \cap \left\{T_{c+1} < \frac{\gamma}{n}\right\};$$

(iii) there are no service completions before time γ/n,

$$\mathcal{A}_s \triangleq \left\{\mathbf{Q}_{i,1}(0) \notin \left(0, \frac{\gamma}{n}\right), \forall\, i\right\};$$

(iv) there are at least γn servers that each have at least 2γ remaining workload at time zero,

$$\mathcal{A}_b \triangleq \left\{N_b \geq \gamma n\right\}.$$

For an interpretation, the event

$$\mathcal{H}_0^+ \triangleq \mathcal{A}_w \cap \mathcal{A}_a \cap \mathcal{A}_s \cap \mathcal{A}_b,$$

corresponds to an unfavorable situation for the dispatcher. This is because, at time zero, the dispatcher's memory contains possibly useful information on at most c distinguished servers (Propositions 4.3.1 and 4.3.2), and has to accommodate $c+1$ arriving jobs by time γ/n. On the other hand, a nontrivial fraction of the servers are busy and will remain so until time γ/n (event $\mathcal{A}_b$), and it is possible that sampling will not reveal any idle servers (as long as the number of sampled servers is not too large). Thus, at least one of the jobs may end up at a busy server, resulting in positive expected delay. In what follows, we go through the just outlined sequence of unfavorable events, and then, in Subsection 4.3.3, we lower bound its probability.

Starting with $\mathcal{H}_0^+$, we define a nested sequence of events, after first introducing some more notation. For $k = 1, \ldots, c+1$, let $\mathbf{S}_k$ be the random (hence denoted by an upper case symbol) vector of servers that are sampled upon the arrival of the k-th

job; its components are denoted by $(\mathbf{S}_k)_i$. For $i = 0, 1, \ldots, |\mathbf{S}_k|$, we let

$$R_{k,i} \triangleq R\Big(M\big(T_k^-\big), W_k, \big((\mathbf{S}_k)_1, \ldots, (\mathbf{S}_k)_{i-1}\big), |\mathbf{S}_k|\Big)$$

be the (random) subset of servers defined in Proposition 4.3.1 (whenever $M\big(T_k^-\big)$, W_k, $\big((\mathbf{S}_k)_1, \ldots, (\mathbf{S}_k)_{i-1}\big)$, and $|\mathbf{S}_k|$ are such that the proposition applies), with the convention that $\big((\mathbf{S}_k)_1, \ldots, (\mathbf{S}_k)_{i-1}\big) = \emptyset$ when $i = 1$. Otherwise, we let $R_{k,i} \triangleq \emptyset$. Furthermore, we define

$$R_k \triangleq \bigcup_{i=1}^{|\mathbf{S}_k|} R_{k,i}.$$

Moreover, let D_k be the destination of the k-th job, and let

$$R'_k \triangleq R'\Big(M\big(T_k^-\big), W_k, \mathbf{S}_k, \mathbf{Q}_{\mathbf{S}_k}(T_k^-)\Big)$$

be the (random) subset of servers defined in Proposition 4.3.2 (whenever $M\big(T_k^-\big)$, W_k, $\mathbf{S}_k$ and $\mathbf{Q}_{\mathbf{S}_k}(T_k^-)$ are such that the proposition applies). Otherwise, we let $R'_k \triangleq \emptyset$. Finally, given a collection of constants $\xi_1, \ldots, \xi_{c+1}$, independent of n and to be determined later, we define a nested sequence of events recursively, by

$$\begin{aligned}
\mathcal{H}_k^- &\triangleq \mathcal{H}_{k-1}^+ \cap \big\{|\mathbf{S}_k| \leq \xi_k\big\},\\
\mathcal{H}_k &\triangleq \mathcal{H}_k^- \cap \big\{(\mathbf{S}_k)_i \in R_{k,i} \cup B,\ \ i = 1, \ldots, |\mathbf{S}_k|\big\},\\
\mathcal{H}_k^+ &\triangleq \mathcal{H}_k \cap \big\{D_k \in \mathbf{S}_k^{set} \cup R'_k \cup B\big\},
\end{aligned} \tag{4.6}$$

for $k = 1, \ldots, c+1$.

4.3.3 Lower bound on the probability of "bad" events

In this subsection, we establish a positive lower bound, valid for all n large enough, for the probability of the event $\mathcal{H}_{c+1}^+$. In order to do this, we will obtain such uniform lower bounds for the probability of $\mathcal{H}_k^+$, for $k \geq 0$, by induction. We start with the base case.

Lemma 4.3.3. *There exists a constant $\alpha_0^+ > 0$, independent of n, such that*

$$\mathbb{P}(\mathcal{H}_0^+) \geq \alpha_0^+.$$

Proof. Note that the event $\mathcal{A}_a$ only depends on the processes of arrivals and spontaneous messages after time zero, $\mathcal{A}_w$ only depends on the i.i.d. workloads $W_1, \ldots, W_{c+1}$, and $\mathcal{A}_s \cap \mathcal{A}_b$ only depends on the initial queue length vector $\mathbf{Q}(0)$. It follows that

$$\mathbb{P}(\mathcal{H}_0^+) = \mathbb{P}(\mathcal{A}_a)\mathbb{P}(\mathcal{A}_w)\mathbb{P}(\mathcal{A}_s \cap \mathcal{A}_b).$$

We will now lower bound each of these probabilities.

Note that $\mathbb{P}(\mathcal{A}_a)$ is the intersection of two independent events. The first is the event that the first arrival in a Poisson process with rate βn happens after time γ/n, or equivalently, it is the event that the first arrival of a Poisson process of rate β happens after time γ, which has positive probability that does not depend on n. The second is the event that $c+1$ arrivals of the delayed renewal process $A_n(t)$ occur before time γ/n, i.e., the event that $T_{c+1} < \gamma/n$. Since the process $(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot))$ is stationary, the first arrival time (T_1) is distributed according to the residual time of typical inter-arrival times. In particular, if F is the cumulative distribution function of typical inter-arrival times of the arrival process $A_n(\cdot)$ (which have mean $1/\lambda n$), the well-known formula for the distribution of residual times gives

$$\begin{aligned}\mathbb{P}\left(T_1 < \frac{\gamma}{n(c+1)}\right) &= \lambda n \int_0^{\frac{\gamma}{n(c+1)}} \Big(1 - F(u)\Big) du \\ &= \lambda \int_0^{\frac{\gamma}{c+1}} \left(1 - F\left(\frac{v}{n}\right)\right) dv.\end{aligned}$$

Recall that Assumption 4.1.2 states that $1 - F(v/n) \geq \delta_v > 0$, for all $v > 0$ sufficiently

small, and for all n. As a result, we have

$$\lambda \int_0^{\frac{\gamma}{c+1}} \left(1 - F\left(\frac{v}{n}\right)\right) dv \geq \lambda \int_0^{\frac{\gamma}{c+1}} \left(1 - F\left(\frac{\gamma}{n(c+1)}\right)\right) dv \tag{4.7}$$

$$= \frac{\lambda\gamma}{c+1} \delta_{\frac{\gamma}{c+1}}, \tag{4.8}$$

for all γ sufficiently small. On the other hand, for $k = 2, \ldots, c+1$, Assumption 4.1.2 also implies that

$$\mathbb{P}\left(T_k - T_{k-1} \leq \frac{\gamma}{n(c+1)}\right) \geq \delta_{\frac{\gamma}{(c+1)}}.$$

Combining this with Equation (4.8), and using the fact that the first arrival time and the subsequent inter-arrival times are independent, we obtain

$$\begin{aligned}
\mathbb{P}\left(T_{c+1} < \frac{\gamma}{n}\right) &\geq \mathbb{P}\left(\left\{T_1 < \frac{\gamma}{n(c+1)}\right\} \cap \bigcap_{k=2}^{c+1} \left\{T_k - T_{k-1} \leq \frac{\gamma}{n(c+1)}\right\}\right) \\
&\geq \mathbb{P}\left(T_1 < \frac{\gamma}{n(c+1)}\right) \prod_{k=2}^{c+1} \mathbb{P}\left(T_k - T_{k-1} \leq \frac{\gamma}{n(c+1)}\right) \\
&\geq \frac{\lambda\gamma}{c+1} \left(\delta_{\frac{\gamma}{c+1}}\right)^{c+1},
\end{aligned}$$

which is a positive constant independent from n.

We also have

$$\begin{aligned}
\mathbb{P}(\mathcal{A}_w) &= \prod_{i=1}^{c+1} \mathbb{P}(W_i \geq 2\gamma) \\
&= \mathbb{P}(W_i \geq 2\gamma)^{c+1},
\end{aligned}$$

which is independent of n, and positive for γ small enough.

We now consider the event $\mathcal{A}_s$. If $\mathcal{A}_s^c$ holds, then there exists a server i such that $0 < \mathbf{Q}_{i,1}(0) \leq \gamma/n$, and thus we have a job departure during $(0, \frac{\gamma}{n}]$. Let X be the number of service completions during $(0, \frac{\gamma}{n}]$. The occurrence of $\mathcal{A}_s^c$ implies $X \geq 1$. Furthermore, the expected number of service completions in steady-state during any

fixed interval must be equal to the expected number of arrivals, so that

$$\mathbb{P}(\mathcal{A}_s^c) \leq \mathbb{E}[X] = (n\lambda)\frac{\gamma}{n} = \lambda\gamma. \tag{4.9}$$

We now consider the event $\mathcal{A}_b$. Recall that

$$N_b = \left| \left\{ i : \sum_{j=1}^{\infty} \mathbf{Q}_{i,j}(0) \geq 2\gamma \right\} \right|.$$

Let

$$N_I = \left| \left\{ i : \sum_{j=1}^{\infty} \mathbf{Q}_{i,j}(0) = 0 \right\} \right|,$$

and

$$N_d = \left| \left\{ i : 0 < \sum_{j=1}^{\infty} \mathbf{Q}_{i,j}(0) < 2\gamma \right\} \right|.$$

Then, $n = N_b + N_I + N_d$. Furthermore, all servers with $0 < \sum_{j=1}^{\infty} \mathbf{Q}_{i,j}(0) < 2\gamma$ will have a departure in $(0, 2\gamma)$. Let Y be the number of departures (service completions) during $(0, 2\gamma)$. Then, $Y \geq N_d$. We use once more that the expected number of service completions in steady-state during any fixed interval must be equal to the expected number of arrivals, to obtain

$$n\lambda 2\gamma = \mathbb{E}[Y] \geq \mathbb{E}[N_d].$$

Furthermore, by applying Little's law to the number of busy servers, in steady-state, we obtain

$$\mathbb{E}[N_I] = (1 - \lambda)n.$$

Hence

$$\mathbb{E}[N_b] = n - \mathbb{E}[N_I] - \mathbb{E}[N_d] \geq n(\lambda - 2\lambda\gamma).$$

On the other hand, we have

$$\begin{aligned}\mathbb{E}[N_b] &\leq \mathbb{P}(N_b \leq \gamma n)\gamma n + \mathbb{P}(N_b > \gamma n)n \\ &\leq \gamma n + \mathbb{P}(N_b \geq \gamma n)n \\ &= \gamma n + \mathbb{P}(\mathcal{A}_b)n.\end{aligned}$$

Combining these last two inequalities, we obtain

$$\mathbb{P}(\mathcal{A}_b) \geq \lambda - 2\lambda\gamma - \gamma. \tag{4.10}$$

Finally, using equations (4.9) and (4.10), we have

$$\begin{aligned}\mathbb{P}(\mathcal{A}_s \cap \mathcal{A}_b) &= \mathbb{P}(\mathcal{A}_b) - \mathbb{P}(\mathcal{A}_b \cap \mathcal{A}_s^c) \\ &\geq \mathbb{P}(\mathcal{A}_b) - \mathbb{P}(\mathcal{A}_s^c) \\ &\geq \lambda - 2\lambda\gamma - \gamma - \gamma\lambda,\end{aligned}$$

which is a positive constant if γ is chosen small enough. □

We now carry out the inductive step, from $k-1$ to k, in a sequence of three lemmas. We make the induction hypothesis that there exists a positive constant α_{k-1}^+ such that $\mathbb{P}(\mathcal{H}_{k-1}^+) \geq \alpha_{k-1}^+$, and we sequentially prove that there exist positive constants α_k^-, α_k, and α_k^+ such that $\mathbb{P}(\mathcal{H}_k^-) \geq \alpha_k^-$ (Lemma 4.3.4), $\mathbb{P}(\mathcal{H}_k) \geq \alpha_k$ (Proposition 4.3.5), and $\mathbb{P}(\mathcal{H}_k^+) \geq \alpha_k^+$ (Lemma 4.3.7).

