Heavy-Traffic Optimal Size- and State-Aware Dispatching

RUNHAN XIE, University of California, Berkeley, USA
ISAAC GROSOF, Carnegie Mellon University, USA and Georgia Institute of Technology, USA
ZIV SCULLY, Cornell University, USA

Dispatching systems, where arriving jobs are immediately assigned to one of multiple queues, are ubiquitous in computer systems and service systems. A natural and practically relevant model is one in which each queue serves jobs in FCFS (First-Come First-Served) order. We consider the case where the dispatcher is *size-aware*, meaning it learns the size (i.e. service time) of each job as it arrives; and *state-aware*, meaning it always knows the amount of work (i.e. total remaining service time) at each queue. While size- and state-aware dispatching to FCFS queues has been extensively studied, little is known about *optimal* dispatching for the objective of minimizing mean delay. A major obstacle is that no nontrivial lower bound on mean delay is known, even in heavy traffic (i.e. the limit as load approaches capacity). This makes it difficult to prove that any given policy is optimal, or even heavy-traffic optimal.

In this work, we propose the first size- and state-aware dispatching policy that provably minimizes mean delay in heavy traffic. Our policy, called *CARD (Controlled Asymmetry Reduces Delay)*, keeps all but one of the queues short, then routes as few jobs as possible to the one long queue. We prove an upper bound on CARD's mean delay, and we prove the first nontrivial lower bound on the mean delay of any size- and state-aware dispatching policy. Both results apply to any number of servers. Our bounds match in heavy traffic, implying CARD's heavy-traffic optimality. In particular, CARD's heavy-traffic performance improves upon that of LWL (Least Work Left), SITA (Size Interval Task Assignment), and other policies from the literature whose heavy-traffic performance is known.

CCS Concepts: • **General and reference** → **Performance**; • **Mathematics of computing** → **Markov processes**; • **Theory of computation** → *Routing and network design problems*; • **Networks** → *Network performance modeling*; *Network performance analysis*.

Additional Key Words and Phrases: dispatching, FCFS, response time, latency, sojourn time, heavy traffic, asymptotic optimality

ACM Reference Format:
Runhan Xie, Isaac Grosof, and Ziv Scully. 2024. Heavy-Traffic Optimal Size- and State-Aware Dispatching. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 1, Article 9 (March 2024), 36 pages. https://doi.org/10.1145/3639035

1 INTRODUCTION

Dispatching, or load balancing, is at the heart of many computer systems, service systems, transportation systems, and systems in other domains. In such systems, jobs arrive over time, and each job must be irrevocably sent to one of multiple queues as soon as it arrives. It is common for each queue to be served in First-Come First-Served (FCFS) order.

Motivated by the ubiquity of dispatching, we study a classical problem in dispatching theory:

Authors' addresses: Runhan Xie, runhan_xie@berkeley.edu, University of California, Berkeley, Department of Industrial Engineering and Operations Research, Berkeley, CA, USA; Isaac Grosof, igrosof@cs.cmu.edu, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, USA and Georgia Institute of Technology, School of Industrial and Systems Engineering, Atlanta, GA, USA; Ziv Scully, zivscully@cornell.edu, Cornell University, School of Operations Research and Information Engineering, Ithaca, NY, USA.

ACM 2476-1249/2024/3-ART9
https://doi.org/10.1145/3639035

How should one dispatch to FCFS queues to minimize jobs' mean response time?[1]

We specifically consider *size- and state-aware dispatching*. This means that the dispatcher learns a job's *size*, or service time, when the job arrives; and the dispatcher always knows how much *work*, or total remaining service time, there is at each queue. We make typical stochastic assumptions about the job arrival process, working with M/G arrivals (see Section 2).

Despite the extensive literature on dispatching in queueing theory (see Section 1.2), optimal size- and state-aware dispatching is an open problem, as highlighted by Hyytiä et al. [23]. The problem is a Markov decision process (MDP), so it can in principle be approximately solved numerically [27]. But the numerical approach has two drawbacks. First, the curse of dimensionality makes computation impractical for large numbers of queues. Second, the solution is specific to a particular instance (meaning a given number of queues, job size distribution, and load) and one has to solve the MDP again for a different instance.

1.1 Our contributions

In this work, we take the first steps towards developing a theoretical understanding of optimal size- and state-aware dispatching, making two main contributions.

- We give the first lower bound on the minimum mean response time achievable under any dispatching policy (Theorem 3.1).
- We propose a new dispatching policy, called *CARD (Controlled Asymmetry Reduces Delay)*, and prove an asymptotically tight upper bound on its mean response time (Theorem 3.3). We illustrate CARD in Figure 1.1.

Our upper and lower bounds match in the heavy-traffic limit as load ρ approaches 1, the maximum load capacity. Specifically, we find an explicit constant K such that the dominant term of both bounds is $\frac{K}{1-\rho}$. This makes CARD the first policy to be proven heavy-traffic optimal, aside from the implicitly specified optimal policy. Characterizing the optimal constant K, which was previously unknown, is another contribution of our work.

How CARD outperforms previous policies. Below, we describe the intuition behind CARD's design in a two-server system. See Figure 1.1 for an illustration.

To minimize mean response time, one generally wants to avoid situations where small jobs need to wait behind large jobs. One way to do this is to dedicate one server to small jobs and the other server to large jobs, where the size cutoff between "small" and "large" is defined such that half the load is due to each size class. This is the approach taken by the SITA (Size Interval Task Assignment) policy [17, 18]. Under SITA, due to Poisson splitting, the dispatching system reduces to two independent M/G/1 systems. As shown by Harchol-Balter et al. [18], SITA can sometimes perform very well, but it can sometimes be much worse than simple LWL (Least Work Left) dispatching, under which the system behaves like a central-queue M/G/2.

As each of LWL and SITA can sometimes be worse than the other in heavy traffic, one might expect that they can be strictly improved upon. Indeed, in Appendix B.1, we show that in the two-server case, both LWL and SITA are strictly suboptimal in heavy traffic. But the question remains: where in LWL or SITA's design is there a specific opportunity for improvement?

Our key observation is that the main reason SITA performs poorly is that its "short server", namely the queue to which it sends small jobs, can accumulate lots of work. CARD avoids this issue by actively regulating the amount of work at the short server. To do so, CARD creates a third class of "medium" jobs, which are on the border between small and large, and sets a threshold which serves as a target amount of work at the short server. Whenever a medium job arrives, CARD

[1] A job's *response time* (a.k.a. sojourn time, latency, delay) is the amount of time between its arrival and its completion.

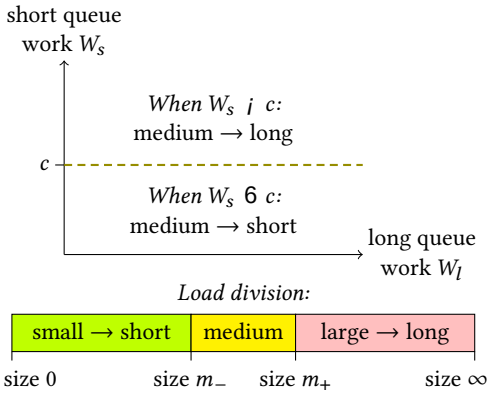


Fig. 1.1. Sketch of the CARD policy for two servers. Small and large jobs are always dispatched to the short or long server, respectively. Medium jobs are dispatched based on whether W_s, the amount of work at the short server, exceeds a threshold c. The size cutoffs m_- and m_+ are chosen so that small and large jobs each constitute slightly less than half the load.

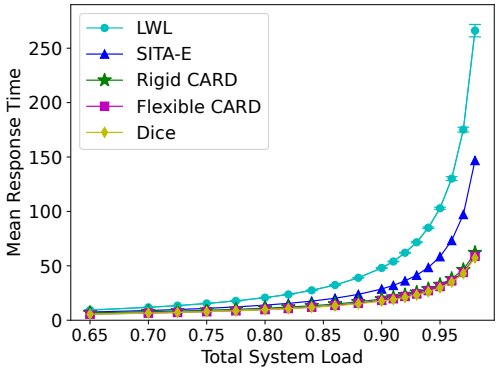


Fig. 1.2. Mean response time as a function of load for several policies, including two versions of CARD. *Rigid CARD* is the version we theoretically analyze, while *Flexible CARD* is modified slightly to improve empirical performance. The job size distribution has coefficient of variation cv = 10. See Section 7 and Figure 7.1(b) for further details.

dispatches it to the short server if and only if the short server has less work than the threshold. This prevents too much work accumulating in the short server, and it also prevents the short server from unduly idling.

CARD's performance beyond heavy traffic. Of course, practical systems rarely operate at loads very near capacity, but our theoretical bounds on CARD's performance are admittedly not tight outside the heavy-traffic regime. As such, we also study CARD in simulation across a wider range of loads. We find empirically that CARD has good performance outside of heavy traffic, but slightly modifying CARD can significantly improves performance. Both the original and modified versions of CARD improve upon traditional heuristics like LWL and SITA, sometimes by an order of magnitude. The modified version is competitive with the Dice policy of Hyytiä and Righter [26], the best known heuristic for the size- and state-aware setting. See Figure 1.2 for an example where at high load, CARD achieves reductions of over 75% relative to LWL and over 50% compared to SITA.

Outline. The remainder of the paper is organized as follows.

- Section 1.2 reviews related work.
- Section 2 presents our model and defines the CARD policy.
- Section 3 states our main results and gives some intuition for why they hold.
- Sections 4–6 prove our results: a lower bound on the performance, namely mean response time, of any policy (Section 4); stability of CARD (Section 5); and an upper bound on CARD's performance, which implies its heavy-traffic optimality ($n = 2$ servers in Section 6, general case in Appendix B.4).
- Section 7 studies CARD outside of heavy traffic via simulation.

We note that a preliminary version of this work appeared as a three-page workshop abstract [57], but it was extremely limited compared to the current version: it treated only the case of two servers and two job sizes, it did not provide any lower bound, and it omitted all proofs.

1.2 Related Work

FCFS dispatching with incomplete information. Whether a dispatching policy is optimal depends critically on the information available to the dispatcher. When the size of the arriving job is unknown, but server states (e.g. number of jobs at each server, work at each server, etc.) are known (state-aware), depending on the server-state information, Round-Robin (RR) [7, 34, 35], Join-Shortest-Queue (JSQ) [51, 55], and LWL [1, 5, 11, 31] are shown to be optimal. The common key idea of these policies is to join the queue with least (or least expected) amount of work.

When only the sizes and the distribution of the arriving jobs are known, SITA is known to be optimal [9]. But this result assumes that the dispatching policy must be entirely static. Recently, it was shown that combining SITA with RR can improve performance [2, 25], which combines SITA with just a little bit of memory, namely which servers most recently received a job.

Perhaps the closest the SITA line of work gets to size- and state-aware dispatching is the SITA-JSQ policy proposed in Wang and Down [50], in which the dispatcher uses the size of the arriving job and number of jobs at each server to make dispatching decisions. CARD is in some ways similar to SITA-JSQ, particularly the "multi-band" variant of CARD introduced in our simulation study (Section 7). But SITA-JSQ does not actively control the amount of work in each queue, and in particular does not maintain a large imbalance between queues. Our lower bound (Section 4) shows this imbalance is necessary for heavy-traffic optimality.

FCFS size- and state-aware dispatching. For size- and state-aware FCFS dispatching, various heuristics have been proposed and studied in simulations. Many of them are based on approximate dynamic programming e.g. [22, 24, 27]. Another class of policies, called sequential dispatching policies, are introduced in [23]. Among the sequential dispatching policies, Dice [23] shows superior performance in simulations and is among the best heuristics that has been developed. In our simulations (Section 7), Dice often slightly outperforms CARD. However, there is no theoretical analysis so far on the performance of Dice, even in heavy traffic.

Heavy-traffic Optimality Results. The aforementioned optimality results are strong in the sense that they either show stochastic ordering optimality on sample paths, or show optimality for any load of jobs. For more complicated policies and systems, characterizing the mean response time for an arbitrary load is a difficult task. Therefore, a large number of works focus on analyzing the heavy-traffic regime and establish optimality therein. One approach is to prove optimality via process limits e.g. [30, 48]. Such approach focuses on the transient regime and interchange of limits are usually not established for analysis in steady state. Another approach is to work directly in the stationary regime and establish heavy-traffic optimality results on mean response times in steady state e.g. [8, 59, 61]. However, these optimality results focus on settings where job sizes are unknown, so they do not address our goal of optimal size-aware dispatching.

Tools and Methodology. Recently, Eryilmaz and Srikant [8] introduced and popularized a Lyapunov drift-based approach that is applied to study the steady-state performance of queueing systems in heavy traffic. The approach has been adopted in studying various switches (e.g. [21, 28, 29, 32, 36, 37, 49]), load-balancing algorithms (e.g. [20, 33, 52, 58, 59, 61]), and other stochastic models (e.g. wireless scheduling, Stein's method, mean-field models). In some sense, our paper applies drift method to continuous-time continuous state Markov processes. Our use of the Rate Conservation Law [42] parallels the use of "zero drift" condition in drift analysis. An important step in drift analysis is establishing state-space collapse. We prove a result of this type in Lemma 5.2.

Other Relevant Work. When scheduling is allowed at the servers, optimal dispatching policies can be very different. When there are multiple parallel SRPT servers, Down and Wu [6] study a multi-layer

dispatching policy and show optimality using a diffusion limit argument. Grosof et al. [14] develop a dispatching policy, called *Guardrails*, that achieves optimal mean response time in heavy traffic.[2] Both of these prior dispatching policies involve, roughly speaking, balancing work evenly across the multiple SRPT servers. This is in contrast to CARD, which maintains a large imbalance between the multiple FCFS servers. One interpretation is that while SRPT prioritizes jobs at each individual server, CARD prioritizes jobs at the dispatching stage, namely by sending shorter jobs to servers with less work.

In recent years, learning-based dispatching policies have also been studied in literature [12, 44]. CARD involves tuning some parameters that depend on the job size distribution, and we thus assume knowledge of the job size distribution. An interesting question for future work is whether CARD's parameters could be learned online in settings where the job size distribution is unknown.

In the context of scheduling jobs on a single server, when SRPT (Shortest Remaining Processing Time) is shown to be optimal [45], Chen and Dong [4] show that having two priority classes is sufficient for a good performance in heavy traffic. The heavy-traffic performance of CARD ends up roughly equivalent to the performance of a single-server system with two priority classes. However, we cannot match the performance demonstrated by Chen and Dong [4]: they decrease the fraction of load in the lower-priority class to zero in heavy traffic, whereas CARD's "lower-priority jobs", namely those sent to the long server, must constitute a roughly $\frac{1}{n}$ fraction of the load.

2 SYSTEM MODEL AND THE CARD POLICY

2.1 Model Description

We consider a system of $n > 2$ identical FCFS (First-Come, First-Served) servers, each of which has its own queue. The system has one central dispatcher, which immediately dispatches jobs to a server when they arrive. We consider M/G job arrivals with (Poisson) arrival rate λ and job size distribution S. We assume $\mathsf{E}[S^2] < \infty$. The system load, namely the average rate at which work arrives, is $\rho = \lambda \mathsf{E}[S]$. We assume a server never idles unless there are no jobs present in its queue.

We use the convention that each server completes work at rate $\frac{1}{n}$, so a job of size s requires ns time in service. This convention means the largest possible stability region is $\rho \in [0, 1)$, regardless of the number of servers n. The convention is also convenient when comparing our system's performance to that of a "resource-pooled" M/G/1 with the same arrival process and server speed 1. We write $\mathsf{E}[W_{\mathrm{M/G/1}}]$ for the mean amount of work in such a resource-pooled M/G/1.

We consider *size- and state-aware* dispatching policies. That is, when a job arrives, the dispatcher may use both the job's size and the system state to decide where to dispatch it to. For our purposes, the most important aspect of the system state is the amount of *work* remaining at each server. We write W_i for the amount of work at server i (but see also Section 2.2), $\mathbf{W} = (W_1, \ldots, W_n)$ for the vector of work amounts, and $W_{\mathrm{all}} = \sum_{i=1}^{n} W_i$ for the total work. We write $W_i(t)$ or $\mathbf{W}(t)$ when discussing work at a specific time t.

The main metric we consider is *mean response time*. A job's response time is the amount of time between its arrival and completion. Due to our $\frac{1}{n}$ service rate convention, if a job of size s is dispatched to a server with w work, the job's response time is $n(w + s)$. We write $\mathsf{E}[T_\pi]$ for the mean response time over all jobs (in the usual limiting long-run average sense) under policy π.

[2] Guardrails is optimal in the sense that the mean response time under Guardrails matches that of a resource-pooled SRPT in heavy-traffic. As Theorem 3.1 suggests, dispatching to FCFS servers cannot in general match the performance of a resource-pooled SRPT in heavy-traffic. Therefore, in heavy-traffic, an optimal dispatching policy to SRPT servers generally outperforms an optimal policy to FCFS servers in terms of mean response time.

Purely for simplicity of notation, we assume the job size distribution S has no atoms. This is to ensure that expressions like $\mathsf{E}[S\mathcal{I}(S \ddot{Y} m)]$ are continuous functions of m. One can generalize all of our definitions and results to distributions with atoms using a lexicographic ordering trick.[3]

2.2 Defining the CARD Policy

We now introduce our policy, *CARD*, which stands for *Controlled Asymmetry Reduces Delay*. We first present it in the context of $n = 2$ servers, then generalize to $n > 2$ servers.

CARD for two servers. In the $n = 2$ case, CARD designates server 1 as the *short server* and server 2 as the *long server*. To emphasize this, when discussing CARD, we write $W_s = W_1$ and $W_\ell = W_2$ for the work at the short and long servers, respectively.

CARD has three threshold parameters to set:

- The two *size thresholds* m_- and m_+, $0 \leq m_- \leq m_+$, divide jobs into small, medium, and large (see below).
- The *work threshold* c, $c > m_+$ is, roughly speaking, a target work level for the short server.

Based on these parameters, CARD dispatches jobs as follows (see also Figure 1.1):

- A *small job*, namely one with size in $[0, m_-)$, is always dispatched to the short server.
- A *medium job*, namely one with size in $[m_-, m_+)$, is dispatched depending on W_s at time of arrival. If $W_s \leq c$, it is sent to the short server, and if $W_s > c$, it is sent to the long server.
- A *large job*, namely one with size in $[m_+, \infty)$, is always dispatched to the long server.

Setting CARD's parameters. There are a range of ways to set m_-, m_+, and c that yield stability and heavy-traffic optimality. We specify these formally in the statements of Theorems 3.2 and 3.3, but we highlight the key points here (see also Section 2.3).

The size thresholds m_- and m_+ should be chosen such that small jobs and large jobs are each less than half the load. Formally, we require

$$\mathsf{E}[S\mathcal{I}(S \ddot{Y} m_-)] \ddot{Y} \tfrac{1}{2}\mathsf{E}[S] \ddot{Y} \mathsf{E}[S\mathcal{I}(S \ddot{Y} m_+)].$$

In particular, we have $m_- \ddot{Y} m \ddot{Y} m_+$, where m is the solution to $\mathsf{E}[S\mathcal{I}(S \ddot{Y} m)] = \frac{1}{2}\mathsf{E}[S]$. As we show in our lower bound (Theorem 3.1), this value m is in some sense the ideal cutoff between small and large jobs. As such, it is important that in heavy traffic, either $m_- \to m$ or $m_+ \to m$ (or both). We do the former in our upper bound (Theorem 3.3).