Lemma 4.3.4. *Suppose that $\mathbb{P}(\mathcal{H}_{k-1}^+) \geq \alpha_{k-1}^+ > 0$ and that the constant ξ_k is chosen to be large enough. Then, there exists a constant $\alpha_k^- > 0$, such that for all n large enough, we have $\mathbb{P}(\mathcal{H}_k^-) \geq \alpha_k^-$.*

Proof. First, recall our assumption that the average message rate (cf. Equation (4.2)) is upper bounded by αn in expectation. Therefore,

$$\mathbb{E}\left[\limsup_{t\to\infty} \frac{1}{t} \sum_{j=1}^{A_n(t)} 2|\mathbf{S}_j|\right] \leq \alpha n,$$

where $A_n(t)$ is the number of arrivals until time t. By Fatou's lemma, we also have

$$\limsup_{t\to\infty} \mathbb{E}\left[\frac{1}{t}\sum_{j=1}^{A_n(t)} 2|\mathbf{S}_j|\right] \leq \alpha n.$$

Recall that the process $(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot))$ is stationary. Then, since the sampled vectors are a deterministic function of the state, and i.i.d. randomization variables, the point process of arrivals with the sampled vectors as marks, is also stationary. As a result, the expression

$$\mathbb{E}\left[\frac{1}{t}\sum_{j=1}^{A_n(t)} 2|\mathbf{S}_j|\right]$$

is independent from t (see Equation (1.2.9) of [7]). In particular, for $t = \gamma/n$, we have that

$$\mathbb{E}\left[\frac{1}{\gamma}\sum_{j=1}^{A_n\left(\frac{\gamma}{n}\right)} 2|\mathbf{S}_j|\right] \leq \alpha. \tag{4.11}$$

Moreover, since $k \leq c+1$, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^{A_n\left(\frac{\gamma}{n}\right)} |\mathbf{S}_j|\right] &\geq \mathbb{E}\left[\sum_{j=1}^{A_n\left(\frac{\gamma}{n}\right)} |\mathbf{S}_j| \,\middle|\, A_n\left(\frac{\gamma}{n}\right) \geq c+1\right] \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \\ &\geq \mathbb{E}\left[|\mathbf{S}_k| \,\middle|\, A_n\left(\frac{\gamma}{n}\right) \geq c+1\right] \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right).\end{aligned}$$

Combining this with Equation (4.11), we obtain

$$\mathbb{E}\left[\frac{2}{\gamma}|\mathbf{S}_k| \,\middle|\, A_n\left(\frac{\gamma}{n}\right) \geq c+1\right] \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \leq \alpha.$$

This yields the upper bound

$$\mathbb{E}\left[|\mathbf{S}_k| \,\middle|\, A_n\left(\frac{\gamma}{n}\right) \geq c+1\right] \leq \frac{\alpha\gamma}{2\mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right)}. \tag{4.12}$$

On the other hand, using the fact that $\mathcal{H}_{k-1}^{+} \subset \{A_n(\gamma/n) \geq c+1\}$, we have

$$\begin{aligned}
\mathbb{P}\big(\mathcal{H}_k^-\big) &= \mathbb{P}\Big(\mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| \leq \xi_k\big\}\Big) \\
&= \mathbb{P}\left(\mathcal{H}_{k-1}^{+} \cap \left\{A_n\left(\frac{\gamma}{n}\right) \geq c+1\right\} \cap \big\{|\mathbf{S}_k| \leq \xi_k\big\}\right) \\
&= \mathbb{P}\left(\mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| \leq \xi_k\big\} \;\middle|\; A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \\
&\geq \mathbb{P}\big(\mathcal{H}_{k-1}^{+}\big) - \mathbb{P}\left(|\mathbf{S}_k| > \xi_k \;\middle|\; A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right). \quad (4.13)
\end{aligned}$$

Furthermore, for any constant $\xi_k > 0$, Markov's inequality implies

$$\mathbb{P}\left(|\mathbf{S}_k| > \xi_k \;\middle|\; A_n\left(\frac{\gamma}{n}\right) \geq c+1\right) \leq \frac{\mathbb{E}\left[\,|\mathbf{S}_k| \;\middle|\; A_n\left(\frac{\gamma}{n}\right) \geq c+1\right]}{\xi_k}, \quad (4.14)$$

which combined with Equation (4.13) yields

$$\mathbb{P}\big(\mathcal{H}_k^-\big) \geq \mathbb{P}\big(\mathcal{H}_{k-1}^{+}\big) - \frac{\mathbb{E}\left[\,|\mathbf{S}_k| \;\middle|\; A_n\left(\frac{\gamma}{n}\right) \geq c+1\right]}{\xi_k} \mathbb{P}\left(A_n\left(\frac{\gamma}{n}\right) \geq c+1\right). \quad (4.15)$$

Applying the inequality (4.12) to the equation above, we obtain

$$\mathbb{P}\big(\mathcal{H}_k^-\big) \geq \mathbb{P}\big(\mathcal{H}_{k-1}^{+}\big) - \frac{\alpha\gamma}{2\xi_k}. \quad (4.16)$$

Finally, combining this with the fact that $\mathbb{P}(\mathcal{H}_{k-1}^{+}) \geq \alpha_{k-1}^{+} > 0$, we have that

$$\mathbb{P}\big(\mathcal{H}_k^-\big) \geq \alpha_{k-1}^{+} - \frac{\alpha\gamma}{2\xi_k} \triangleq \alpha_k^-,$$

which is positive for all ξ_k large enough. □

Proposition 4.3.5. *Suppose that $\mathbb{P}(\mathcal{H}_k^-) \geq \alpha_k^-$, and that the constant ξ_k is chosen large enough. Then, there exists a constant $\alpha_k > 0$, such that for all n large enough, we have $\mathbb{P}(\mathcal{H}_k) \geq \alpha_k$.*

Proof. Recall the definitions

$$\mathcal{H}_k = \mathcal{H}_k^- \cap \big\{(\mathbf{S}_k)_i \in R_{k,i} \cup B, \;\; i = 1, \ldots, |\mathbf{S}_k|\big\},$$

and

$$\mathcal{H}_k^- = \mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| \leq \xi_k\}.$$

For $i = 1, \ldots, |\mathbf{S}_k|$, let us denote

$$H_{k,i} \triangleq \big\{(\mathbf{S}_k)_i \in R_{k,i} \cup B\big\}.$$

Then,

$$\begin{aligned}
\mathbb{P}(\mathcal{H}_k) &= \mathbb{P}\Big(\mathcal{H}_k^- \cap \big\{(\mathbf{S}_k)_i \in R_{k,i} \cup B,\ i = 1, \ldots, |\mathbf{S}_k|\big\}\Big) \\
&= \sum_{\ell} \mathbb{P}\left(\mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\} \cap \bigcap_{i=1}^{\ell} H_{k,i}\right) \\
&= \sum_{\ell} \mathbb{P}\left(\bigcap_{i=1}^{\ell} H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\}\right) \mathbb{P}\big(\mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\}\big) \\
&= \sum_{\ell} \mathbb{P}\big(\mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\}\big) \prod_{i=1}^{\ell} \mathbb{P}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\} \cap \bigcap_{j=1}^{i-1} H_{k,j}\right), \quad (4.17)
\end{aligned}$$

where the sum is over all integers ℓ such that the conditional probabilities above are well-defined. Intuitively, in the last step, we are treating the selection of the random vector $\mathbf{S}_k$ as a sequential selection of its components, which leads us to consider the product of suitable conditional probabilities. The next lemma provides a lower bound for the factors in this product.

Lemma 4.3.6. *For all n large enough, we have*

$$\mathbb{P}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\} \cap \bigcap_{j=1}^{i-1} H_{k,j}\right) \geq \frac{\gamma}{2},$$

for all $\ell \leq \xi_k$ and $i \leq \ell$ such that the conditional probability above is well-defined.

The idea of the proof of this lemma is that when a next component, $(\mathbf{S}_k)_i$ is chosen, it is either a "distinguished" server, in the set $R_{k,i}$, or else it is a server chosen uniformly outside the set $R_{k,i}$ (cf. Proposition 4.3.1), in which case it has a substan-

tial probability of being a busy server, in the set B. Although the intuition is clear, the formal argument is rather tedious and is deferred to Subsection 4.4.2.

Applying Lemma 4.3.6 to Equation (4.17), and using the fact that $\mathbb{P}\big(\mathcal{H}_k^-\big) \geq \alpha_k^- > 0$, we obtain

$$\begin{aligned}
\mathbb{P}(\mathcal{H}_k) &\geq \sum_{\ell} \mathbb{P}\big(\mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| = \ell\}\big) \left(\frac{\gamma}{2}\right)^{\ell} \\
&\geq \mathbb{P}\big(\mathcal{H}_{k-1}^+ \cap \{|\mathbf{S}_k| \leq \xi_k\}\big) \left(\frac{\gamma}{2}\right)^{\xi_k}, \\
&= \mathbb{P}\big(\mathcal{H}_k^-\big) \left(\frac{\gamma}{2}\right)^{\xi_k} \\
&\geq \alpha_k^- \left(\frac{\gamma}{2}\right)^{\xi_k} \triangleq \alpha_k > 0,
\end{aligned}$$

for all n large enough. □

Lemma 4.3.7. *Suppose that $\mathbb{P}(\mathcal{H}_k) \geq \alpha_k$. Then, there exist a constant $\alpha_k^+ > 0$, such that for all n large enough, we have $\mathbb{P}(\mathcal{H}_k^+) \geq \alpha_k^+$.*

The proof is similar to the proof of Proposition 4.3.5 but with $\xi_k = 1$, and it is omitted. Intuitively, choosing the destination of a job has the same statistical properties as choosing one more server to sample, which brings us back to the setting of Proposition 4.3.5.

This concludes the induction step. It follows that there exists a constant $\alpha_{c+1}^+ > 0$, which is independent of n, and such that $\mathbb{P}(\mathcal{H}_{c+1}^+) \geq \alpha_{c+1}^+$.

4.3.4 Upper bound on the number of useful distinguished servers

Let us provide some intuition on what comes next. The dispatcher initially may treat in a non-typical manner the servers in an initial set of at most c distinguished servers. As servers get sampled, the dispatcher acquires and possibly stores information about other servers. Ultimately, at the time of the $(c+1)$-st arrival, the dispatcher may have acquired information and therefore treat in a special manner (i.e., asymmetrically)

the servers in the set

$$\overline{R} \triangleq \bigcup_{k=1}^{c+1} (R_k \cup R'_k)\,, \tag{4.18}$$

Recall that, for $k = 1, \ldots, c+1$, we have

$$R_k = \bigcup_{i=1}^{|\mathbf{S}_k|} R_{k,i}, \tag{4.19}$$

where each of the sets in the union has cardinality at most c, by Proposition 4.3.1. Furthermore, for $k = 1, \ldots, c+1$, the cardinality of R'_k is also at most c, by Proposition 4.3.2. It follows that

$$\left|\overline{R}\right| \leq c \sum_{k=1}^{c+1} \left(1 + |\mathbf{S}_k|\right). \tag{4.20}$$

If we are to rely solely on this upper bound, the size of $\overline{R}$ can be larger than $c+1$, and it is possible in principle that the knowledge of so many "distinguished" servers (in the set $\overline{R}$) is enough for the dispatcher to identify $c+1$ idle servers to which to route the first $c+1$ jobs. On the other hand, under the event $\mathcal{H}^+_{c+1}$, all new information comes from servers that are "busy" (in the set B), and hence cannot be useful for the dispatching decisions. The next proposition states that for every sample path $\omega \in \mathcal{H}^+_{c+1}$, the set of idle (and therefore, potentially useful) servers on which information is available, namely, the set $\overline{R}\backslash B$, has cardinality of at most c.

Proposition 4.3.8. *The event $\mathcal{H}^+_{c+1}$ implies the event $\left|\overline{R}\backslash B\right| \leq c$.*

Proof. Let us fix a realization $\omega \in \mathcal{H}^+_{c+1}$. We will upper bound the number of distinct images of the set $\overline{R}\backslash B$ under permutations of the set $\mathcal{N}_n$ of servers, which will lead to an upper bound on the cardinality of the set itself. In order to simplify notation, we will suppress the notational dependence on ω of all random variables for the rest of this proof.

We introduce a subset of the set of all possible permutations of $\mathcal{N}_n$, with this subset being rich enough to lead to the desired bound. Towards this goal, we define

the set

$$F \triangleq \bigcup_{k=1}^{c+1} \left(\bigcup_{i=1}^{|\mathbf{S}_k|} \Big[\big\{ (\mathbf{S}_k)_i \big\} \backslash R_{k,i} \Big] \cup \Big[\{D_k\} \backslash \big(R'_k \cup \mathbf{S}_k^{set} \big) \Big] \right). \tag{4.21}$$

This is the set of servers that were sampled, or that were chosen as the destination for a job, which were not in the distinguished sets $R_{k,i}$, or $R'_k \cup \mathbf{S}_k^{set}$, respectively.

Using our assumption $\omega \in \mathcal{H}_{c+1}^+$ and the definition of $\mathcal{H}_{c+1}^+$, we have

$$\bigcup_{k=1}^{c+1} \bigcup_{i=1}^{|\mathbf{S}_k|} \big\{ (\mathbf{S}_k)_i \big\} \backslash R_{k,i} \subset B, \quad \text{and} \quad \bigcup_{k=1}^{c+1} \{D_k\} \backslash \big(R'_k \cup \mathbf{S}_k^{set} \big) \subset B.$$

As a result, we have $F \subset B$, and thus

$$\big(\overline{R} \backslash B \big) \cap F = \emptyset. \tag{4.22}$$

Let Σ be the set of permutations σ of the server set $\mathcal{N}_n$ that:

(i) preserve the ordering of $\overline{R} \backslash B$ in the sense defined in Section 2.1,

(ii) fix the set $\big(\overline{R} \cap B \big) \cup F$, and

(iii) satisfy $\sigma\big(\overline{R} \backslash B \big) \cap \big(\overline{R} \backslash B \big) = \emptyset$.

Consider two permutations $\sigma, \tau \in \Sigma$ such that $\sigma\big(\overline{R} \backslash B \big) = \tau\big(\overline{R} \backslash B \big)$. Then, the fact that σ and τ both preserve the order of $\overline{R} \backslash B$ implies that $\sigma(i) = \tau(i)$, for all $i \in \overline{R} \backslash B$.