The work threshold c must balance a tradeoff between two concerns. On one hand, we want there to be little work at the short server so that small jobs have low response times. On the other hand, we do not want the short server to run out of work, as excessive idling could increase response times or even cause instability. Roughly speaking, this means setting $c = \Theta\left(\frac{1}{1-\rho}^{p}\right)$ for a suitable choice of $p \in (0, 1)$.

It is convenient in our proofs to ensure $c > m_+$, so we assume this throughout. It also makes intuitive sense that a single medium job should not bring the short server from empty to above the work threshold. However, this assumption can be easily relaxed at the cost of a little more computation in the proofs.

Generalizing CARD to any number of servers. We now generalize the above policy to $n > 2$ servers. Here we focus on an extension that prioritizes simplicity of analysis while still achieving optimal heavy-traffic performance. In our simulation study (Section 7), we consider a more complex variant which has better performance at practical loads.

[3]Have the system assign each job an i.i.d. uniform $U \in [0, 1]$ independent of its size S, and replace comparisons $S < m$ with comparisons $(S, U) \prec (m, v)$ for some $v \in [0, 1]$, where $\prec$ is the lexicographic order. If $\mathsf{E}[S\mathcal{I}((S, U) \prec (m, v))]$ has a jump discontinuity at m, varying v interpolates continuously between the left and right limits.

The basic idea of n-server CARD is to reduce to the two-server case. We use the same three parameters m_-, m_+, and c, and we define small, medium, and large jobs in the same way. The only difference is that instead of one short and one long server, we use $n-1$ short servers $1,\dots,n-1$ and a single long server n. We thus write $W_{s_i} = W_i$ and $W_\ell = W_n$ when discussing n-server CARD. Abusing notation slightly, we write simply W_s when discussing a generic short server whose index is not important. Jobs are dispatched as follows:

- A small job is always dispatched to a uniformly random short server.
- A medium job is dispatched as follows. The dispatcher selects a uniformly random short server $N \in \{1,\dots,n-1\}$ and inspects its amount of work W_{s_N}. If $W_{s_N} \leq c$, the job is dispatched to the chosen short server N, and if $W_{s_N} > c$, it is dispatched to the long server.
- A large job is always dispatched to the long server.

Another way to view n-server CARD is in the following distributed manner. Suppose that instead of one dispatcher, we have $n-1$ independent "subdispatchers", each associated with a short server, and suppose that all jobs arrive at a uniformly random dispatcher. Then n-server CARD is the result of each of the subdispatchers using two-server CARD, except they all share the same long server.

The way we set the parameters of n-server CARD is essentially the same as how we set the parameters of two-server CARD. The only difference is that instead of wanting small and large jobs to both have less than half the load, we want small jobs to be less than a $1-\frac{1}{n}$ fraction of the load, and we want large jobs to be less than a $\frac{1}{n}$ fraction of the load. We therefore set

$$\mathsf{E}[S\mathcal{I}(S \leq m_-)] \leq 1-\frac{1}{n}\,\mathsf{E}[S] \leq \mathsf{E}[S\mathcal{I}(S \leq m_+)].$$

This means $m_- \leq m \leq m_+$, where now m is the solution to $\mathsf{E}[S\mathcal{I}(S \leq m)] = 1-\frac{1}{n}\,\mathsf{E}[S]$.

2.3 Key Definitions for Main Results and Analysis

We state our main results and perform our analysis in terms of the following quantities.

Drift-related quantities. The following quantities are related to characterizing *drifts*, which are the average rates at which work increases or decreases in various situations.

- Let $\varepsilon = 1-\rho$. If both servers are busy, then W_{all} has drift $-\varepsilon$.
- Let ρ_s, ρ_m, and ρ_ℓ be the loads due to small, medium, and large jobs, respectively:

$$\rho_s = \lambda\mathsf{E}[S\mathcal{I}(S \leq m_-)], \qquad \rho_m = \lambda\mathsf{E}[S\mathcal{I}(m_- < S \leq m_+)], \qquad \rho_\ell = \lambda\mathsf{E}[S\mathcal{I}(S > m_+)].$$

- Let α and β be the following quantities related to the drift of W_s:

$$\alpha = \frac{1}{n} - \frac{1}{n-1}\rho_s, \qquad \beta = \frac{1}{n-1}(\rho_s+\rho_m) - \frac{1}{n}.$$

  If $W_s < c$, then W_s has drift $-\alpha$, and if $0 < W_s \leq c$, then W_s has drift $+\beta$.
- Let $\delta \in (0,\varepsilon]$ be a bound on the probability the short server is idle, i.e. $\mathsf{P}[W_s = 0] \leq \delta$. We show how to set CARD's parameters to achieve this bound in Theorem 3.2(a).

To specify CARD's m_- and m_+ parameters, it suffices to specify α and β: these determine ρ_s and ρ_m, which in turn determine m_- and m_+. Moreover, for any given β, we show in Theorem 3.2 how to set CARD's c parameter to achieve $\mathsf{P}[W_s = 0] \leq \delta$. As such:

> Instead of specifying m_-, m_+, and c directly, we specify α, β, and δ.

In particular, Theorem 3.3 specifies how α, β, and δ should scale as functions of ε.

Heavy traffic. Our main results consider the $\varepsilon \downarrow 0$ limit, which we call the *heavy-traffic* regime. This is equivalent to $\lambda \uparrow 1/\mathsf{E}[S]$. In particular, we leave the number of servers fixed.

Underlying our results are explicit bounds that hold even outside the limiting regime (see e.g. Theorem 6.11). Because of our focus on heavy traffic, we assume for convenience that $\varepsilon \ddot{Y} \frac{1}{n}$. In particular, this ensures we can set $\beta \, ¡ \, 0$, which ensures that W_s always drifts towards c. The case where $\varepsilon \, ¡ \, \frac{1}{n}$ and $\beta \ddot{Y} 0$ is less interesting, as then both W_s and W_ℓ always drift towards 0.

Performance-related quantities. The following quantities are used in our response time bounds (Theorems 3.1 and 3.3). Define K_{CARD} and m such that

$$K_{\text{CARD}} = \frac{\mathsf{E}[S]}{\mathsf{E}[S \mid S > m]} = n\mathsf{P}[S > m]. \tag{2.1}$$

This characterization of m is equivalent to the aforementioned $\mathsf{E}[S\mathcal{I}(S \ddot{Y} m)] = \left(1 - \frac{1}{n}\right)\mathsf{E}[S]$. In Theorem 3.3, we show that, roughly speaking, $\mathsf{E}[T_{\text{CARD}}] \approx K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}]$, where

$$\mathsf{E}[W_{\text{M/G/1}}] = \frac{\lambda \mathsf{E}[S^2]}{2\varepsilon}$$

is the mean work in a resource-pooled M/G/1 (Section 2.1).

3 MAIN RESULTS AND KEY IDEAS

We now present our main results, followed by some intuition for why they hold. See Sections 4–6 for the proofs, with some details deferred to Appendix B.

Our first result is a lower bound on the mean response time for any dispatching policy.

Theorem 3.1. *Under any dispatching policy π and for any $\varepsilon \in (0, 1)$,*

$$\mathsf{E}[T_\pi] > K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}] - \frac{(n-1)\mathsf{E}[S^2]}{2m} + n\mathsf{E}[S].$$

Proof. See Section 4.

The rest of our results are about CARD: stability for all $\varepsilon \, ¡ \, 0$, and heavy-traffic optimality as $\varepsilon \downarrow 0$. Both results are stated as sufficient conditions on CARD's parameters under which it achieves the corresponding property. See Sections 2.2 and 2.3 for descriptions of and notation for CARD's parameters.

Theorem 3.2. *Let $\delta \, ¡ \, 0$, and consider CARD with threshold*

$$c = \frac{n(n-1)m_+}{\beta} \log \frac{n+1}{n\beta\delta}.$$

Then,

(a) Each short server satisfies $\mathsf{P}[W_s = 0] \, 6 \, \delta$.
(b) If $\delta \ddot{Y} \frac{n}{n-1}\varepsilon$, then the system is stable. Specifically, the set $\{(0, \dots, 0)\}$ is positive recurrent for the process $\mathsf{W}(t) = (W_1(t), \dots, W_n(t))$.

Proof. See Section 5 and Appendix B.2.

Theorem 3.3. *For any fixed number of servers $n > 2$, if CARD's parameters are set such that*

$$\alpha = \Theta(1), \qquad \beta = \Theta\left(\varepsilon^{1/3} \log \frac{1}{\varepsilon}\right)^{2/3}, \qquad \text{and} \qquad c = \frac{n(n-1)m_+}{\beta} \log \frac{n+1}{n\beta\delta},$$

in the $\varepsilon \downarrow 0$ limit, then CARD achieves mean response time bounded by

$$\mathsf{E}[T_{\text{CARD}}] \leq K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}] + O\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)^{1/3}.$$

In particular, CARD is heavy-traffic optimal: $\limsup_{\varepsilon\downarrow 0}\frac{\mathsf{E}[T_{\text{CARD}}]}{\mathsf{E}[T_\pi]} \leq 1$ *for any dispatching policy π.*

Proof. See Section 6 for the case of $n = 2$ servers and Appendix B.4 for the general case.

3.1 Intuition for Lower Bound on All Policies

We now give some intuition for Theorem 3.1. We focus on the heavy-traffic regime, where our aim is to show that the best possible mean response time is roughly $\mathsf{E}[T] \approx K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}]$.

To begin, recall $\mathsf{E}[T] = \frac{1}{\lambda}\mathsf{E}[N]$, where $\mathsf{E}[N]$ is the mean number of jobs in the system. The key idea is to relate $\mathsf{E}[N]$ to the mean amount of work $\mathsf{E}[W_{\text{all}}]$. This is helpful because one can easily show $\mathsf{E}[W_{\text{all}}] > \mathsf{E}[W_{\text{M/G/1}}]$ (see e.g. Theorem 6.1).

How can we relate $\mathsf{E}[N]$ to $\mathsf{E}[W_{\text{all}}]$? In heavy traffic, most jobs in the system are waiting in a queue and have yet to enter service. We thus approximate $\mathsf{E}[N] \approx \mathsf{E}[W_{\text{all}}]/\mathsf{E}[S_{\text{queue}}]$, where $\mathsf{E}[S_{\text{queue}}]$ is the mean size of jobs waiting in a queue. This means minimizing $\mathsf{E}[T]$ amounts to maximizing the mean size of jobs waiting in the queue. This makes sense in light of the fact that when studying scheduling policies beyond FCFS, serving small jobs ahead of large jobs reduces mean response time [45].

What is the largest that $\mathsf{E}[S_{\text{queue}}]$ can be? Because we are restricted to FCFS service, the only mechanism by which we can affect the sizes of jobs in the system is dispatching. In particular, we can dispatch jobs of different sizes to different servers. Suppose, for example, that servers $1, \ldots, n-1$ have a negligible amount of work, meaning nearly all of the work is at server n. Then $\mathsf{E}[S_{\text{queue}}]$ would be the average size of jobs dispatched to server n, which could be much greater than $\mathsf{E}[S]$. The best we could hope to do is $\mathsf{E}[S_{\text{queue}}] = \mathsf{E}[S \mid S > m]$ for as high a threshold m as possible. But in heavy traffic, we need server n to handle a $\frac{1}{n}$ fraction of the load, so the largest value of m possible solves $\mathsf{E}[S\mathcal{I}(S > m)] = \frac{1}{n}\mathsf{E}[S]$. This is equivalent to the characterization of m from (2.1), so it leads to $\mathsf{E}[T]/\mathsf{E}[W_{\text{all}}] \approx \mathsf{E}[S]/\mathsf{E}[S \mid S > m] = K_{\text{CARD}}$. Observing $\mathsf{E}[W_{\text{all}}] > \mathsf{E}[W_{\text{M/G/1}}]$ completes the bound.

To make this reasoning rigorous, it turns out that reasoning directly in terms of $\mathsf{E}[S_{\text{queue}}]$ is difficult. We instead prove Theorem 3.1 using a potential-function approach. However, the potential function and manipulations we perform on it were directly inspired by the intuition:

> The best-case scenario is to dedicate one server to the jobs of size at least m, and to ensure that all other servers have a negligible amount of work.

3.2 Intuition for Upper Bound on CARD

We now give some intuition for Theorem 3.3. By the lower bound intuition above, CARD is already well on its way to achieving the best-case scenario: it attempts to keep the amount of work at the $n-1$ short servers near c, and the long server only serves medium and large jobs. To show CARD matches the lower bound in heavy traffic, it would suffice to show the following.

- CARD does not have much more work than a resource-pooled M/G/1: $\mathsf{E}[W_{\text{all}}] \approx \mathsf{E}[W_{\text{M/G/1}}]$.
  - Roughly speaking, this amounts to showing that we avoid situations where one server is idle while another server has lots of work (see Theorem 6.1).
- CARD's short servers do not exceed c work by too much: $\mathsf{E}[W_s] \approx c$.
  - We also need to set c such that it is negligible in heavy traffic.
- CARD rarely dispatches medium jobs to the long server: $\mathsf{P}[W_s \leq c] \approx 1$.

Our main tool for showing these and related properties is examining what we call *below-above cycles*. Consider a particular short server. It alternates between *below periods*, during which $W_s \leq c$, and *above periods*, during which $W_s > c$. It turns out that much of our analysis rests on below-above cycles not being too long. One reason for this is that when enough short servers are in above periods, the long server is temporarily overloaded. Long periods of transient overload could cause $\mathsf{E}[W_{\text{all}}]$ to be significantly greater than $\mathsf{E}[W_{\text{M/G/1}}]$. Short below-above cycles prevent this possibility. See Section 6.1 for more details about how we use below-above cycles.

4 UNIVERSAL LOWER BOUND

Theorem 3.1. *Under any dispatching policy π and for any $\varepsilon \in (0,1)$,*

$$\mathsf{E}[T_\pi] > K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}] - \frac{(n-1)\mathsf{E}[S^2]}{2m} + n\mathsf{E}[S].$$

Before diving into the proof, we give the high-level idea for $n = 2$ servers.

Suppose an arrival occurs while $W_1 \ddot{Y} W_2$. For that individual arrival, its response time if it were sent to queue i would be $2W_i$ because each server processes work at rate $1/2$, so the "benefit" of sending it to queue 1 instead of queue 2 is $2(W_2 - W_1)$. Reasoning symmetrically if $W_1 \ddot{Y} W_2$, we conclude that the benefit of dispatching jobs to the shorter queue is proportional to $|W_2 - W_1|$.

The main challenge is therefore to show that no dispatching policy can both frequently dispatch to the shorter queue, and also maintain large difference $|W_2 - W_1|$ between the queues. The key observation is that if we dispatch the job to the shorter queue, then $|W_2 - W_1|$ decreases, so the next arrival would see less benefit. That is, we can view $|W_2 - W_1|$ as a type of resource: dispatching jobs to the shorter queue depletes it, while dispatching jobs to the longer queue replenishes it. It is thus best to dispatch shorter jobs to the shorter queue, which slowly depletes $|W_2 - W_1|$, and dispatch longer jobs to the longer queue, which quickly replenishes $|W_2 - W_1|$. To formalize the idea of viewing $|W_2 - W_1|$ as a resource, we use the potential function $\frac{1}{2}(W_2 - W_1)^2$.

The proof below handles any number of servers n. The idea is essentially the same as the $n = 2$ case, except we look at the work differences $|W_i - W_j|$ for every pair of servers $i < j$.

Proof of Theorem 3.1. Consider an arbitrary stationary dispatching policy π. We first introduce notation for π's dispatching decisions. Suppose a job of random size S arrives and observes work vector $\mathbf{W} = (W_1, \ldots, W_n)$. We denote by W_{choice} the work at the queue the arrival is dispatched to. Note that while S is independent of $\mathbf{W}$, it is *not* independent of W_{choice}. We also write $W_{\text{all}} = \sum_{i=1}^{n} W_i$ for the total work at all queues. Because each server does work at rate $1/n$, we can write $\mathsf{E}[T_\pi]$ as

$$\mathsf{E}[T_\pi] = n\mathsf{E}[W_{\text{choice}} + S] = \mathsf{E}[W_{\text{all}}] + \mathsf{E}[nW_{\text{choice}} - W_{\text{all}}] + n\mathsf{E}[S]. \tag{4.1}$$

The main task is to give a lower bound on $\mathsf{E}[nW_{\text{choice}} - W_{\text{all}}]$. To do so, we apply the rate conservation law of Miyazawa [42] to $V(\mathbf{W})$, where

$$V(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{\tilde{n}}\sum_{j=1}^{\tilde{i}} (w_i - w_j)^2.$$

The value of $V(\mathbf{W})$ can change in two ways.

- Work is done continuously at each nonempty queue. We denote this average continuous change by $\mathsf{E}[D_t V(\mathbf{W})]$.
- Arrivals add work to whichever queue the dispatcher chooses. By PASTA (Poisson Arrivals See Time Averages) [56], this yields average change $\lambda\mathsf{E}[V(\mathbf{W} + S\mathbf{e}_{\text{choice}}) - V(\mathbf{W})]$, where $\mathbf{e}_{\text{choice}}$ is the standard basis vector with a 1 indicating the queue the job is dispatched to.

The rate conservation law [42] states that the average rate of change of $V(\mathbf{W})$ is zero, so

$$\mathsf{E}[D_t V(\mathbf{W})] + \lambda \mathsf{E}[V(\mathbf{W} + S\mathbf{e}_{\text{choice}}) - V(\mathbf{W})] = 0. \tag{4.2}$$

We now investigate each of the two terms in (4.2). We first observe that $\mathsf{E}[D_t V(\mathbf{W})] \leq 0$, because in the absence of arrivals, for any two queues i and j, the absolute difference $|W_i - W_j|$ either decreases (if exactly one server is idle) or stays constant (otherwise). Therefore,

$$\mathsf{E}[V(\mathbf{W} + S\mathbf{e}_{\text{choice}}) - V(\mathbf{W})] \geq 0.$$

Expanding the definition of $V(\mathbf{w})$ and writing $\sum_{i<\text{choice}}$ for sums over all queues other than the one the job is dispatched to, we obtain

$$\begin{aligned}
0 \leq \frac{1}{2}\mathsf{E}\left[\sum_{i<\text{choice}} (W_{\text{choice}} + S - W_i)^2 - (W_{\text{choice}} - W_i)^2\right] \\
= \frac{n-1}{2}\mathsf{E}[S^2] + \mathsf{E}\left[\sum_{i<\text{choice}} S(W_{\text{choice}} - W_i)\right] \\
= \frac{n-1}{2}\mathsf{E}[S^2] + \mathsf{E}[S(nW_{\text{choice}} - W_{\text{all}})].
\end{aligned}$$

Subtracting both sides from $m\mathsf{E}[nW_{\text{choice}} - W_{\text{all}}]$ and using the fact that

$$-W_{\text{all}} \leq nW_{\text{choice}} - W_{\text{all}} \leq (n-1)W_{\text{all}},$$

we obtain

$$\begin{aligned}
m\mathsf{E}[nW_{\text{choice}} - W_{\text{all}}] &> \mathsf{E}[(m-S)(nW_{\text{choice}} - W_{\text{all}})] - \frac{n-1}{2}\mathsf{E}[S^2] \\
&> -\mathsf{E}\left[(S-m)^+(n-1)W_{\text{all}} - (m-S)^+ W_{\text{all}}\right] - \frac{n-1}{2}\mathsf{E}[S^2] \\
&\overset{(a)}{=} -\mathsf{E}\left[(n-1)(S-m)^+ + (m-S)^+\right]\mathsf{E}[W_{\text{all}}] + \frac{n-1}{2}\mathsf{E}[S^2] \\
&= -\left(m - \mathsf{E}[S] + n\mathsf{E}[(S-m)^+]\right)\mathsf{E}[W_{\text{all}}] + \frac{n-1}{2}\mathsf{E}[S^2],
\end{aligned} \tag{4.3}$$

where (a) follows from the fact that an arriving job's size S is independent of the work vector $\mathbf{W}$ it observes upon arrival.