Lemma 4.3.9. *Let $\sigma, \tau \in \Sigma$, and let σ_M and τ_M, respectively, be associated permutations of the memory states as specified in Assumption 4.1.1 (Symmetry). Let $m(0)$ be the initial memory state, at time 0. If $\sigma_M\big(m(0)\big) = \tau_M\big(m(0)\big)$, then $\sigma\big(\overline{R}\big) = \tau\big(\overline{R}\big)$.*

Loosely speaking, Lemma 4.3.9 asserts that for the given sample path, permutations σ, τ in Σ that lead to different sets $\overline{R}$ of distinguished servers must also lead (through σ_M and τ_M) to different initial memory states. The proof is an elementary consequence of our symmetry assumption on the underlying dynamics. However, it is tedious and is deferred to Subsection 4.4.3.

By Lemma 4.3.9, and for $\sigma \in \Sigma$, distinct images $\sigma(\overline{R})$ must correspond to distinct memory states $\sigma_M(m(0))$. Since the number of different memory states is upper bounded by n^c, this implies that

$$\Big|\big\{\sigma(\overline{R}) : \sigma \in \Sigma\big\}\Big| \leq n^c.$$

Furthermore, since every $\sigma \in \Sigma$ fixes the set $\overline{R} \cap B$, we have

$$\Big|\big\{\sigma\big(\overline{R}\backslash B\big) : \sigma \in \Sigma\big\}\Big| = \Big|\big\{\sigma(\overline{R}) : \sigma \in \Sigma\big\}\Big| \leq n^c. \tag{4.23}$$

Recall now that the only restrictions on the image $\sigma\big(\overline{R}\backslash B\big)$ under permutations in $\sigma \in \Sigma$ is that the set $\big(\overline{R} \cap B\big) \cup F$ is fixed, and that $\sigma\big(\overline{R}\backslash B\big) \cap \big(\overline{R}\backslash B\big) = \emptyset$. This implies that $\sigma\big(\overline{R}\backslash B\big)$ can be any set of the same cardinality within $\big(\overline{R} \cup F\big)^c$. It follows that

$$\Big|\big\{\sigma\big(\overline{R}\backslash B\big) : \sigma \in \Sigma\big\}\Big| \geq \binom{n - |\overline{R} \cup F|}{|\overline{R}\backslash B|}. \tag{4.24}$$

Recall also that under the event $\mathcal{H}_{c+1}^{+}$ we must have $\big|\mathbf{S}_k\big| \leq \xi_k$, for $k = 1, \ldots, c+1$. Thus $\big|F\big| \leq \xi_1 + \cdots + \xi_{c+1} + c + 1 \triangleq f$, and using Equation (4.20), $|\overline{R}| \leq c(\xi_1 + \cdots + \xi_{c+1}) + c + 1 \triangleq \theta$. Combining these two upper bounds, we obtain

$$\binom{n - |\overline{R} \cup F|}{|\overline{R}\backslash B|} \geq \binom{n - (f+\theta)}{|\overline{R}\backslash B|}.$$

Combining this with equations (4.23) and (4.24), we obtain the inequality

$$n^c \geq \binom{n - (f+\theta)}{|\overline{R}\backslash B|}. \tag{4.25}$$

Finally, using the bound $\big|\overline{R}\big| \leq \theta$, and applying Lemma 4.4.1, we conclude that in order for this equation to hold for all n large enough, we must have $\big|\overline{R}\backslash B\big| \leq c$. $\square$

4.3.5 Completing the proof

We are now ready to complete the proof, by arguing that at least one of the first $c+1$ arrivals must be sent to a server that is either known to be busy or to a server on which no information is available, and therefore has positive probability of being busy.

Recall that for any fixed sample path in $\mathcal{H}^+_{c+1}$, we have (cf. Equation (4.6))

$$\left\{D_1, \ldots, D_{c+1}\right\} \subset B \cup \bigcup_{k=1}^{c+1} \left(\mathbf{S}_k^{set} \cup R'_k\right).$$

Furthermore the event $\mathcal{H}^+_{c+1}$ implies that $(\mathbf{S}_k)_i \in R_{k,i} \cup B$, for $i = 1, \ldots, |\mathbf{S}_k|$ and $k = 1, \ldots, c+1$. Therefore,

$$\mathbf{S}_k^{set} \subset \bigcup_{i=1}^{|\mathbf{S}_k|} R_{k,i} \cup B = R_k \cup B, \tag{4.26}$$

for $k = 1, \ldots, c+1$. It follows that

$$\begin{aligned}
\left\{D_1, \ldots, D_{c+1}\right\} &\subset B \cup \bigcup_{k=1}^{c+1} \left(\mathbf{S}_k^{set} \cup R'_k\right) && (4.27)\\
&\subset B \cup \bigcup_{k=1}^{c+1} \left(R_k \cup R'_k\right) && (4.28)\\
&= B \cup \overline{R}. && (4.29)
\end{aligned}$$

Moreover, Proposition 4.3.8 states that $\left|\overline{R} \backslash B\right| \leq c$. Thus, either (a) there exists k such that $D_k \in B$, or (b) $D_i \in \overline{R} \backslash B$ for $i = 1, \ldots, c+1$, and hence there exists a pair k, l, with $k < l$, such that $D_k = D_l$. We will now show that in both cases, the queueing delay is at least γ.

Let L_k be the queueing delay of the k-th arrival. Recall that for $i \in B$, we have $\mathbf{Q}_{i,1}(0) > 2\gamma$. Then, for case (a), with $D_k = i \in B$ we have

$$L_k = \left(\mathbf{Q}_{i,1}(0) - T_k\right)^+ \geq 2\gamma - \frac{\gamma}{n} \geq \gamma > 0.$$

On the other hand, for case (b), we have

$$L_l \geq \left[W_k - (T_l - T_k)\right]^+ \geq 2\gamma - \left(\frac{\gamma}{n} - 0\right) \geq \gamma > 0.$$

In both cases, we have

$$\sum_{j=1}^{c+1} L_j \geq \gamma.$$

Since this is true for every sample path in $\mathcal{H}_{c+1}^+$, we obtain

$$\mathbb{E}\left[\sum_{j=1}^{c+1} L_j \,\middle|\, \mathcal{H}_{c+1}^+\right] \geq \gamma. \tag{4.30}$$

Finally, recall that the process $(\mathbf{Q}(t), M(t), Z(t))_{t\geq 0}$ is stationary, with invariant probability measure π_n. Then, setting $t = \gamma/n$ in Equation (4.3), we obtain

$$\begin{aligned}
\mathbb{E}_{\pi_n}^0\left[L_0\right] &= \frac{1}{\lambda\gamma}\mathbb{E}\left[\sum_{j=1}^{A_n\left(\frac{\gamma}{n}\right)} L_j\right] \\
&\geq \frac{1}{\lambda\gamma}\mathbb{E}\left[\sum_{j=1}^{A_n\left(\frac{\gamma}{n}\right)} L_j \,\middle|\, \mathcal{H}_{c+1}^+\right]\mathbb{P}\left(\mathcal{H}_{c+1}^+\right) \\
&\geq \frac{1}{\lambda\gamma}\mathbb{E}\left[\sum_{j=1}^{c+1} L_j \,\middle|\, \mathcal{H}_{c+1}^+\right]\mathbb{P}\left(\mathcal{H}_{c+1}^+\right),
\end{aligned}$$

where the last inequality comes from the fact that $\mathcal{H}_{c+1}^+ \subset \left\{A_n(\gamma/n) \geq c+1\right\}$. Combining this with Equation (4.30) and the fact that $\mathbb{P}\left(\mathcal{H}_{c+1}^+\right) \geq \alpha_{c+1}^+ > 0$, we obtain

$$\mathbb{E}_{\pi_n}^0\left[L_0\right] \geq \frac{\alpha_{c+1}^+}{\lambda} > 0.$$

As the constant in the lower bound does not depend on n, this completes the proof of the theorem.

4.4 Additional proofs

4.4.1 A combinatorial inequality

We record here an elementary fact.

Lemma 4.4.1. *Let us fix positive integer constants a and c. Suppose that b satisfies*

$$\binom{n-a}{b} \leq n^c. \tag{4.31}$$

As long as n is large enough, we must have $b \leq c$ or $b \geq n-a-c$.

Proof. Suppose that $b = c+1$. The quantity $\binom{n-a}{c+1}$ is a polynomial in n of degree $c+1$ and therefore, when n is large, (4.31) cannot hold. In the range $c+1 \leq b \leq (n-a)/2$, the quantity $\binom{n-a}{b}$ increases with b, and hence (4.31) cannot hold either. Using the symmetry of the binomial coefficient, a similar argument is used to exclude the possibility that $(n-a)/2 \leq b \leq n-a-c-1$. □

4.4.2 Proof of Lemma 4.3.6

In order to simplify notation, we introduce the following. For any $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, and $b \in \mathcal{P}(\mathcal{N}_n)$, we define the event

$$\mathcal{A}_{m,w,b} \triangleq \big\{M(T_k^-) = m, B = b, W_k = w\big\},$$

and we let $\mathbb{P}_{m,w,b}$ be the conditional probability measure

$$\mathbb{P}_{m,w,b}(\,\cdot\,) \triangleq \mathbb{P}\big(\,\cdot \mid \mathcal{A}_{m,w,b}\big).$$

Let us fix some $\ell \le \xi_k$ and some $i \le \ell$. We have

$$\mathbb{P}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \{|\mathbf{S}_k| = \ell\} \cap \bigcap_{j=1}^{i-1} H_{k,j} \right) = \int_{m,w,b} \mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1} H_{k,j} \right) \cdot \mathrm{d}\mathbb{P}\left(\mathcal{A}_{m,w,b} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \{|\mathbf{S}_k| = \ell\} \cap \bigcap_{j=1}^{i-1} H_{k,j} \right).$$

Moreover,

$$\mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1} H_{k,j} \right) = \sum_{\mathbf{s}} \mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1} \{(\mathbf{S}_k)_j = \mathbf{s}_j\} \right) \cdot \mathbb{P}_{m,w,b}\left(\bigcap_{j=1}^{i-1} \{(\mathbf{S}_k)_j = \mathbf{s}_j\} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1} H_{k,j} \right),$$

where the sum is over all $(i-1)$-dimensional vectors $\mathbf{s}$ whose components are distinct indices of servers, and such that the conditional probabilities above are well-defined.

It is not hard to see that the desired result follows immediately once we establish the following claim.

Claim 4.4.2. *For all n large enough, we have*

$$\mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^{+} \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1} \{(\mathbf{S}_k)_j = \mathbf{s}_j\} \right) \ge \frac{\gamma}{2}, \tag{4.32}$$

for all $(m, w, b, \mathbf{s})$ such that the conditional probability above is well-defined.

Proof. Let us fix some $(m, w, b, \mathbf{s})$. Since $\mathcal{H}_{k-1}^{+}$ implies $|B| \ge \gamma n$, we have

$$|b| \ge \gamma n. \tag{4.33}$$

On the other hand, recall that

$$H_{k,i} = \{(\mathbf{S}_k)_i \in R_{k,i} \cup B\},$$

where

$$\mathbf{S}_k = f_1\big(M(T_k^-), W_k, U_k\big),$$

and $R_{k,i}$ is equal to the set

$$R\Big(M(T_k^-), W_k, \big((\mathbf{S}_k)_1, \ldots, (\mathbf{S}_k)_{i-1}\big), |\mathbf{S}_k|\Big)$$

defined in Proposition 4.3.1, whenever the proposition applies. Otherwise, we have $R_{k,j} = \emptyset$. In any case, $R_{k,j}$ is a deterministic function of the same random variables. Then, conditioned on $M(T_k^-) = m$, $W_k = w$, $B = b$, $\big((\mathbf{S}_k)_1, \ldots, (\mathbf{S}_k)_{j-1}\big) = \mathbf{s}$, and $|\mathbf{S}_k| = \ell$, we have

$$H_{k,i} = \Big\{\Big(f_1\big(m, w, U_k\big)\Big)_i \in r_{k,i} \cup b\Big\},$$

where $r_{k,i}$ denotes the corresponding realization of the random set $R_{k,i}$. Note that the only randomness left in this event comes from U_k, which is a randomization random variable that is chosen independent from all the events prior to time T_k^-. It follows that $H_{k,i}$ is conditionally independent from $\mathcal{H}_{k-1}^+$, and thus

$$\mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \mathcal{H}_{k-1}^+ \cap \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1}\{(\mathbf{S}_k)_j = \mathbf{s}_j\}\right) = \mathbb{P}_{m,w,b}\left(H_{k,i} \;\middle|\; \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1}\{(\mathbf{S}_k)_j = \mathbf{s}_j\}\right).$$

We now define the event $G_{k,\mathbf{s},i,\ell}$ to be

$$G_{k,\mathbf{s},i,\ell} \triangleq \big\{|\mathbf{S}_k| = \ell\big\} \cap \bigcap_{j=1}^{i-1}\{(\mathbf{S}_k)_j = \mathbf{s}_j\}.$$

We are interested in bounding $\mathbb{P}_{m,w,b}\left(H_{k,i} \mid G_{k,\mathbf{s},i,\ell}\right)$, which we decompose into two

terms:

$$\begin{aligned}\mathbb{P}_{m,w,b}\big(H_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) &= \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in r_{k,i} \cup b \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &= \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &\qquad + \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in b\backslash r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big). \qquad (4.34)\end{aligned}$$

Since the conditional probability measure $\mathbb{P}_{m,w,b}(\,\cdot \mid G_{k,\mathbf{s},i,\ell})$ is well-defined, and since $\ell \leq \xi_k$ and $\xi_l \in o(n)$ for all n large enough, Proposition 4.3.1 applies and yields

$$\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i = s \,\big|\, G_{k,\mathbf{s},i,\ell}\big) = \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i = s' \,\big|\, G_{k,\mathbf{s},i,\ell}\big), \qquad (4.35)$$

for all $s, s' \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\}$. As a result,

$$\begin{aligned}&\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in b\backslash r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &\quad \geq \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in b\backslash(r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\}) \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &\quad = \frac{|b\backslash(r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\})|}{n - |r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\}|} \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\} \,\big|\, G_{k,\mathbf{s},i,\ell}\big).\end{aligned}$$

Moreover, using the facts that $|b| \geq \gamma n$ (Equation (4.33)), $|r_{k,i}| \leq c$ (Proposition 4.3.1), and $i \leq \ell$, we obtain

$$\begin{aligned}&\frac{|b\backslash(r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\})|}{n - |r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\}|} \cdot \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &\qquad \geq \frac{\gamma n - c - \ell}{n} \cdot \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\ &\qquad \geq \frac{\gamma}{2} \cdot \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\} \,\big|\, G_{k,\mathbf{s},i,\ell}\big),\end{aligned}$$

when n is large enough. Finally, since the elements of the vector $\mathbf{S}_k$ are distinct,

$$\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}\} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) = \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big),$$

and therefore

$$\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in b \backslash r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \geq \frac{\gamma}{2}\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big).$$

We now substitute into Equation (4.34), and obtain

$$\begin{aligned}
&\mathbb{P}_{m,w,b}\big(H_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\
&\qquad \geq \mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \in r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) + \frac{\gamma}{2}\mathbb{P}_{m,w,b}\big((\mathbf{S}_k)_i \notin r_{k,i} \,\big|\, G_{k,\mathbf{s},i,\ell}\big) \\
&\qquad \geq \frac{\gamma}{2},
\end{aligned}$$

for all n large enough. □

4.4.3 Proof of Lemma 4.3.9

We first prove a claim about the set-valued functions R and R' introduced in propositions 4.3.1 and 4.3.2, respectively.