We now substitute the bound from (4.3) into (4.1), obtaining

$$\mathsf{E}[T_\pi] = \frac{\mathsf{E}[S] - n\mathsf{E}[(S-m)^+]}{m}\mathsf{E}[W_{\text{all}}] - \frac{(n-1)\mathsf{E}[S^2]}{2m} + n\mathsf{E}[S].$$

The bound follows from $\mathsf{E}[W_{\text{all}}] \geq \mathsf{E}[W_{\text{M/G/1}}]$ (see e.g. Theorem 6.1) and (2.1), which implies

$$\mathsf{E}[S] - n\mathsf{E}[(S-m)^+] = \mathsf{E}[S] - n\mathsf{E}[S\mathcal{I}(S \geq m)] + mn\mathsf{P}[S \geq m] = mn\mathsf{P}[S \geq m] = mK_{\text{CARD}}.$$

5 CARD STABILITY ANALYSIS

Proving CARD's stability is more than a straightforward application of the Foster-Lyapunov theorem, which is widely used to establish stability of queueing systems. The main obstacle here is that the long server alternates between being underloaded and overloaded. It is thus difficult to find a Lyapunov function that is negative outside a compact set.

To overcome this obstacle, we use a result of Foss et al. [10, Theorem 1]. Notice that, under CARD, W_s is itself a Markov process because the decision of where to dispatch a job depends only on the work at the shorter server. Roughly, [10, Theorem 1] says that since W_s is a Markov process of its own, if it is ergodic, then it suffices to do a drift analysis of W_ℓ, *averaged over* the stationary

distribution of W_s. Of course, we first need to show that W_s is ergodic. Our proof for CARD's stability therefore proceeds in three steps.

- We show that the short server's work $W_s(t)$, as a Markov process of its own, is Harris ergodic (Lemma 5.1).
- With the stability of $W_s(t)$ in hand, we bound the idleness probability of the short server in steady state (Lemmas 5.2 and 5.3).
- We apply the result of Foss et al. [10, Theorem 1] (Theorem 3.2) to show stability whenever the long server is *on average* not overloaded. Our bound on the short server's idleness probability from the previous step thus gives a sufficient condition for stability.

Armed with these key ideas, the proofs themselves are relatively straightforward, with the bulk of the work being computation. As such, we defer most of these computation details to Appendix B.2.

Lemma 5.1. *W_s is Harris ergodic for any $\varepsilon > 0$.*

Proof sketch. The proof uses a Foster-Lyapunov theorem for continuous-time Markov processes [41, Theorem 4.2]. The key step is to verify that the Lyapunov function $V(w_s) = w_s$ has bounded drift when $w_s \leq c$ and negative drift when $w_s > c$. This is true because when $w_s > c$, we only send small jobs to the short server. We defer the details to Appendix B.2.

We establish our short server idleness bound by first proving a general bound on the probability that W_s is lower than c by a general amount x. The idleness bound follows by plugging in $x = c$.

Lemma 5.2. *Suppose $\theta > 0$ satisfies $(S_{s,m}^{\check{Z}})_e(\theta) > \frac{1}{n(n-1)\beta+n-1}$, where $(S_{s,m}^{\check{Z}})_e(\cdot)$ is the Laplace transform of the equilibium distribution of the size of small and medium jobs, $S_{s,m} = (S \mid S \ddot{Y} m_+)$. Then for all $x \in [0, c]$,*

$$\mathsf{P}[W_s \ddot{Y} c - x] \leq \frac{(n(n-1)\beta + n - 1)(S_{s,m}^{\check{Z}})_e(\theta)}{(n(n-1)\beta + n - 1)(S_{s,m}^{\check{Z}})_e(\theta) - 1} e^{-\theta x}.$$

Proof sketch. This result is a Chernoff-type bound on $(c - W_s)^+$, so the main task is to bound $\mathsf{E}[\exp(\theta(c - W_s)^+)]$. We do this by applying the rate conservation law [42] to $\exp(\theta(c - W_s)^+)$. We defer the details to Appendix B.2.

Lemma 5.3. *We have the following bound on the idleness of the short server,*

$$\mathsf{P}[W_s = 0] \leq \frac{n+1}{n\beta} \exp\left(-\frac{\beta c}{n(n-1)m_+}\right).$$

Proof. Let $\theta = \frac{\beta}{n(n-1)} \frac{1}{m_+}$. Since $\beta \ddot{Y} \frac{1}{n(n-1)}$ and all small and medium jobs have length at most m_+, we have $(S_{s,m}^{\check{Z}})_e(\theta) > 1 - \theta \mathsf{E}[(S_{s,m})_e] > 1 - \frac{\beta}{n(n-1)}$. We can therefore apply Lemma 5.2, from which the bound follows by the computation below and setting $x = c$:

$$\begin{aligned}\mathsf{P}[W_s \ddot{Y} c - x] &\leq \frac{(n(n-1)\beta + n - 1)\left(1 - \frac{\beta}{n(n-1)}\right)}{(n(n-1)\beta + n - 1)\left(1 - \frac{\beta}{n(n-1)}\right) - 1} \exp\left(-\frac{\beta x}{n(n-1)m_+}\right) \\ &\leq \frac{n+1}{n\beta} \exp\left(-\frac{\beta x}{n(n-1)m_+}\right).\end{aligned}$$

We defer the proof of Theorem 3.2 to Appendix B.2.

6 CARD MEAN RESPONSE TIME ANALYSIS

With the lower bound from Theorem 3.1 in mind, our next step is to establish an upper bound on the mean response time under CARD. We focus here on the two-server case. The general case uses the same ideas but has more complicated computations, so we defer its proof to Appendix B.4.

Let $\mathsf{E}[T_{\text{CARD},s}]$, $\mathsf{E}[T_{\text{CARD},m}]$, and $\mathsf{E}[T_{\text{CARD},\ell}]$ be the mean response times of small, medium, and large jobs under CARD, respectively. We have

$$\begin{aligned}\mathsf{E}[T_{\text{CARD}}] &= p_s\mathsf{E}[T_{\text{CARD},s}] + p_m\mathsf{E}[T_{\text{CARD},m}] + p_\ell\mathsf{E}[T_{\text{CARD},\ell}] \\ &\leq 2\mathsf{E}[S] + 2p_s\mathsf{E}[W_s] + 2p_m c\mathsf{P}[W_s \leq c] + 2p_m\mathsf{E}[W_\ell \mathcal{I}(W_s > c)] + 2p_\ell\mathsf{E}[W_\ell]. \quad (6.1)\end{aligned}$$

where the inequality follows from how CARD dispatches jobs, the PASTA property [56], and the fact that the servers complete work at rate $1/2$. The main difficulty of analyzing (6.1) lies in bounding $\mathsf{E}[W_\ell]$ and $\mathsf{E}[W_\ell \mathcal{I}(W_s > c)]$. We now give a high-level overview of the obstacles and our approach.

6.1 Key Ingredients: Work Decomposition, Below-Above Cycles, and Palm Inversion

To bound $\mathsf{E}[W_\ell]$, it suffices to bound $\mathsf{E}[W_{\text{all}}]$. The following theorem, called the *work decomposition law* [46, 47], provides a way to bound $\mathsf{E}[W_{\text{all}}]$. We state it below in a way that is specialized to our system.

Theorem 6.1. *Denote by I the fraction of servers that are idle in steady state, namely*

$$I = \frac{1}{n}\sum_{i=1}^{\tilde{n}} \mathcal{I}(W_i = 0).$$

The steady-state mean total work $\mathsf{E}[W_{\text{all}}]$ satisfies

$$\mathsf{E}[W_{\text{all}}] = \mathsf{E}[W_{\text{M/G/1}}] + \frac{\mathsf{E}[IW_{\text{all}}]}{\varepsilon} = \frac{\frac{\lambda}{2}\mathsf{E}[S^2] + \mathsf{E}[IW_{\text{all}}]}{\varepsilon},$$

where $\mathsf{E}[W_{\text{M/G/1}}]$ is the work in an M/G/1 with arrival rate λ and job size distribution S.