Claim 4.4.3. *For every $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, $\mathbf{s} \in \mathcal{R}_n$ with $|\mathbf{s}| \in o(n)$, $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, and for $\ell = |\mathbf{s}|+1, \ldots, n$, and for every permutation σ, we have $R\big(\sigma_M(m), w, \sigma(\mathbf{s}), \ell\big) = \sigma\big(R(m,w,\mathbf{s},\ell)\big)$, and $R'\big(\sigma_M(m), w, \sigma(\mathbf{s}), \mathbf{q}\big) = \sigma\big(R'(m,w,\mathbf{s},\mathbf{q})\big)$.*

Proof. In order to simplify notation, we suppress the dependence on w of the functions R, R', and f_1 throughout the proof of the lemma.

Let U be a uniform random variable over $[0,1]$. For every $m \in \mathcal{M}_n$, we define the random vector $\mathbf{S}(m) = f_1(m, U)$. Recall that $R\big(m, \mathbf{s}, \ell\big) \subset \mathcal{N}_n \backslash \mathbf{s}^{set}$ is the unique set of minimal cardinality such that

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{S}(m)_{|\mathbf{s}|+1} = j \;\middle|\; \big\{|\mathbf{S}(m)| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}(m)_i = \mathbf{s}_i\big\}\right) \\
&\qquad\qquad = \mathbb{P}\left(\mathbf{S}\big(m\big)_{|\mathbf{s}|+1} = j' \;\middle|\; \big\{|\mathbf{S}(m)| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}(m)_i = \mathbf{s}_i\big\}\right),
\end{aligned}$$

for all $j, j' \notin R(m,\mathbf{s},\ell) \cup \mathbf{s}^{set}$. It is not hard to see, e.g., by replacing j, j' in the above

equality by $\sigma^{-1}(j), \sigma^{-1}(j') \notin R(m, \mathbf{s}, \ell) \cup \mathbf{s}^{set}$, that $\sigma\big(R\big(m, \mathbf{s}, \ell\big)\big) \subset \mathcal{N}_n \backslash \sigma(\mathbf{s}^{set})$ is the unique set of minimal cardinality such that

$$\mathbb{P}\left(\sigma\big(\mathbf{S}(m)_{|\mathbf{s}|+1}\big) = j \;\middle|\; \big\{|\sigma(\mathbf{S}(m))| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\sigma(\mathbf{S}(m)_i) = \sigma(\mathbf{s}_i)\big\}\right)$$
$$= \mathbb{P}\left(\sigma\big(\mathbf{S}(m)_{|\mathbf{s}|+1}\big) = j' \;\middle|\; \big\{|\sigma(\mathbf{S}(m))| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\sigma(\mathbf{S}(m)_i) = \sigma(\mathbf{s}_i)\big\}\right),$$

for all $j, j' \notin \sigma\big(R\big(m, \mathbf{s}, \ell\big)\big) \cup \sigma(\mathbf{s}^{set})$. On the other hand, the symmetry assumption states that

$$\sigma\big(\mathbf{S}(m)\big) \stackrel{d}{=} \mathbf{S}\big(\sigma_M(m)\big).$$

Combining the last two equalities we get that $\sigma\big(R\big(m, \mathbf{s}, \ell\big)\big) \subset \mathcal{N}_n \backslash \sigma(\mathbf{s}^{set})$ is the unique set of minimal cardinality such that

$$\mathbb{P}\left(\mathbf{S}\big(\sigma_M(m)\big)_{|\mathbf{s}|+1} = j \;\middle|\; \big\{\big|\mathbf{S}\big(\sigma_M(m)\big)\big| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}\big(\sigma_M(m)\big)_i = \sigma(\mathbf{s}_i)\big\}\right)$$
$$= \mathbb{P}\left(\mathbf{S}\big(\sigma_M(m)\big)_{|\mathbf{s}|+1} = j' \;\middle|\; \big\{\big|\mathbf{S}\big(\sigma_M(m)\big)\big| = \ell\big\} \cap \bigcap_{i=1}^{|\mathbf{s}|} \big\{\mathbf{S}\big(\sigma_M(m)\big)_i = \sigma\big(\mathbf{s}_i\big)\big\}\right),$$

for all $i, j \notin \sigma\big(R\big(m, \mathbf{s}, \ell\big)\big) \cup \sigma(\mathbf{s}^{set})$. However, this is exactly the definition of $R(\sigma_M(m), \sigma(\mathbf{s}), \ell)$ (uniqueness is crucial at this point), so we have

$$\sigma\big(R\big(m, \mathbf{s}, \ell\big)\big) = R\big(\sigma_M(m), \sigma(\mathbf{s}), \ell\big).$$

The proof of $R'\big(\sigma_M(m), \sigma(\mathbf{s}), \mathbf{q}\big) = \sigma\big(R'(m, \mathbf{s}, \mathbf{q})\big)$ is analogous (this time making use of the symmetry of the mapping f_2) and is omitted. □

We continue with the proof of Lemma 4.3.9. Under the event $\mathcal{H}_{c+1}^{+}$, we have $(\mathbf{S}_1)_i \in R_{1,i} \cup B$, for $i = 1, \ldots, |\mathbf{S}_1|$. Applying Claim 4.4.3 and the fact $m(t_1^-) = m(0)$,

which implies that $\sigma_M\big(m(t_1^-)\big) = \tau_M\big(m(t_1^-)\big)$, we obtain

$$\begin{aligned}
\sigma\big(R_{1,1}\big) &= \sigma\Big(R\big(m(t_1^-), w_1, \emptyset, |\mathbf{S}_1|\big)\Big) \\
&= R\Big(\sigma_M\big(m(t_1^-)\big), w_1, \emptyset, |\mathbf{S}_1|\Big) \\
&= R\Big(\tau_M\big(m(t_1^-)\big), w_1, \emptyset, |\mathbf{S}_1|\Big) \\
&= \tau\Big(R\big(m(t_1^-), w_1, \emptyset, |\mathbf{S}_1|\big)\Big) \\
&= \tau\big(R_{1,1}\big). \qquad (4.36)
\end{aligned}$$

Now recall that σ and τ preserve the order of $\overline{R}\backslash B$ and fix $\overline{R} \cap B$, so in particular they preserve the order of $R_{1,1}\backslash B \subset \overline{R}\backslash B$ and fix $R_{1,1} \cap B \subset \overline{R} \cap B$. Combining this with Equation (4.36), we must have $\sigma(i) = \tau(i)$, for all $i \in R_{1,1}$. If $(\mathbf{S}_1)_1 \in R_{1,1}$, this implies that

$$\tau\big((\mathbf{S}_1)_1\big) = \sigma\big((\mathbf{S}_1)_1\big). \qquad (4.37)$$

On the other hand, if $(\mathbf{S}_1)_1$ does not belong to $R_{1,1}$, then, from the definition of F, we must have $(\mathbf{S}_1)_1 \in F$. Since σ and τ fix the set F, we conclude that Equation (4.37) must hold in all cases.

Proceeding inductively, and using the same argument, we obtain

$$\sigma\big(R_{1,i}\big) = \tau\big(R_{1,i}\big), \qquad (4.38)$$

for $i = 1, \ldots, |\mathbf{S}_1|$, and $\sigma(i) = \tau(i)$, for all $i \in \mathbf{S}_1^{set}$. It follows that $\sigma(\mathbf{S}_1) = \tau(\mathbf{S}_1)$. Combining this with the fact that $\sigma_M\big(m(t_1^-)\big) = \tau_M\big(m(t_1^-)\big)$, and applying Claim

4.4.3 twice, we obtain

$$\begin{aligned}\sigma\Big(R_1'\Big) &= \sigma\Big(R'\big(m(t_1^-), w_1, \mathbf{S}_1, \mathbf{q}_{\mathbf{S}_1}(t_1^-)\big)\Big)\\ &= R'\Big(\sigma_M\big(m(t_1^-)\big), w_1, \sigma\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-)\Big)\\ &= R'\Big(\tau_M\big(m(t_1^-)\big), w_1, \tau\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-)\Big)\\ &= \tau\Big(R'\big(m(t_1^-), w_1, \mathbf{S}_1, \mathbf{q}_{\mathbf{S}_1}(t_1^-)\big)\Big)\\ &= \tau\Big(R_1'\Big). \end{aligned} \tag{4.39}$$

Now recall that σ and τ preserve the order of $\overline{R}\backslash B$ and fix $\overline{R}\cap B$, so in particular they preserve the order of $R_1'\backslash B \subset \overline{R}\backslash B$ and fix $R_1'\cap B \subset \overline{R}\cap B$. Combining this with Equation (4.39), we must have $\sigma(i) = \tau(i)$, for all $i \in R_1'$. Furthermore, recall that we also have that $\sigma(i) = \tau(i)$, for all $i \in \mathbf{S}_1^{set}$. If $D_1 \in R_1' \cup \mathbf{S}_1^{set}$, this implies that

$$\sigma\big(D_1\big) = \tau\big(D_1\big). \tag{4.40}$$

On the other hand, if D_1 does not belong to $R_1' \cup \mathbf{S}_1^{set}$, then, from the definition of F, we must have $D_1 \in F$. Since σ and τ fix the set F, we conclude that Equation (4.40) must hold in all cases.

We now consider a memory update. Using the symmetry assumption, we have

$$\begin{aligned}\sigma_M\big(m(t_1)\big) &= \sigma_M\Big(f_3\big(m(t_1^-), w_1, \mathbf{S}_1, \mathbf{q}_{\mathbf{S}_1}(t_1^-), D_1\big)\Big)\\ &= f_3\Big(\sigma_M\big(m(t_1^-)\big), w_1, \sigma\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-), \sigma\big(D_1\big)\Big).\end{aligned}$$

Then, since $\sigma_M(m(t_1^-)) = \tau_M(m(t_1^-))$, $\sigma\big(\mathbf{S}_1\big) = \tau\big(\mathbf{S}_1\big)$, and $\tau\big(D_1\big) = \sigma\big(D_1\big)$, we have

$$\begin{aligned}&f_3\Big(\sigma_M\big(m(t_1^-)\big), w_1, \sigma\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-), \sigma\big(D_1\big)\Big)\\ &\qquad = f_3\Big(\tau_M\big(m(t_1^-)\big), w_1, \tau\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-), \tau\big(D_1\big)\Big).\end{aligned}$$

Using the symmetry assumption once again, we obtain

$$f_3\Big(\tau_M\big(m(t_1^-)\big), w_1, \tau\big(\mathbf{S}_1\big), \mathbf{q}_{\mathbf{S}_1}(t_1^-), \tau\big(D_1\big)\Big) = \tau_M\Big(f_3\big(m(t_1^-), w_1, \mathbf{S}_1, \mathbf{q}_{\mathbf{S}_1}(t_1^-), D_1\big)\Big)$$
$$= \tau_M\big(m(t_1)\big).$$

We conclude that

$$\sigma_M\big(m(t_1)\big) = \tau_M\big(m(t_1)\big).$$

Finally, since the memory states at time t_1 are still equal, we can proceed inductively by applying the same argument to obtain that, for $k = 2, \ldots, c+1$, we have $\sigma\big(R_{k,i}\big) = \tau\big(R_{k,i}\big)$ for $i = 1, \ldots, |\mathbf{S}_k|$, and $\sigma\big(R'_k\big) = \tau\big(R'_k\big)$. It follows that $\sigma\big(\overline{R}\big) = \tau\big(\overline{R}\big)$.

4.5 Conclusions and future work

In this chapter, we showed that when we have a limited amount of memory and a modest budget of messages per unit of time, and under a symmetry assumption, all dispatching policies result in queueing delay that is uniformly bounded away from zero. In particular, this implies that the queueing delay does not vanish as the system size increases.

The main result of this chapter complements those of Chapter 3, in which we showed that if we have a little more of either resource, i.e., if the number of memory bits or the message rate grows faster with n, then there exists a symmetric policy that drives the queueing delay to zero as $n \to \infty$. Consequently, we now have necessary and sufficient conditions on the amount of resources available to a central dispatcher, in order to achieve a vanishing queueing delay as the system size increases.