The key component we need to bound from Theorem 6.1 is $\mathsf{E}[IW_{\text{all}}]$. We would like to study

$$\begin{aligned}IW_{\text{all}} &= (W_s + W_\ell)\ \tfrac{1}{2}\mathcal{I}(W_s = 0) + \tfrac{1}{2}\mathcal{I}(W_\ell = 0) \\ &= \tfrac{1}{2}W_\ell\mathcal{I}(W_s = 0) + \tfrac{1}{2}W_s\mathcal{I}(W_\ell = 0)\end{aligned}$$

The main difficulty here is to bound $\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)]$. Since CARD dispatches differently to the long server based on the state of the short server, W_ℓ depends on the state of W_s. Such a dependency also poses challenges in analyzing $\mathsf{E}[W_\ell \mid W_s > c]$, when even knowing $\mathsf{E}[W_\ell]$ is not sufficient.

Under CARD, W_s alternates between being above and below the threshold c. Such a behavior naturally leads to renewal intervals consists of the "above" periods and "below" periods.

Definition 6.2. We partition time into alternating intervals, called *below periods* and *above periods*, as follows:

- A time t is in a *below period* if $W_s(t) \leq c$.
- A time t is in an *above period* if $W_s(t) > c$.

A *below-above cycle* is then a complete below period followed by a complete above period. Below-above cycles start at times t for which $W_s(t) = c$. We can partition time into below-above cycles.

We introduce the following notation for working with below periods, above periods, and below-above cycles:

- We write $\mathsf{E}_c^0[\cdot]$ for the Palm expectation [3] taken at the start of a below-above cycle. Roughly speaking, $\mathsf{E}_c^0[\cdot]$ = "$\mathsf{E}[\cdot \mid$ a below period starts at time 0]", but the formal definition avoids conditioning on a measure-zero event.

- In the context of a below-above cycle starting at time 0, meaning $W_s(0) = c$, we denote the lengths of the below and above period by B and A, respectively:

$$B = \inf\{t > 0 : W_s(t) < c\},$$
$$A = \inf\{t > B : W_s(t) = c\}.$$

  Abusing notation slightly, we also use B and A to denote the lengths of the below and above period in a generic below-above cycle, not necessarily one that starts at time 0.

Why are above and below periods helpful for analyzing CARD? Within an above or below period, CARD does not change how it dispatches jobs, making it easier to analyze W_ℓ within *one* below-above cycle. The Palm inversion formula [3], which is a generalization of the celebrated renewal-reward theorem, allows us to connect the average behavior of W_ℓ within one below and above cycle to a steady-state average. For example, it implies

$$\mathsf{E}[W_\ell] = \frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^{A+B} W_\ell(t)\,\mathrm{d}t\right], \qquad \mathsf{E}[W_\ell \mathcal{I}(W_s < c)] = \frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_B^{A+B} W_\ell(t)\,\mathrm{d}t\right].$$

Our high-level idea is to relate both of these quantities to $\mathsf{E}_c^0[W_\ell(0)]$, the mean work at the long server at the start of a below-above cycle. We show in Lemmas 6.6 and 6.7 that, roughly speaking,

$$\mathsf{E}[W_\ell] \approx \mathsf{E}_c^0[W_\ell(0)], \qquad \mathsf{E}[W_\ell \mathcal{I}(W_s < c)] \approx \mathsf{E}_c^0[W_\ell(0)]\,\mathsf{P}[W_s < c]. \tag{6.2}$$

The rest of this section is organized as follows.

- Section 6.2 analyzes the behavior of the short server. In particular, we show that above and below cycles are not too long.
- Section 6.3 analyzes the behavior of the long server. Using the fact that above and below cycles are not too long, we show (6.2). As part of this, we bound $\mathsf{E}[W_{\mathsf{all}}]$.
- Section 6.4 assembles the pieces to prove Theorem 3.3.

6.2 Analyzing the Short Server and Below-Above Cycles

In this section, we bound various quantities relating to work at the short server and the below-above cycles. Of particular importance are the mean *excesses* of the above and below periods $\mathsf{E}[A_e]$ and $\mathsf{E}[B_e]$, as they are used to better understand the relations between $\mathsf{E}[W_\ell]$ and $\mathsf{E}_c^0[W_\ell(0)]$.

The techniques we use to obtain bounds on $\mathsf{E}[A_e]$ and $\mathsf{E}[B_e]$ also immediately yield bounds on $\mathsf{E}[A]$ and $\mathsf{E}[B]$. Despite not using these bounds, given that they help complete the picture of how the system behaves, we state them, too.

As a reminder, the *excess* or *equilibrium distribution* of a random variable V is the distribution V_e whose probability density function is $f(t) = \mathsf{P}[V > t]/\mathsf{E}[V]$. The excess arises naturally in renewal theory [3, 16, 43]. Most important for our purposes is the fact that

$$\mathsf{E}[V_e] = \frac{\mathsf{E}[V^2]}{2\mathsf{E}[V]}. \tag{6.3}$$

Lemma 6.3.

$$\mathsf{E}[B] \leq \frac{m_+}{\beta}, \qquad \mathsf{E}[B_e] \leq \frac{c+m_+}{\beta} \leq \frac{2c}{\beta}, \qquad \textit{and} \qquad \mathsf{E}[(B_e)_e] \leq \frac{c+m_+}{\beta} \leq \frac{2c}{\beta}.$$

Proof. Suppose that at time 0, the short server has $W_s(0) = v \leq c$ work, so time 0 is in a below period. Let $\tau(v)$ be the time until the end of the below period. We will show

$$\mathsf{E}[\tau(v)] \leq \frac{c+m_+-v}{\beta} \leq \frac{c+m_+}{\beta} \leq \frac{2c}{\beta}, \tag{6.4}$$

where the last step follows because $c > m_+$ (Section 2.2). This implies all three of the bounds.

- A below period starts with c work at the short server, so $\mathsf{E}[B] = \mathsf{E}[\tau(c)] \leq \frac{m_+}{\beta}$.
- The excesses B_e and $(B_e)_e$ can both be interpreted as the distribution of the amount of time until the below period ends, starting from some random amount of work at the short server, so their means can each be written as $\mathsf{E}[\tau(V)]$ for an appropriate variable V.

It remains only to show (6.4), which we do using a supermartingale argument. Suppose $W_s(0) = v$ as above, and define $X(t) = c - W_s(t) + \beta t$. We now show that $X(t)$ is a supermartingale with respect to the Markov process $W_s(t)$. Let

- $\Delta_s(u,t)$ be the amount of work completed by the short server during $(u,t]$ and
- $\Sigma_s(u,t)$ be the amount of work that arrives to the short server during $(u,t]$.

For any $0 \leq u \leq t$, we have

$$\begin{aligned}\mathsf{E}[X(t) \mid W_s(u)] - X(u) &= \mathsf{E}[W_s(u) - W_s(t) \mid W_s(u)] + \beta(t-u)\\ &= \mathsf{E}[\Delta_s(u,t) - \Sigma_s(u,t) \mid W_s(u)] + \beta(t-u)\\ &\leq \mathsf{E}\left[\tfrac{1}{2}(t-u) - \Sigma_s(u,t) \mid W_s(u)\right] + \beta(t-u)\\ &= (t-u)\left[\tfrac{1}{2} - \rho_s - \rho_\ell + \beta\right] = 0,\end{aligned}$$

so $X(t)$ is indeed a supermartingale. Applying the optional stopping theorem to $X(t)$ and $\tau(v)$, which we justify below, yields

$$c - v = \mathsf{E}[X(0)] > \mathsf{E}[X(\tau(v))] = c - W_s(\tau(v)) + \beta\tau(v) \overset{(a)}{\geq} -m_+ + \beta\tau(v),$$

from which (6.4) follows. Above, (a) uses the fact that all medium jobs have size at most m_+, so at the moment the below period ends, the short server's work can jump to at most $c + m_+$.

All that remains is to verify that we can indeed apply the optional stopping theorem.

- We have $\mathsf{E}[\tau(v)] < \infty$ by positive recurrence of $W_s(t)$.
- We have uniform integrability, namely $\lim_{t\to\infty} \mathsf{E}[X(t)\,\mathbb{I}(\tau(v) > t)] = 0$, thanks to the following two observations. First, $\mathsf{E}[W_s(t)\,\mathbb{I}(\tau(v) > t)] \to 0$ because $c - W_s(t) \in [0,c]$ when t is in a below period. Second, $\mathsf{E}[\beta t\,\mathbb{I}(\tau(v) > t)] \leq \mathsf{E}[\beta\tau(v)\,\mathbb{I}(\tau(v) > t)] \to 0$ because $\mathsf{E}[\tau(v)] < \infty$.

Lemma 6.4.

$$\mathsf{E}[W_s - c \mid W_s > c] \leq \frac{m_+}{4\alpha} \qquad \text{and} \qquad \mathsf{E}[(W_s - c)^2 \mid W_s > c] \leq \frac{m_+^2}{8\alpha^2}$$

Proof sketch. Each above period starts with $W_s - c \in [0, m_+]$. Until the end of the above period, $W_s - c$ evolves like the amount of work in an M/G/1 queue with server speed $1/2$, job size distribution S_s, and work arrival rate $\rho_s < 1/2$. This means $(W_s - c \mid W_s > c)$ has the same distribution as an M/G/1 with vacations, where the vacation length distribution is that of $W_s - c$ at the start of an above period. The desired bounds follow from the work decomposition formula for the M/G/1 with vacations [13] and the observation that both job sizes and vacation lengths are bounded by m_+. We defer the details to Appendix B.3.

Lemma 6.5.

$$\mathsf{E}[A] \leq \frac{m_+}{\alpha} \qquad \text{and} \qquad \mathsf{E}[A_e] \leq \frac{m_+}{4\alpha^2}.$$

Proof. As in the proof sketch of Lemma 6.4, we view the short server during an above period as an M/G/1 with server speed $1/2$ and work arrival rate ρ_s, so the mean drift of W_s is $-(1/2 - \rho_s) = -\alpha$. By standard results for M/G/1 busy periods [16], starting from $W_s - c = v$, it takes v/α time in expectation for the above period to end.