There are several interesting directions for future research. For example:

(i) All the policies in the literature that achieve a vanishing queueing delay need a message rate at least equal to the arrival rate λn. We conjecture that this is not a necessary condition for a policy to have a vanishing queueing delay, as

long as it has access to the incoming job sizes and the memory is sufficiently large.

(ii) Although the message rate is only constrained through its time average, the memory has a hard bound on its size that always has to be satisfied. It would be interesting to explore whether we obtain the same results by constraining the memory size only through its average.

(iii) In light of the symmetry assumption in Theorem 4.1.1, an immediate open question is whether the result still holds without this assumption. Our proof relies heavily on symmetry and is hard to generalize. However, perhaps (non-symmetric) policies that use the memory to store the beginning and the end of streaks of idle servers could achieve a vanishing queueing delay in the low memory and low message rate regime where symmetric policies cannot do it.

Chapter 5

Stability vs resources tradeoff in heterogeneous systems

While in chapters 3 and 4 we focused on the tradeoff between the resources (local memory and message rate) and the expected queueing delay of a typical job in systems with homogeneous servers, in this chapter we focus on systems where servers have heterogeneous service rates, and study the tradeoffs between the stability region of policies and the resources utilized by them.

More concretely, we start by introducing a simple dispatching policy that has the largest capacity region possible while requiring a memory of size (in bits) logarithmic on the number of servers and a positive (but arbitrarily small) message rate. This establishes sufficient conditions on the amount of resources that are required to implement policies with the largest capacity region.

In order to establish necessary conditions on the amount of resources with the same goal in mind, we introduce a unified framework for dispatching policies (slightly more general than the one introduced in Chapter 4). Then, we leverage the same combinatorial approach developed in Chapter 4 to show that all policies with a memory size (in bits) that is sublogarithmic in the number of servers and with an average message rate that is proportional to the arrival rate have a reduced stability region.

The rest of the chapter is organized as follows. The model and the main results

are presented in Section 5.1. In sections 5.2 and 5.3 we provide the proofs of our main results. Finally, in Section 5.4 we present our conclusions and suggestions for future work.

5.1 Model and main results

In this section we present the specific modeling assumptions, the performance metric of interest, and our main results. In Subsection 5.1.1 we describe the model and our assumptions. In Subsection 5.1.2 we introduce a simple dispatching policy and show that it is always stable. In Subsection 5.1.3 we introduce a unified framework that defines a slightly broader set of dispatching policies than the one presented in Chapter 4. In Subsection 5.1.4 we present our negative result on the instability of resource constrained policies within this set of policies. Finally, in Subsection 5.1.5 we combine the results in this chapter to better understand the tradeoff between resources and stability.

5.1.1 Modeling assumptions and performance metric

We now introduce a refinement of the modeling assumptions for the basic model introduced in Section 2.2. First, throughout this chapter we assume that, for all $i = 1, \ldots, n$, the i-th server has constant service rate $\mu_i > 0$. In order to maintain the same total processing power as in the homogeneous case of chapters 3 and 4, we only allow server processing rates in the set

$$\Sigma_n \triangleq \left\{ \mu \in (0, \infty)^n : \sum_{i=1}^{n} \mu_i = n \right\}. \tag{5.1}$$

On the other hand, jobs arrive to the system as a single renewal process of rate λn (for some fixed $\lambda \in (0, 1)$), and are i.i.d., independent from the arrival process, and have a general distribution with unit mean. Finally, the central dispatcher has to route each incoming job to a queue immediately upon arrival (i.e., jobs cannot be queued at the dispatcher).

As in chapters 3 and 4, the dispatcher has limited information on the state of the queues and on the rate of the servers; it can only rely on a limited amount of local memory and on messages that provide partial information about the state and parameters of the system. These messages (which are assumed to be instantaneous) can be sent from a server to the dispatcher at any time, or from the dispatcher to a server (in the form of queries) at the time of an arrival or at the time of a spontaneous message. Messages from a server can only contain information about the state of its own queue (number of remaining jobs and the remaining workload of each one) and about its processing rate. Within this context, a system designer has the freedom to choose a messaging policy, as well as the rules for updating the memory and for selecting the destination of an incoming job.

Regarding the performance metric, our focus is on the **stability region** of a policy under the arrival rate λ, i.e., the largest subset of server rates $\Gamma_n(\lambda) \subset \Sigma_n$ such that the policy is stable for all $\mu \in \Gamma_n(\lambda)$. We will formalize this definition in Subsection 5.1.4.

5.1.2 Universally stable policy

In this subsection we propose a simple dispatching policy with the largest possible stability region (i.e., with stability region equal to Σ_n, for all $\lambda \in (0, 1)$).

Policy description

For any fixed value of n, the policy that we study operates as follows.

a) **Memory:** The dispatcher maintains a register with the ID of a single server.

b) **Dispatching rule:** Whenever a new job arrives, it is sent to the server whose ID is stored in memory (the server ID in memory does not change at this point).

c) **Spontaneous messages:** Each server sends messages to the dispatcher as an independent Poisson process of rate $\alpha_n > 0$, informing the dispatcher of its

queue length (i.e., of the number of job in its queue or in service). When a message from a server arrives to the dispatcher, the dispatcher stores the ID of this server only if the sender's queue is shorter than the queue of the server that is currently stored in memory. In order to make this comparison, the length of the queue of the currently stored server is obtained by sending a query to it.

Remark 5.1.1. This policy requires only $\lceil \log_2(n) \rceil$ bits of memory, and an arbitrarily small (but positive) average message rate of $3\alpha_n n$.

Main result

Note that when the arrival process is Poisson and the service times are exponential, the behavior of the system under this policy can be modeled as a continuous-time Markov chain $\big(\mathbf{Q}(\cdot), I(\cdot)\big)$, where $\mathbf{Q}(\cdot) = \big(\mathbf{Q}_1(\cdot), \ldots, \mathbf{Q}_n(\cdot)\big)$ is the queue lengths vector, and $I(\cdot)$ is the ID of the server stored in memory. In this setting, the stability of the policy is established with the following result.

Theorem 5.1.1. *If the arrival process is Poisson, and the job sizes are exponential, then the stability region of the policy described above is Σ_n, for any $\lambda \in (0, 1)$.*

The proof is given in Section 5.2.

This result states that, at least in the Markovian case, the stability region of our proposed policy is the whole set of admissible rates Σ_n. Moreover, it implies that $\lceil \log_2(n) \rceil$ bits of memory and an average message rate of $3\alpha_n n$ (which can be arbitrarily small) are sufficient for a policy to be always stable. This is much more economical than the most efficient always stable policy in the literature (the Persistent-Idle policy [5], reviewed in Section 4.2), which requires a memory of size (in bits) of order n, and a message rate also of order n.

Remark 5.1.2. Since the proposed policy requires an arbitrarily small message rate, this policy is most useful for applications where a large stability region and a small communication overhead are preferred. Furthermore, since its operation does not

depend explicitly on the rates of the servers, it would continue to work even if the service rates changed slowly over time, which makes it robust.

5.1.3 Unified framework for dispatching policies

In this subsection we present a unified framework that describes memory-based dispatching policies in systems with heterogeneous servers, which slightly generalizes the one introduced in Chapter 4. As in Chapter 4, let c_n be the number of memory bits available to the dispatcher. We define the corresponding set of memory states to be $\mathcal{M}_n \triangleq \{1, \ldots, 2^{c_n}\}$. Furthermore, we define the set of possible states at a server as the set of nonnegative sequences $\mathcal{Q} \triangleq \mathbb{R}_+^{\mathbb{Z}_+}$, where a sequence specifies the remaining workload of each job in that queue, including the one that is being served. (In particular, an idle server is represented by the zero sequence.) As long as a queue has a finite number of jobs, the queue state is a sequence that has only a finite number of non-zero entries. The reason that we include the workload of the jobs in the state is that we wish to allow for a broad class of policies, that can take into account the remaining workload in the queues. In particular, we allow for information-rich messages that describe the full workload sequence at the server that sends the message. We are interested in the process

$$\mathbf{Q}(\cdot) = \big(\mathbf{Q}_1(\cdot), \ldots, \mathbf{Q}_n(\cdot)\big) = \Big(\big(\mathbf{Q}_{1,j}(\cdot)\big)_{j=1}^{\infty}, \ldots, \big(\mathbf{Q}_{n,j}(\cdot)\big)_{j=1}^{\infty}\Big),$$

which takes values in the set $\mathcal{Q}^n$, and describes the evolution of the workload of each job in each queue. We are also interested in the process $M(\cdot)$ that describes the evolution of the memory state, and in the process $Z(\cdot)$ that describes the remaining time until the next arrival of a job.

Fundamental processes and initial conditions

The processes of interest will be driven by the following common fundamental processes:

1. **Arrival process:** A delayed renewal counting process $A_n(\cdot)$ with rate λn, and event times $\{T_k\}_{k=1}^{\infty}$, defined on a probability space $(\Omega_A, \mathcal{A}_A, \mathbb{P}_A)$.

2. **Spontaneous messages process:** A Poisson counting process $R_n(\cdot)$ with rate βn, and event times $\{T_k^s\}_{k=1}^{\infty}$, defined on a probability space $(\Omega_R, \mathcal{A}_R, \mathbb{P}_R)$.

3. **Job sizes:** A sequence of i.i.d. random variables $\{W_k\}_{k=1}^{\infty}$ with mean one, defined on a probability space $(\Omega_W, \mathcal{A}_W, \mathbb{P}_W)$.

4. **Randomization variables:** Five independent and individually i.i.d. sequences of random variables $\{U_k\}_{k=1}^{\infty}$, $\{V_k\}_{k=1}^{\infty}$, $\{X_k\}_{k=1}^{\infty}$, $\{Y_k\}_{k=1}^{\infty}$, and $\{J_k\}_{k=1}^{\infty}$, uniform on $[0,1]$, defined on a common probability space $(\Omega_U, \mathcal{A}_U, \mathbb{P}_U)$.

5. **Initial conditions:** Random variables $\mathbf{Q}(0)$, $M(0)$, and $Z(0)$, defined on a common probability space $(\Omega_0, \mathcal{A}_0, \mathbb{P}_0)$.

The whole system will be defined on the associated product probability space

$$\big(\Omega_A \times \Omega_R \times \Omega_W \times \Omega_U \times \Omega_0, \mathcal{A}_A \times \mathcal{A}_R \times \mathcal{A}_W \times \mathcal{A}_U \times \mathcal{A}_0, \mathbb{P}_A \times \mathbb{P}_R \times \mathbb{P}_W \times \mathbb{P}_U \times \mathbb{P}_0\big),$$

to be denoted by $(\Omega, \mathcal{A}, \mathbb{P})$. All of the randomness in the system originates from these fundamental processes, and everything else is a deterministic function of them.

A construction of sample paths

We provide a construction of a Markov process $(\mathbf{Q}(\cdot), M(\cdot), Z(\cdot))$, taking values in the set $\mathcal{Q}^n \times \mathcal{M}_n \times \mathbb{R}_+$. The memory process $M(\cdot)$ is piecewise constant, and can only jump at the time of an event. All processes considered will have the càdlàg property (right-continuous with left limits) either by assumption (e.g., the underlying fundamental processes) or by construction.

There are three types of events: job arrivals, spontaneous messages, and service completions. We now describe the sources of these events, and what happens when they occur.

Job arrivals: At the time of the k-th event of the arrival process A_n, which occurs at time T_k and involves a job with size W_k, the following transitions happen sequentially but instantaneously:

1. First, the dispatcher chooses a vector of distinct servers $\mathbf{S}_k$, from which it solicits information about their state, according to

$$\mathbf{S}_k = f_1\Big(M\left(T_k^-\right), W_k, U_k\Big),$$

where $f_1 : \mathcal{M}_n \times \mathbb{R}_+ \times [0,1] \to \mathcal{R}_n$ is a measurable function defined by the policy.

2. Then, messages are sent to the servers in the vector $\mathbf{S}_k$, and the servers respond with messages containing their queue states and their service rates; thus, the information received by the dispatcher are the vectors $\mathbf{Q}_{\mathbf{S}_k}$ and $\mu_{\mathbf{S}_k}$. This results in $2|\mathbf{S}_k|$ messages exchanged. Using this information, the destination of the incoming job is chosen to be

$$D_k = f_2\Big(M\big(T_k^-\big), W_k, \mathbf{S}_k, \mathbf{Q}_{\mathbf{S}_k}\big(T_k^-\big), \mu_{\mathbf{S}_k}, V_k\Big),$$

where $f_2 : \mathcal{M}_n \times \mathbb{R}_+ \times \mathcal{R}_n \times \left(\cup_{i=0}^n \mathcal{Q}^i \times \mathbb{R}_+^i\right) \times [0,1] \to \mathcal{N}_n$ is a measurable function defined by the policy. Note that the destination of a job not only depends on the current memory state, the job size, the vector of queried servers, and the state of their queues, but also on the rates of the queried servers.

3. Finally, the memory state is updated according to

$$M(T_k) = f_3\Big(M\big(T_k^-\big), W_k, \mathbf{S}_k, \mathbf{Q}_{\mathbf{S}_k}\big(T_k^-\big), \mu_{\mathbf{S}_k}, D_k\Big),$$

where $f_3 : \mathcal{M}_n \times \mathbb{R}_+ \times \mathcal{R}_n \times \left(\cup_{j=0}^n \mathcal{Q}^j \times \mathbb{R}_+^j\right) \times \mathcal{N}_n \to \mathcal{M}_n$ is a measurable function defined by the policy. Note that the new memory state is obtained using the same information as for selecting the destination, plus the destination of the job, including the rates of the queried servers.

Spontaneous messages: At the time of the k-th event of the spontaneous message process R_n, which occurs at time T_k^s, the i-th server sends a spontaneous message to the dispatcher if and only if

$$g_1\Big(\mathbf{Q}\big(T_k^s\big), \mu, X_k\Big) = i,$$

where $g_1 : \mathcal{Q}^n \times \mathbb{R}_+^n \times [0,1] \to \{0\} \cup \mathcal{N}_n$ is a measurable function defined by the policy. On the other hand, no message is sent when $g_1\big(\mathbf{Q}(T_k^s), \mu, X_k\big) = 0$. Note that the dependence of g_1 on $\mathbf{Q}$ and μ allows the message rate at each server to depend on all servers' current workloads, and on their rates. This allows for policies that let servers with higher service rates send messages at a higher rate than servers with slower service rates.

When a spontaneous message from server i arrives to the dispatcher, the following transitions happen sequentially but instantaneously:

1. First, the dispatcher chooses a vector of distinct servers $\mathbf{S}_k^s$, from which it solicits information about their state, according to

   $$\mathbf{S}_k^s = g_2\Big(M\left(T_k^-\right), i, \mathbf{Q}_i\big(T_k^s\big), \mu_i, Y_k\Big),$$

   where $g_2 : \mathcal{M}_n \times \mathcal{N}_n \times \mathcal{Q} \times \mathbb{R}_+ \times [0,1] \to \mathcal{R}_n$ is a measurable function defined by the policy. Note that the set of servers that are sampled not only depends on the current memory state but also on the index, queue state, and rate of the server that sent the message.

2. Then, messages are sent to the servers in the vector $\mathbf{S}_k^s$, and the servers respond with messages containing their queue states and their service rates; thus, the information received by the dispatcher are the vectors $\mathbf{Q}_{\mathbf{S}_k^s}$ and $\mu_{\mathbf{S}_k^s}$. This results in $2|\mathbf{S}_k^s|$ messages exchanged. Using this information, the memory is updated to the new memory state

   $$M(T_k^s) = g_3\Big(M\big(T_k^{s-}\big), i, \mathbf{Q}_i\big(T_k^s\big), \mu_i, \mathbf{S}_k^s, \mathbf{Q}_{\mathbf{S}_k^s}\big(T_k^s\big), \mu_{\mathbf{S}_k^s}\Big),$$

where $g_3 : \mathcal{M}_n \times \mathcal{N}_n \times \mathcal{Q} \times \mathbb{R}_+ \times \mathcal{R}_n \times \left(\cup_{j=0}^{n} \mathcal{Q}^j \times \mathbb{R}_+^j \right) \to \mathcal{M}_n$ is a measurable function defined by the policy.

Service completions: Let $\{T_k^d(i)\}_{k=1}^{\infty}$ be the sequence of departure times at the i-th server. At those times, the i-th server sends a message to the dispatcher if and only if

$$h_1\Big(\mathbf{Q}_i\big(T_k^d(i)\big), \mu_i, J_k\Big) = 1,$$

where $h_1 : \mathcal{Q} \times \mathbb{R}_+ \times [0,1] \to \{0,1\}$ is a measurable function defined by the policy. In that case, the memory is updated to the new memory state

$$M\Big(T_k^d(i)\Big) = h_2\Big(M\big(T_k^d(i)^-\big), i, \mathbf{Q}_i\big(T_k^d(i)\big), \mu_i\Big),$$

where $h_2 : \mathcal{M}_n \times \mathcal{N}_n \times \mathcal{Q} \times \mathbb{R}_+ \to \mathcal{M}_n$ is a measurable function defined by the policy. On the other hand, no message is sent when $h_1\big(\mathbf{Q}_i(T_k^d(i)), \mu_i, J_k\big) = 0$.

Remark 5.1.3. Note that this framework generalizes the one in Chapter 4, not only by taking into account the rates of the servers to make decisions, but also by allowing the dispatcher to sample servers whenever a spontaneous message arrives.

We now introduce a symmetry assumption on the policies, which is slightly weaker than the one introduced in Chapter 4.

Assumption 5.1.1. (Weakly symmetric policies.) We assume that the dispatching policy is weakly symmetric, in the following sense. For any given permutation of the servers σ, there exists a corresponding (not necessarily unique) permutation σ_M of the memory states $\mathcal{M}_n$ that satisfies all of the following properties:

1. For every $m \in \mathcal{M}_n$ and $w \in \mathbb{R}_+$, and if U is a uniform random variable on $[0,1]$, then

$$\sigma\Big(f_1(m, w, U)\Big) \stackrel{d}{=} f_1\big(\sigma_M(m), w, U\big),$$

where $\stackrel{d}{=}$ stands for equality in distribution.

2. For every $m \in \mathcal{M}_n$, $w \in \mathbb{R}_+$, $\mathbf{s} \in \mathcal{R}_n$, $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, and $\mu_{\mathbf{s}} \in \mathbb{R}_+^{|\mathbf{s}|}$, and if V is a uniform random variable on $[0,1]$, then

$$\sigma\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, \mu_{\mathbf{s}}, V\big)\Big) \stackrel{d}{=} f_2\big(\sigma_M(m), w, \sigma(\mathbf{s}), \mathbf{q}, \mu_{\mathbf{s}}, V\big).$$

Remark 5.1.4. This assumption prevents any bias for or against a server, unless it is encoded in the memory in a sufficiently detailed way so that the assumption is satisfied. For example, in order to implement (in a weakly symmetric way) the randomized dispatching policy where incoming jobs are sent to a server with a probability proportional to its processing rate, the dispatching probabilities have to be encoded in memory in a sufficiently detailed way as to satisfy the second condition in Assumption 5.1.1.

Remark 5.1.5. Note that Assumption 5.1.1 is weaker than the symmetry assumption (Assumption 4.1.1) introduced in Chapter 4 because it does not impose any restrictions on the function f_3. Otherwise, the conditions imposed on f_1 and f_2 are the same.

In particular, the weakening of the symmetry assumption allows for the implementation of policies using less memory. For example, the Round-Robin policy can be implemented in a weakly symmetric way with only $\lceil \log_2(n) \rceil$ bits of memory (to store the ID of the server that was the last destination of a job), while a symmetric implementation of the same policy requires $\lceil (n+1)\log_2(n) \rceil$ bits of memory (to also keep a list of the IDs stored in memory).

Remark 5.1.6. Note that the policy introduced in Subsection 5.1.2 falls within the class of policies defined by this universal framework, and is also weakly symmetric (i.e., it satisfies Assumption 5.1.1).

5.1.4 Instability of resource constrained policies

Before stating the main result of this subsection, we first define the **average message rate** between the dispatcher and the servers as

$$\limsup_{t\to\infty}\frac{1}{t}\left[\sum_{k=1}^{A_n(t)}2|\mathbf{S}_k|+\sum_{k=1}^{R_n(t)}\Big(1+2|\mathbf{S}_k^s|\Big)\mathbb{1}_{\mathcal{N}_n}\Big(g_1\big(\mathbf{Q}\big(T_k^s\big),\mu,X_k\big)\Big)\right.$$
$$\left.+\sum_{i=1}^{n}\sum_{k:\,T_k^d(i)<t}\mathbb{1}_{\{1\}}\Big(h_1\big(\mathbf{Q}_i\big(T_k^d(i)\big),\mu_i,Y_k\big)\Big)\right]. \quad (5.2)$$

Second, we provide a formal definition of our performance metric: the stability region of a policy. For each n, given a policy and an arrival rate λ, the **stability region** of the policy under the arrival rate λ is the largest subset of server rates $\Gamma_n(\lambda)\subset\Sigma_n$ such that the process $\big(\mathbf{Q}(\cdot),M(\cdot),Z(\cdot)\big)$ is positive Harris recurrent for all server rates in $\Gamma_n(\lambda)$.

We are now ready to state the main result of this subsection. It asserts that within the class of weakly symmetric policies that we consider, and under some upper bounds on the memory size and the message rate, the stability region does not contain all possible rates.

Theorem 5.1.2 (Instability of resource constrained policies)**.** *For any fixed n, and for any constants $\lambda\in(0,1)$ and $\alpha_n>0$, there exists a stability region $\Gamma_n(\lambda,\alpha_n)\subsetneq\Sigma_n$ with the following property. Consider a weakly symmetric memory-based dispatching policy, i.e., that satisfies Assumption 5.1.1, with at most $c_n\in o\big(\log(n)\big)$ bits of memory, and with an average message rate (cf. Equation 5.2) upper bounded by $\alpha_n\in o\big(n^2\big)$ almost surely. Then, for all n large enough, the stability region of the policy under the arrival rate λ is contained in $\Gamma_n(\lambda,\alpha_n)$.*

The proof is given in Section 5.3.

5.1.5 Stability vs resources tradeoff

In this subsection, we summarize the results of this paper. First, recall that Theorem 5.1.1 implies that with at least $\lceil \log_2(n) \rceil$ bits and an arbitrarily small message rate, we can obtain a policy (which is weakly symmetric) that is always stable. Second, Theorem 5.1.2 states that weakly symmetric policies with $o\big(\log(n)\big)$ bits of memory and a message rate of order $o\big(n^2\big)$ cannot be always stable. Finally, note that a policy which sends incoming jobs to each server with a probability proportional to the server's rate can be implemented by querying all servers at the time of each arrival. This requires a message rate of order $\Theta\big(n^2\big)$, and it is always stable. The three regimes are depicted in Figure 5-1.

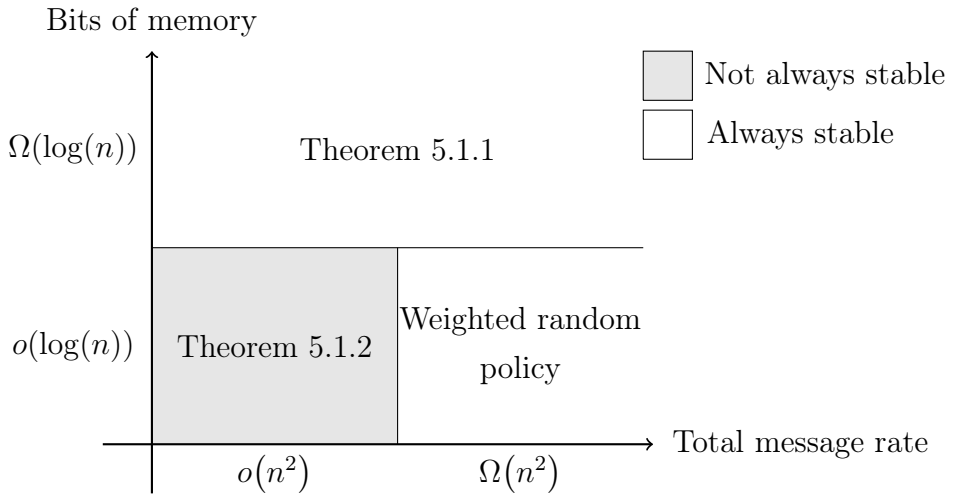

Figure 5-1: Resource requirements for stable policies.

5.2 Proof of Theorem 5.1.1

Let us fix some n, and some arbitrary vector of processing rates in Σ_n. Let μ_{min} and μ_{max} be the smallest and largest processing rates in the chosen vector, respectively. In particular, note that they are positive.

We will use the Foster-Lyapunov criterion to show that the continuous-time Markov chain $\big(\mathbf{Q}(\cdot), I(\cdot)\big)$ is positive recurrent. First, note that this process has state space

$\mathbb{Z}_+^n \times \{1, \ldots, n\}$. Its transition rates, denoted by $r_{\cdot \to \cdot}$, are as follows, where we use $\mathbf{e}_j$ to denote the j-th unit vector in $\mathbb{Z}_+^n$:

1. Since incoming jobs are sent to the queue whose ID is stored in memory, each queue sees arrivals with rate:

$$r_{(\mathbf{q},i)\to(\mathbf{q}+e_j,i)} = \lambda n \mathbb{1}_{\{i\}}(j).$$

2. Transitions due to service completions occur according to the processing rate of each server, and they do not affect the ID stored in memory:

$$r_{(\mathbf{q},i)\to(\mathbf{q}-e_j,i)} = \mu_j \mathbb{1}_{[1,\infty)}\big(\mathbf{q}_j\big).$$

3. Spontaneous messages are sent from each server to the dispatcher at a rate equal to α_n, but the ID stored in memory only changes if the sender of the message has a shorter queue:

$$r_{(\mathbf{q},i)\to(\mathbf{q},j)} = \alpha_n \mathbb{1}_{[0,\mathbf{q}_i-1]}\big(\mathbf{q}_j\big).$$

Note that the Markov process $\big(\mathbf{Q}(\cdot), I(\cdot)\big)$ on the state space $\mathbb{Z}_+^n \times \{1, \ldots, n\}$ is clearly irreducible, with all states reachable from each other. To show positive recurrence, we define the Lyapunov functions

$$\Xi_1(\mathbf{q}, i) \triangleq \frac{2\mu_{max}}{\alpha_n}\mathbf{q}_i,$$

$$\Xi_2(\mathbf{q}, i) \triangleq \sum_{j=1}^{n} \mathbf{q}_j^2,$$

and

$$\Xi(\mathbf{q}, i) \triangleq \Xi_1(\mathbf{q}, i) + \Xi_2(\mathbf{q}, i), \tag{5.3}$$

and note that

$$\sum_{(\mathbf{q}',i')\neq(\mathbf{q},i)} \Xi(\mathbf{q}', i') r_{(\mathbf{q},i)\to(\mathbf{q}',i')} < \infty, \qquad \forall\, (\mathbf{q}, i) \in \mathbb{Z}_+^n \times \{1, \ldots, n\}.$$