- The $\mathsf{E}[A]$ bound follows from the fact that at the start of an above period, $W_s - c \leq m_+$, implying $\mathsf{E}[A] \leq m_+/\alpha$.
- The $\mathsf{E}[A_e]$ bound follows from the fact that the residual time of an above period is distributed as A_e. But the residual time of an above period is the same as the amount of time until an above period ends starting from the stationary distribution of $W_s - c$ conditional on being in an above period. This means $\mathsf{E}[A_e] = \mathsf{E}[W_s - c \mid W_s > c]/\alpha$, so the result follows from Lemma 6.4.

6.3 Analyzing the Long Server

In this section, we bound differences between $\mathsf{E}_c^0[W_\ell(0)]$ and $\mathsf{E}[W_\ell]$, $\mathsf{E}[W_\ell \mathcal{I}(W_s > c)]$, and $\mathsf{E}[W_\ell \mathcal{I}(W_s = c)]$, separately. These bounds will help us upper bound $\mathsf{E}[W_{\text{all}}]$, thereby obtaining a bound on $\mathsf{E}[W_\ell]$.

Let q_A and q_B be the probabilities of being in an above or below period, respectively. That is,

$$q_A = \mathsf{P}[W_s > c] = \frac{\mathsf{E}[A]}{\mathsf{E}[A+B]} \qquad \text{and} \qquad q_B = \mathsf{P}[W_s \leq c] = \frac{\mathsf{E}[B]}{\mathsf{E}[A+B]}, \tag{6.5}$$

where the expressions in terms of expectations of A and B follow from renewal-reward theorem.

Lemma 6.6.

$$\mathsf{E}[W_\ell] - \mathsf{E}_c^0[W_\ell(0)] \leq \left(\sqrt{q_A \mathsf{E}[A_e]} + \sqrt{q_B \mathsf{E}[B_e]}\right)^2 \leq \frac{q_A m_+}{2\alpha^2} + \frac{4 q_B c}{\beta}.$$

Proof. The long server workload process can be described as

$$W_\ell(t) = \left(W_\ell(0) - \Delta_\ell(0,t) + \Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t)\right)^+, \tag{6.6}$$

where

- $\Delta_l(0,t)$ is the total work processed by the long server in $(0,t]$,
- $\Sigma_\ell^m(0,t)$ is the total work added to the long server from medium job arrivals in $(0,t]$, and
- $\Sigma_\ell^\ell(0,t)$ is the total work added to the long server due to large job arrivals in $(0,t]$.

Applying the Palm inversion formula [3] to W_ℓ gives

$$\begin{aligned}
\mathsf{E}[W_\ell] &= \frac{1}{\mathsf{E}[A+B]} \mathsf{E}_c^0 \left[\int_0^{A+B} \left(W_\ell(0) - \Delta_\ell(0,t) + \Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t)\right)^+ \mathrm{d}t \right] \\
&\overset{(a)}{=} \mathsf{E}_c^0[W_\ell(0)] + \frac{1}{\mathsf{E}[A+B]} \mathsf{E}_c^0 \left[\int_0^{A+B} \max\left(-\Delta_\ell(0,t) + \Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t), -W_\ell(0)\right) \mathrm{d}t \right],
\end{aligned}$$

where (a) holds since $W_\ell(0)$, the amount of long server work at time 0, is independent of $A+B$, the length of the below-above cycle starting at time 0.

We now bound $\mathsf{E}[W_\ell] - \mathsf{E}_c^0[W_\ell(0)]$ separately from above and below. To obtain a lower bound, we bound the integrand below by $-\Delta(t)$, obtaining

$$\mathsf{E}[W_\ell] - \mathsf{E}_c^0[W_\ell(0)] > -\frac{1}{\mathsf{E}[A+B]} \mathsf{E}_c^0 \left[\int_0^{A+B} \Delta_\ell(0,t)\, \mathrm{d}t \right] \overset{(b)}{>} -\frac{1}{\mathsf{E}[A+B]} \mathsf{E}_c^0 \left[\int_0^{A+B} \frac{t}{2}\, \mathrm{d}t \right] = -\frac{\mathsf{E}[(A+B)^2]}{4\mathsf{E}[A+B]},$$

where (b) holds because the server completes work at rate $\frac{1}{2}$ while it is busy. To obtain an upper bound, we bound the integrand above by $\Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t)$. We first bound its conditional expectation given A and B. Notice that $\Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t)$ consists of arrivals of large jobs during $(0,t]$ and medium jobs during $(B,t]$. Neither of these types of arrivals impacts the lengths of the above and below periods, so

$$\mathsf{E}_c^0[\Sigma_\ell^m(0,t) + \Sigma_\ell^\ell(0,t) \mid A, B] = \rho_m (t-B)^+ + \rho_\ell t \leq t. \tag{6.7}$$

From (6.7) and a computation similar to the lower bound, we obtain

$$\mathsf{E}[W_\ell] - \mathsf{E}_c^0[W_\ell(0)] \le \frac{\mathsf{E}[(A+B)^2]}{2\mathsf{E}[A+B]}.$$

Combining this with the lower bound, the result follows from (6.3), (6.5), and Cauchy-Schwarz:

$$\frac{\mathsf{E}[(A+B)^2]}{2\mathsf{E}[A+B]} \le \frac{\mathsf{E}[A^2] + \sqrt{\mathsf{E}[A^2]\mathsf{E}[B^2]} + \mathsf{E}[B^2]}{2\mathsf{E}[A+B]} = q_A\mathsf{E}[A_e] + 2\sqrt{q_A\mathsf{E}[A_e]\, q_B\mathsf{E}[B_e]} + q_B\mathsf{E}[B_e].$$

To complete the proof, we use the AM-GM inequality (arithmetic mean > geometric mean) on $\sqrt{q_A\mathsf{E}[A_e]\, q_B\mathsf{E}[B_e]}$, then apply our bounds on $\mathsf{E}[A_e]$ and $\mathsf{E}[B_e]$ from Lemmas 6.3 and 6.5.

Lemma 6.7.

$$\mathsf{E}[W_\ell \mathcal{I}(W_s < c)] - q_A\mathsf{E}_c^0[W_\ell(0)] \le q_A\mathsf{E}[A_e] + 2\sqrt{q_A\mathsf{E}[A_e]\, q_B\mathsf{E}[B_e]} \le \frac{q_A m_+}{4\alpha^2} + \frac{\sqrt{2q_Aq_Bm_+c}}{\alpha\sqrt{\beta}}.$$

Proof. Similar to that of Lemma 6.6. See Appendix B.3.

Lemma 6.8.

$$\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)] \le \delta\mathsf{E}_c^0[W_\ell(0)] + \sqrt{\delta\mathsf{E}[B_e^2]} \le \delta\mathsf{E}_c^0[W_\ell(0)] + \frac{2c\sqrt{2\delta}}{\beta}.$$

Proof. Applying Palm inversion formula [3] to $W_\ell \mathcal{I}(W_s = 0)$ yields

$$\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)] = \frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B W_\ell(t)\, \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right],$$

where we can end the integral at B because we only have $W_s(t) = 0$ during below periods, which corresponds to $t \in [0, B)$. We further expand the right-hand side using (6.6). No medium jobs are dispatched to the short server during below periods, so

$$\begin{aligned}\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)] &\le \frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B (W_\ell(0) + \Sigma_\ell^\ell(0,t))\, \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right] \\ &\overset{(a)}{=} \frac{\mathsf{E}_c^0[W_\ell(0)]}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right] + \frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B \Sigma_\ell^\ell(0,t)\, \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right].\end{aligned}$$

where (a) follows from the independence of $W_\ell(0)$ and $\int_0^B \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t$. To analyze the first term, we observe that by the Palm inversion formula [3] and Theorem 3.2,

$$\frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right] = \mathsf{E}[\mathcal{I}(W_s = 0)] = \mathsf{P}[W_s = 0] \le \delta.$$

To analyze the second term, we apply (6.7), yielding

$$\mathsf{E}_c^0\left[\int_0^B \Sigma_\ell^\ell(0,t)\, \mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right] \le \mathsf{E}_c^0\left[\int_0^B t\mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right].$$

The right-hand side is difficult to compute directly due to the dependency of B and W_s. To resolve this, we apply the Palm inversion formula [3] to $B_a\mathcal{I}(W_s = 0)$, where $B_a(t)$ is the age process of the below-above cycle, namely the amount of time since the current cycle began. This yields

$$\frac{1}{\mathsf{E}[A+B]}\mathsf{E}_c^0\left[\int_0^B t\mathcal{I}(W_s(t) = 0)\, \mathrm{d}t\right] = \mathsf{E}[B_a\mathcal{I}(W_s = 0)]$$

Thus, to bound $\mathcal{T}_3$, it suffices to bound $\mathsf{E}[B_a \mathcal{I}(W_s = 0)]$. By Cauchy-Schwarz,

$$\mathsf{E}[B_a \mathcal{I}(W_s = 0)] \le \sqrt{\mathsf{E}[B_a^2]\mathsf{P}[W_s = 0]} \overset{\text{(b)}}{=} \sqrt{\mathsf{E}[B_e^2]\mathsf{P}[W_s = 0]} \overset{\text{(c)}}{\le} \sqrt{\delta \mathsf{E}[B_e^2]},$$

where (b) follows because B_a has distribution B_e, and (c) follows from Theorem 3.2. The result then follows from bounding $\mathsf{E}[B_e^2]$ using (6.3) and Lemma 6.3.