We also define the finite set

$$F_n \triangleq \left\{ (\mathbf{q}, i) \in \mathbb{Z}_+^n \times \{1, \ldots, n\} : \sum_{j=1}^{n} \mathbf{q}_j < \frac{\lambda n \left(1 + \frac{2\mu_{max}}{\alpha_n}\right) + n + 1}{2 \min\{1 - \lambda, \mu_{min}\}} \right\}. \tag{5.4}$$

For any state $(\mathbf{q}, i)$, we have

$$\begin{aligned}
&\sum_{(\mathbf{q}', i') \in \mathbb{Z}_+^n \times \{1,\ldots,n\}} \Big[\Xi_1(\mathbf{q}', i') - \Xi_1(\mathbf{q}, i)\Big] r_{(\mathbf{q},i)\to(\mathbf{q}',i')} \\
&\qquad = \lambda n \left(\frac{2\mu_{max}}{\alpha_n}\right) - \sum_{j=1}^{n} 2\mu_{max} \big(\mathbf{q}_i - \mathbf{q}_j\big)^+ - \frac{2\mu_{max}}{\alpha_n} \mu_i \mathbb{1}_{[1,\infty)}\big(\mathbf{q}_i\big) \\
&\qquad \leq \lambda n \left(\frac{2\mu_{max}}{\alpha_n}\right) - \sum_{j=1}^{n} 2\mu_{max} \big(\mathbf{q}_i - \mathbf{q}_j\big)^+, \qquad (5.5)
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{(\mathbf{q}', i') \in \mathbb{Z}_+^n \times \{1,\ldots,n\}} \Big[\Xi_2(\mathbf{q}', i') - \Xi_2(\mathbf{q}, i)\Big] r_{(\mathbf{q},i)\to(\mathbf{q}',i')} \\
&\qquad = \lambda n \left(2\mathbf{q}_i + 1\right) - \sum_{j=1}^{n} \mu_j \big(2\mathbf{q}_j - 1\big) \mathbb{1}_{[1,\infty)}\big(\mathbf{q}_j\big) \\
&\qquad = \lambda n \left(2\mathbf{q}_i + 1\right) + \sum_{j=1}^{n} \mu_j \mathbb{1}_{[1,\infty)}\big(\mathbf{q}_j\big) - 2\sum_{j=1}^{n} \mu_j \mathbf{q}_j \\
&\qquad \leq \lambda n \left(2\mathbf{q}_i + 1\right) + n - 2\sum_{j=1}^{n} \mu_j \mathbf{q}_j, \qquad (5.6)
\end{aligned}$$

where in the last inequality we used that the vector of server rates μ is in Σ_n, which means that

$$\sum_{j=1}^{n} \mu_j = n. \tag{5.7}$$

Combining equations (5.3), (5.5), and (5.6), for any state $(\mathbf{q}, i) \notin F_n$, we have

$$
\begin{aligned}
&\sum_{(\mathbf{q}',i')\in\mathbb{Z}_+^n\times\{1,\dots,n\}} \Big[\Xi(\mathbf{q}',i') - \Xi(\mathbf{q},i)\Big] r_{(\mathbf{q},i)\to(\mathbf{q}',i')} \\
&\qquad \le \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n + 2\lambda n\mathbf{q}_i - 2\sum_{j=1}^{n}\mu_j\mathbf{q}_j + \mu_{max}(\mathbf{q}_i-\mathbf{q}_j)^+ \\
&\qquad \le \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n + 2\lambda n\mathbf{q}_i - 2\sum_{j=1}^{n}\mu_j\Big[\mathbf{q}_j + (\mathbf{q}_i-\mathbf{q}_j)^+\Big] \\
&\qquad = \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n + 2\lambda n\mathbf{q}_i - 2\sum_{j=1}^{n}\mu_j\max\big\{\mathbf{q}_i,\,\mathbf{q}_j\big\} \\
&\qquad = \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n + 2\lambda n\mathbf{q}_i - 2\sum_{j=1}^{n}\mu_j\Big[\mathbf{q}_i + (\mathbf{q}_j-\mathbf{q}_i)^+\Big] \\
&\qquad = \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n + 2\lambda)n\mathbf{q}_i - 2\mathbf{q}_i\sum_{j=1}^{n}\mu_j - 2\sum_{j=1}^{n}\mu_j(\mathbf{q}_j-\mathbf{q}_i)^+ \\
&\qquad \overset{(*)}{=} \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n - 2(1-\lambda)n\mathbf{q}_i - 2\sum_{j=1}^{n}\mu_j(\mathbf{q}_j-\mathbf{q}_i)^+ \\
&\qquad \le \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n - 2(1-\lambda)n\mathbf{q}_i - 2\mu_{min}\sum_{j=1}^{n}(\mathbf{q}_j-\mathbf{q}_i)^+ \\
&\qquad \le \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n - 2\min\{1-\lambda,\mu_{min}\}\sum_{j=1}^{n}\mathbf{q}_i + (\mathbf{q}_j-\mathbf{q}_i)^+ \\
&\qquad = \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n - 2\min\{1-\lambda,\mu_{min}\}\sum_{j=1}^{n}\max\big\{\mathbf{q}_i,\,\mathbf{q}_j\big\} \\
&\qquad \le \lambda n\left(1+\frac{2\mu_{max}}{\alpha_n}\right) + n - 2\min\{1-\lambda,\mu_{min}\}\sum_{j=1}^{n}\mathbf{q}_j \\
&\qquad \le -1,
\end{aligned}
$$

where in equality $(*)$ we used Equation (5.7), and in the last inequality we used the fact that $(\mathbf{q}, i) \notin F_n$ and the definition of the finite set F_n (Equation (5.4)). Then, the Foster-Lyapunov criterion [18] implies the positive recurrence of the Markov chain $\big(\mathbf{Q}(\cdot), I(\cdot)\big)$. Finally, since this is true for all server rates in Σ_n, we conclude that Σ_n is the stability region of the policy.

5.3 Proof of Theorem 5.1.2

Fix λ, and consider a vector of server rates in Σ_n where $\lfloor n/2 \rfloor$ servers have rate $\epsilon_n > 0$. We will show that, for any given λ, and for all ϵ_n small enough, every resource constrained dispatching policy that is weakly symmetric (i.e., that it satisfies Assumption 5.1.1) overloads the slow servers.

The high-level outline of the proof is as follows. In Subsection 5.3.1 we show that under our weak symmetry assumption, the constrain on the number of bits available implies that the dispatcher treats all servers in a symmetric way, in some appropriate sense.

Then, in Subsection 5.3.2 we combine the results obtained in Subsection 5.3.1 with the bound on the average message rate to show that jobs are sent to slow servers (i.e., to servers with service rate ϵ_n) with a positive rate that is bounded away from zero. This implies that the total workload of the servers diverges for all ϵ_n small enough, thus completing the proof.

5.3.1 Local limitations of finite memory

We first note that if there are at most $2^{c_n} \in o(n)$ memory states, then the distribution of the sampled servers is uniform.

Lemma 5.3.1. *Let U be a uniform random variable over $[0,1]$. For all n large enough, for every memory state $m \in \mathcal{M}_n$, for every possible job size $w \in \mathbb{R}_+$, and for any vector of servers $\mathbf{s} \in \mathcal{R}_n$ with $|\mathbf{s}| \in o(n)$, we have*

$$\mathbb{P}\Big(f_1\big(m,w,U\big) = \mathbf{s}\Big) = \mathbb{P}\Big(f_1\big(m,w,U\big) = \sigma(\mathbf{s})\Big),$$

for every permutation σ.

Proof. This is a corollary of Proposition 4.3.1. □

Similarly, we argue that if there are at most $2^{c_n} \in o(n)$ memory states, then the distribution of the destination of the incoming job is uniform (or zero) outside the set of sampled servers.

Lemma 5.3.2. *Let V be a uniform random variable over $[0,1]$. For all n large enough, for every memory state $m \in \mathcal{M}_n$, every vector of indices $\mathbf{s} \in \mathcal{R}_n$ with $|\mathbf{s}| \in o(n)$, every queue vector state $\mathbf{q} \in \mathcal{Q}^{|\mathbf{s}|}$, every rate vector $\mu_{\mathbf{s}} \in \mathbb{R}_+^{|\mathbf{s}|}$, and every job size $w \in \mathbb{R}_+$, we have*

$$\mathbb{P}\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, \mu_{\mathbf{s}}, V\big) = j\Big) = \mathbb{P}\Big(f_2\big(m, w, \mathbf{s}, \mathbf{q}, \mu_{\mathbf{s}}, V\big) = k\Big),$$

for all $j, k \in \mathcal{N}_n \backslash \mathbf{s}^{set}$.

Proof. This is a corollary of Proposition 4.3.2. □

5.3.2 High arrival rate to slow servers

For every $t \geq 0$, let $\mathcal{W}^n(t)$ be the total remaining workload in the system at time t.

Lemma 5.3.3. *For every λ, there exists a constant $a_n(\lambda) > 0$ such that*

$$\liminf_{t\to\infty} \frac{\mathcal{W}^n(t)}{t} \geq \big[a_n(\lambda) - \epsilon_n\big] n, \qquad a.s.,$$

for all n large enough.

Proof. Let $\overline{A}_n(t)$ be the counting process of arrivals with a job size of at least $1/2$, and let us define

$$p_{1/2} \triangleq \mathbb{P}\left(W_1 \geq \frac{1}{2}\right).$$

Since the total arrivals are a renewal process of rate λn, and the job sizes $\{W_k\}_{k=1}^{\infty}$ are i.i.d. with unit mean, then $\overline{A}_n(t)$ is a renewal counting process of rate $\lambda n p_{1/2} > 0$. On the other hand, since the average message rate (cf. Equation 5.2) is upper bounded by αn almost surely, we have

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{k=1}^{\overline{A}_n(t)} 2|\mathbf{S}_k| \leq \alpha_n, \qquad a.s.$$

Combining this with the fact that

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}2\left(\frac{\alpha_n}{\lambda np_{1/2}}\right)\mathbb{1}_{\left\{|\mathbf{S}_k|>\frac{\alpha_n}{\lambda np_{1/2}}\right\}}\leq\limsup_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}2|\mathbf{S}_k|,$$

we obtain

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}\mathbb{1}_{\left\{|\mathbf{S}_k|>\frac{\alpha_n}{\lambda np_{1/2}}\right\}}\leq\frac{\lambda np_{1/2}}{2}.$$

This in turn implies that

$$\begin{aligned}
\liminf_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}\mathbb{1}_{\left\{|\mathbf{S}_k|\leq\frac{\alpha_n}{\lambda np_{1/2}}\right\}} &= \liminf_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}\left(1-\mathbb{1}_{\left\{|\mathbf{S}_k|>\frac{\alpha_n}{\lambda np_{1/2}}\right\}}\right)\\
&= \liminf_{t\to\infty}\frac{\overline{A}_n(t)}{t}+\liminf_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}-\mathbb{1}_{\left\{|\mathbf{S}_k|>\frac{\alpha_n}{\lambda np_{1/2}}\right\}}\\
&= \lambda np_{1/2}-\limsup_{t\to\infty}\frac{1}{t}\sum_{k=1}^{\overline{A}_n(t)}\mathbb{1}_{\left\{|\mathbf{S}_k|>\frac{\alpha_n}{\lambda np_{1/2}}\right\}}\\
&\geq \frac{\lambda np_{1/2}}{2},\qquad a.s. \qquad (5.8)
\end{aligned}$$

Let $N_{\epsilon_n}\subset\mathcal{N}_n$ be the set of servers with service rate ϵ_n, which was assumed to have cardinality $\lfloor n/2\rfloor$. Then, Lemma 5.3.1 implies that

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{S}_k^{set}\subset N_{\epsilon_n}\;\middle|\;|\mathbf{S}_k|\leq\frac{\alpha_n}{\lambda np_{1/2}}\right)\\
&\qquad\geq\frac{\binom{\lfloor n/2\rfloor}{\lfloor\alpha_n/\lambda np_{1/2}\rfloor}}{\binom{n}{\lfloor\alpha_n/\lambda np_{1/2}\rfloor}}\\
&\qquad=\frac{\lfloor n/2\rfloor\big(\lfloor n/2\rfloor-1\big)\cdots\big(\lfloor n/2\rfloor-\lfloor\alpha_n/\lambda np_{1/2}\rfloor+1\big)}{n\big(n-1\big)\cdots\big(n-\lfloor\alpha_n/\lambda np_{1/2}\rfloor+1\big)}\\
&\qquad\geq\left(\frac{1}{3}\right)^{\frac{\alpha_n}{\lambda np_{1/2}}},
\end{aligned}$$

for all $k\geq 1$, and for all n large enough, where in the last inequality we used that

$\alpha_n \in o\big(n^2\big)$. Combining this with Equation (5.8), we obtain

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{k=1}^{\overline{A}_n(t)} \mathbb{1}_{\left\{|\mathbf{S}_k| \leq \frac{\alpha_n}{\lambda n p_{1/2}}, \mathbf{S}_k^{set} \subset N_{\epsilon_n}\right\}} \geq \frac{\lambda n p_{1/2}}{2} \left(\frac{1}{3}\right)^{\frac{\alpha_n}{\lambda n p_{1/2}}}, \tag{5.9}$$

almost surely, for all n large enough. Furthermore, Lemma 5.3.2 implies that

$$\begin{aligned}
\mathbb{P}\left(D_k \in N_{\epsilon_n} \,\middle|\, \mathbf{S}_k^{set} \subset N_{\epsilon_n},\, |\mathbf{S}_k| \leq \frac{\alpha_n}{\lambda n p_{1/2}}\right) & \\
&\geq \mathbb{P}\left(D_k \in N_{\epsilon_n} \,\middle|\, D_k \notin \mathbf{S}_k^{set},\, \mathbf{S}_k^{set} \subset N_{\epsilon_n},\, |\mathbf{S}_k| \leq \frac{\alpha_n}{\lambda n p_{1/2}}\right) \\
&= \frac{\left\lfloor \frac{n}{2} \right\rfloor - \frac{\alpha_n}{\lambda n p_{1/2}}}{n} \\
&\geq \frac{1}{3},
\end{aligned}$$

for all $k \geq 1$, and for all n large enough, where in the last inequality we used that $\alpha_n \in o\big(n^2\big)$. Combining this with Equation (5.9), we obtain

$$\begin{aligned}
\liminf_{t\to\infty} \frac{1}{t} \sum_{k=1}^{\overline{A}_n(t)} \mathbb{1}_{\left\{D_k \in N_{\epsilon_n}\right\}} &\geq \liminf_{t\to\infty} \frac{1}{t} \sum_{k=1}^{\overline{A}_n(t)} \mathbb{1}_{\left\{D_k \in N_{\epsilon_n}, |\mathbf{S}_k| \leq \frac{\alpha_n}{\lambda n p_{1/2}}, \mathbf{S}_k^{set} \subset N_{\epsilon_n}\right\}} \\
&\geq \frac{\lambda n p_{1/2}}{6} \left(\frac{1}{3}\right)^{\frac{\alpha_n}{\lambda n p_{1/2}}}, \qquad a.s.,
\end{aligned}$$

for all n large enough. Note that this is a lower bound on the average rate of arrival of jobs with size at least $1/2$, to the servers with service rate ϵ_n. On the other hand, those servers have a total processing rate of $\epsilon_n \lfloor n/2 \rfloor$ units of workload per unit of time. Then, since the total workload of the system is at least as much as the workload of the servers with rate ϵ_n, we have

$$\begin{aligned}
\liminf_{t\to\infty} \frac{\mathcal{W}^n(t)}{t} &\geq \liminf_{t\to\infty} \frac{1}{t} \sum_{k=1}^{\overline{A}_n(t)} \frac{1}{2} \mathbb{1}_{\left\{D_k \in N_{\epsilon_n}\right\}} - \epsilon_n \left\lfloor \frac{n}{2} \right\rfloor \\
&\geq \left[\frac{\lambda p_{1/2}}{6} \left(\frac{1}{3}\right)^{\frac{\alpha_n}{\lambda n p_{1/2}}} - \epsilon_n\right] n,
\end{aligned}$$

for all n large enough. □

Note that Lemma 5.3.3 implies that, for all n large enough, the total workload of the system increases at least linearly with time when $\lfloor n/2 \rfloor$ of the servers have rate $\epsilon_n < a_n(\lambda)$. Since this is true for every weakly symmetric policy with $o(\log(n))$ bits of memory, and with an average message rate upper bounded by $\alpha_n \in o\big(n^2\big)$ almost surely, it follows that, for all n large enough, the stability region of all such policies are contained in the subset of server rates $\Gamma_n(\lambda, \alpha_n) \subsetneq \Sigma_n$ that excludes the ones where $\lfloor n/2 \rfloor$ of the servers have rate $\epsilon_n < a_n(\lambda)$.

5.4 Conclusions and future work

In this chapter, we proposed a simple but efficient dispatching policy that requires a memory of size (in bits) logarithmic in the number of servers, and an arbitrarily small message rate message rate, and showed that it has the largest possible stability region. The key for the stability properties of this policy is the fact that it never chooses the destination of a job by either random sampling of the servers (like JSQ(d)) or by random dispatching of the job (like JIQ).

On the other hand, we showed that when we have a memory size (in bits) sublogarithmic in the number of servers, and a message rate sublinear in the square of the arrival rate, all weakly symmetric dispatching policies have a sub-optimal stability region. We leave as an open question whether a policy with a memory size (in bits) sublogarithmic in the number of servers and a message rate superlinear in the arrival rate can have the largest possible stability region.

There are several interesting directions for future research. For example:

(i) Policies can have the largest possible stability region using an arbitrarily small message rate, as long as the size of the memory (in bits) is logarithmic in the number of servers. However, their delay performance is not completely understood. For example, the rate of increase of the expected delay as the messaging rate decreases.

(ii) Although a memory of size (in bits) that is logarithmic in the number of servers

is necessary in order to have policies with the largest possible stability region when the message rate is at most proportional to the arrival rate, we conjecture that, if the average message rate is allowed to be superlinear in the arrival rate, then there are dispatching policies that have the largest possible stability region, even with no memory.

Chapter 6

Concluding remarks

This thesis is centered around the role of information in large-scale distributed service systems. Our results demonstrate that with enough resources and the appropriate dispatching policies, we can obtain the same asymptotic performance as in systems with many more resources available.

Some of the open problems that concern specific models have been stated at the end of the corresponding chapters. Thus, we now focus on higher level issues that could provide interesting directions for future research.

Detailed performance metrics. The main performance metric used throughout this thesis is the *expected* delay of a typical job, and for the most part we were interested in whether the queueing delay of a typical job converges to zero or not, when the system size increases. However, it provides no insight as to how fast it converges to zero, which is especially relevant for moderately sized systems. Furthermore, while the expectation of the delay is quite informative, it obscures other properties of the delay that can be equally relevant. For example, in many applications the variance or the tail of the delay might be more relevant than its expectation, especially if there are deadlines or penalties incurred for large differences between delays of different jobs (e.g., in video streaming).

Note that in Chapter 3 we proposed policies that drive the expected queueing delay to zero, and characterized the minimum amount of resources required to do it.

However, the tail of the delay distribution seems to be exponential (worse than the superexponential tails observed in policies such as JSQ(d)). Furthermore, we are not taking into account the speed of convergence in the design or analysis of the policies. We conjecture that the speed of convergence depends critically on the amount of resources available, which would make it an interesting line of future work.

Abandonments in the queues. All jobs in this thesis were considered to have infinite patience, in the sense that the jobs stay in the system until service completion. However, this is not always the case in practice. For example, if there are people waiting to receive service, they might leave the queue if they have to wait for too long. Furthermore, in applications such as video streaming, the frames to be downloaded have deadlines after which they become useless.

While in both applications mentioned above there is patience involved, job/customer behavior can be quite different. For example, customers in a queue might be aware of the length of the queue and of how fast the queue is moving, which could make them abandon the queue sooner or later than with a fixed patience. Since the vast literature on this subject assumes a fixed patience, there is an opportunity to explore the different dynamics that arise from different patience models.

Hierarchical architectures. In this thesis we considered distributed service systems, where all servers operate in parallel and there is a single dispatcher that makes all decisions. However, there are many applications where there are several dispatchers and/or where there are sets of servers in parallel, all of which have to process the jobs before they leave the system.

In this setting, we might still want to study the tradeoff between resources, stability, and delay, or design appropriate replication policies. However, since the dynamics introduced by the sequential nature of multi-stage systems seems to be significantly different from those considered in this thesis, we suspect that one would need a different set of analytical techniques and problem formulations in order to study the same issues.

Bibliography

[1] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen. Parallel randomized load balancing. *Random Structures and Algorithms*, 13(2):159–188, 1998.

[2] R. Aghajani and K. Ramanan. The hydrodynamic limit of a randomized load balancing network. arXiv:1707.02005, 2017.

[3] N. Alon, E. Lubetzky, and O. Gurel-Gurevich. Choice-memory tradeoff in allocations. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009.

[4] S. Asmussen. *Applied Probability and Queues*. Springer, 2003.

[5] R. Atar, I. Keslassy, G. Mendelson, A. Orda, and S. Vargaftik. Persistent-Idle Load-Distribution. Preprint, 2018.

[6] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM Journal on Computing*, 29(1):180–200, 1999.

[7] F. Baccelli and P. Brémaud. *Elements of Queueing Theory*. Springer, 2003.

[8] R. Badonnel and M. Burgess. Dynamic pull-based load balancing for autonomic servers. In *Proceedings of the Network Operations and Management Symposium (NOMS)*, 2008.

[9] I. Benjamini and Y. Makarychev. Balanced allocations: memory performance tradeoffs. *The Annals of Applied Probability*, 22(4):1642–1649, 2012.

[10] D. Bertsimas, D. Gamarnik, and J. N. Tsitsiklis. Performance of multiclass Markovian queueing networks via piecewise linear Lyapunov functions. *The Annals of Applied Probability*, 11(4):1384–1428, 2002.

[11] P. Billingsley. *Convergence of Probability Measures*. Wiley, second edition, 1999.

[12] M. Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems: Theory and Applications*, 30:89–148, 1998.

[13] M. Bramson, Y. Lu, and B. Prabhakar. Asymptotic independence of queues under randomized load balancing. *Queueing Systems*, 71:247–292, 2012.

[14] M. Bramson, Y. Lu, and B. Prabhakar. Decay tails at equilibrium for FIFO join the shortest queue networks. *The Annals of Applied Probability*, 23(5):1841–1878, 2013.

[15] D. Feitelson and M. A. Jette. Improved utilization and responsiveness with gang scheduling. In *Proceedings of the Job Scheduling Strategies for Parallel Processing (IPPS)*, pages 238–261, 1997.

[16] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer-Science, 1988.

[17] S. Foss and A. L. Stolyar. Large-scale Join-Idle-Queue system with general service times. *Journal of Applied Probability*, 54(4):995–1007, 2017.

[18] F. G. Foster. On the stochastic matrices associated with certain queueing processes. *The Annals of Mathematical Statistics*, 24:355–360, 1953.

[19] D. Gamarnik, J. N. Tsitsiklis, and M. Zubeldia. Delay, memory, and messaging tradeoffs in distributed service systems. In *Proceedings of the ACM SIGMETRICS Conference*, 2016.

[20] D. Gamarnik, J. N. Tsitsiklis, and M. Zubeldia. Delay, memory, and messaging tradeoffs in distributed service systems. *Stochastic Systems*, 8(1):45–74, 2018.

[21] D. Gamarnik, J. N. Tsitsiklis, and M. Zubeldia. A lower bound on the queueing delay in resource constrained load balancing. Under revision in *The Annals of Applied Probability*, 2019.

[22] V. Gupta and N. Walton. Load balancing in the non-degenerate slowdown regime. *Operations Research*, 2019.

[23] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, 14(3):502–525, 1982.

[24] M. Harchol-Balter, M. E. Crovella, and C. D. Murta. On choosing a task assignment policy for a distributed server system. *IEEE Journal of Parallel and Distributed Computing*, 59(2):204–228, 1999.

[25] T. Hellemans and B. Van Houdt. On the power-of-d-choices with least loaded server selection. *POMACS*, 2(2), 2018.

[26] P. J. Hunt and T. G. Kurtz. Large loss networks. *Stochastic Processes and their Applications*, 53(2):363–378, 1994.

[27] M. D. Kirszbraun. Uber die zusammenziehende und Lipschitzsche Transformationen. *Fund. Math*, 22:77–108, 1934.

[28] T. G. Kurtz. *Approximation of Population Processes*. Society for Industrial and Applied Mathematics, 1981.

[29] C. Lenzen and R. Wattenhofer. Tight bounds for parallel randomized load balalcing. *Distributed Computing*, pages 1–16, 2014.

[30] S. G. Lobanov and O. G. Smolyanov. Ordinary differential equations in locally convex spaces. *Uspekhi Mat. Nauk*, 49:93–168, 1994.

[31] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg. Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation*, 68(11):1056–1071, Nov. 2011.

[32] M. Mitzenmacher. Analyzing distributed Join-Idle-Queue: A fluid limit approach. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, 2016.

[33] M. Mitzenmacher, B. Prabhakar, and D. Shah. Load balancing with memory. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[34] M. D. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, U.C. Berkeley, 1996.

[35] D. Mukherjee, S. Borst, J. van Leeuwaarden, and P. Whiting. Universality of Power-of-d Load Balancing Schemes. In *Proceedings of the Workshop on Mathematical performance Modeling and Analysis (MAMA)*, 2016.

[36] C. Nair, B. Prabhakar, and D. Shah. The randomness in randomized load balancing. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, pages 912–921, 2001.

[37] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.

[38] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis: Queues, Communications, and Computing*. Chapman & Hall, 1995.

[39] G. D. Stamoulis and J. N. Tsitsiklis. Optimal distributed policies for choosing among multiple servers. In *Proceedings of the 30th Conference on Decision and Control*, pages 815–820, 1991.

[40] A. L. Stolyar. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems: Theory and Applications*, 80(4):341–361, 2015.

[41] A. L. Stolyar. Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers. *Queueing Systems: Theory and Applications*, 85(2), 2017.

[42] J. N. Tsitsiklis and K. Xu. On the power of (even a little) resource pooling. *Stochastic Systems*, 2:1–66, 2012.

[43] M. van der Boor, S. Borst, and J. van Leeuwaarden. Load balancing in large-scale systems with multiple dispatchers. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2017.

[44] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: an asymptotic approach. *Problems of Information Transmission*, 32(1):15–27, 1996.

[45] W. Winston. Optimality of the shortest line discipline. *Applied Probability*, 14:181–189, 1977.

[46] K. Xu and S.-Y. Yun. Reinforcement with fading memories. In *Proceedings of the ACM SIGMETRICS Conference*, 2018.

[47] L. Ying, R. Srikant, and X. Kang. The power of slightly more than one sample in randomized load balancing. In *Proceedings of the IEEE Conference on Computer Communications*, 2015.