Lemma 6.9.

$$\mathsf{E}[W_\ell] \le \mathsf{E}[W_{\text{all}}] \le \left(1 + \frac{\delta}{\varepsilon}\right) \mathsf{E}[W_{\text{M/G/1}}] + 2c + \frac{m_+\sqrt{q_A}}{2\alpha\sqrt{\varepsilon}} + \frac{4c\sqrt{\delta}}{\alpha^2\beta\varepsilon}.$$

Proof. We use Theorem 6.1 to bound $\mathsf{E}[W_{\text{all}}]$, which amounts to analyzing $\mathsf{E}[IW_{\text{all}}]$. We have

$$IW_{\text{all}} = (W_s + W_\ell)\left(\tfrac{1}{2}\mathcal{I}(W_s = 0) + \tfrac{1}{2}\mathcal{I}(W_\ell = 0)\right) = \tfrac{1}{2}W_\ell \mathcal{I}(W_s = 0) + \tfrac{1}{2}W_s \mathcal{I}(W_\ell = 0).$$

Combining Lemmas 6.6 and 6.8 and noting $\mathsf{E}[W_\ell] \le \mathsf{E}[W_{\text{all}}]$ yields a bound on $\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)]$:

$$\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)] \le \delta\left(\mathsf{E}[W_{\text{all}}] + \frac{q_A m_+}{4\alpha^2} + \frac{4q_B c}{\beta}\right) + \frac{2c\sqrt{2\delta}}{\beta}.$$

To bound $\mathsf{E}[W_s \mathcal{I}(W_\ell = 0)]$, we compute

$$\begin{aligned}
\mathsf{E}[W_s \mathcal{I}(W_\ell = 0)] &\overset{\text{(a)}}{\le} \mathsf{E}[(c + (W_s - c)^+)\mathcal{I}(W_\ell = 0)] \\
&\overset{\text{(b)}}{\le} c\mathsf{P}[W_\ell = 0] + \sqrt{\mathsf{E}[((W_s - c)^+)^2]\,\mathsf{P}[W_\ell = 0]} \\
&= c\mathsf{P}[W_\ell = 0] + \sqrt{q_A \mathsf{E}[(W_s - c)^2 \mid W_s > c]\,\mathsf{P}[W_\ell = 0]} \\
&\overset{\text{(c)}}{\le} 2\varepsilon c + \frac{m_+\sqrt{q_A \varepsilon}}{2\alpha},
\end{aligned}$$

where (a) follows from $W_s \le c + (W_s - c)^+$, (b) follows from Cauchy-Schwarz, and (c) follows from Lemma 6.4 and the fact that $\varepsilon = \frac{1}{2}\mathsf{P}[W_s = 0] + \frac{1}{2}\mathsf{P}[W_\ell = 0] > \frac{1}{2}\mathsf{P}[W_\ell = 0]$. Combining the bounds on $\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)]$ and $\mathsf{E}[W_s \mathcal{I}(W_\ell = 0)]$ with Theorem 6.1, we obtain

$$\begin{aligned}
\mathsf{E}[W_{\text{all}}] &= \mathsf{E}[W_{\text{M/G/1}}] + \frac{1}{\varepsilon}\left(\tfrac{1}{2}\mathsf{E}[W_\ell \mathcal{I}(W_s = 0)] + \tfrac{1}{2}\mathsf{E}[W_s \mathcal{I}(W_\ell = 0)]\right) \\
&\le \mathsf{E}[W_{\text{M/G/1}}] + c + \frac{m_+\sqrt{q_A}}{4\alpha\sqrt{\varepsilon}} + \frac{\delta}{2\varepsilon}\mathsf{E}[W_{\text{all}}] + \frac{q_A m_+ \delta}{8\alpha^2\varepsilon} + \frac{2q_B c\delta}{\beta\varepsilon} + \frac{c\sqrt{2\delta}}{\beta\varepsilon}.
\end{aligned}$$

The result follows after rearranging and simplifying. We use the fact that we have defined the parameters such that $c > m_+$ and $\delta \le \varepsilon$ (Sections 2.2 and 2.3), which means $1/\left(1 - \frac{\delta}{2\varepsilon}\right) \le 1 + \frac{\delta}{\varepsilon} \le 2$. And, using the fact that $\alpha, \beta \le \frac{1}{2}$, we loosely bound the terms with a $\sqrt{\delta}$ factor by

$$\frac{q_A m_+ \delta}{8\alpha^2\varepsilon} + \frac{2q_B c\delta}{\beta\varepsilon} + \frac{c\sqrt{2\delta}}{\beta\varepsilon} \le (q_A + q_B)\frac{c\delta}{2\alpha^2\beta\varepsilon} + \frac{c\sqrt{2\delta}}{\beta\varepsilon} \le \frac{2c\sqrt{\delta}}{\alpha^2\beta\varepsilon}.$$

6.4 Bounding Mean Response Time

We now prove Theorem 3.3, our main upper bound result. It follows as a corollary of a more explicit bound, which we state in Theorem 6.11. To simplify the computations, we assume that $\beta > 2\delta$, but we could remove this assumption at the cost of slightly complicating the expressions.

Lemma 6.10. *If $\beta > 2\delta$, then $q_A \le \frac{2\beta}{\alpha+\beta}$ and $q_B \le \frac{\alpha}{\alpha+\beta}$.*

Proof. The short server is stable, so the load of jobs arriving to it equals the average rate it completes work. This means $\rho_s + \rho_m \mathsf{P}[W_s \le c] = \frac{1}{2}\mathsf{P}[W_s > 0]$. Theorem 3.2 implies $\mathsf{P}[W_s > 0] \in [1-\delta, 1]$, so the bound follows from the definitions of α and β and the $\beta > 2\delta$ assumption.

Theorem 6.11. *In a system with $n = 2$ servers, if $\delta \le \varepsilon \le \frac{1}{2}$ and $\beta > 2\delta$, then by setting c according to Theorem 3.2, CARD achieves mean response time bounded by*

$$\mathsf{E}[T_{\mathrm{CARD}}] \le K_{\mathrm{CARD}} + \frac{4\beta}{\alpha+\beta}\left(1+\frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + 2\mathsf{E}[S] + 44m_+ \max\left(\frac{\beta}{\alpha^2(\alpha+\beta)}, \sqrt{\frac{\beta}{\alpha^2\varepsilon(\alpha+\beta)}}, \frac{\log\frac{3}{2\beta\delta}}{\beta(\alpha+\beta)}, \sqrt{\frac{\log\frac{3}{2\beta\delta}}{\beta}}, \frac{\sqrt{\delta}\log\frac{3}{2\beta\delta}}{\alpha^2\beta^2\varepsilon}\right).$$

Proof sketch. It suffices to bound the work quantities on the right-hand side of (6.1).

- Lemma 6.4 implies $\mathsf{E}[W_s] \le c + q_A\mathsf{E}[W_s - c \mid W_s > c] \le c + \frac{q_A m_+}{\alpha}$.
- Lemmas 6.6 and 6.7 imply, after some simplification,

$$\mathsf{E}[W_\ell \mathcal{I}(W_s > c)] \le q_A\mathsf{E}[W_\ell] + \frac{q_A m_+}{\alpha^2} + \frac{4q_A q_B c}{\beta} + \frac{\sqrt{2q_A q_B m_+ c}}{\alpha\sqrt{\beta}}.$$

We use these with Lemmas 6.9 and 6.10 to express the right-hand side in terms of α, β, δ, and m_+, then simplify. We defer the details to Appendix B.3.

Proof of Theorem 3.3 for $n = 2$ servers. The bound follows directly from plugging the parameter choices into Theorem 6.11, and comparing with the lower bound in Theorem 3.1 implies heavy-traffic optimality. But the main question is why these are the right ways to set the parameters.

If we set $\delta = \Theta(\varepsilon^d)$ for fixed d, the only expression in Theorem 6.11 that is increasing as a function of d is $\log\frac{3}{2\beta\delta} = d\Theta\log\frac{1}{\varepsilon}$. We thus ignore factors of $\sqrt{\delta}$ when determining α and β. One can check at the end that $d > 3$ suffices.

Observe that we want $\beta/\alpha \downarrow 0$ to ensure the multiplier of $\mathsf{E}[W_{\mathrm{M/G/1}}]$ approaches K_{CARD}. If we substitute $\beta = \kappa\alpha$ into Theorem 6.11, then for any fixed κ, the resulting expression is a decreasing function of α, so we set $\alpha = \Theta(1)$. With this choice, the largest terms from the maximum in Theorem 6.11 are $\Theta\left(\sqrt{\beta/\varepsilon}\right)$ and $\Theta\left(\frac{1}{\beta}\log\frac{1}{\varepsilon}\right)$, which are balanced by $\beta = \Theta\left(\varepsilon^{1/3}\log\frac{1}{\varepsilon}^{2/3}\right)$.

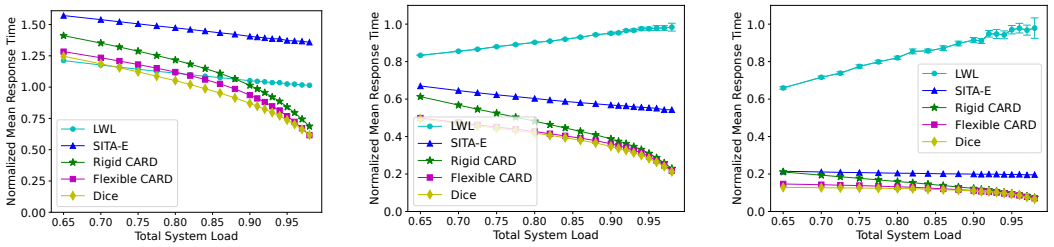
7 SIMULATIONS

We have established the optimality of CARD as load approaches capacity. In this section, we investigate the performance of CARD in moderate traffic via simulations. We aim to provide insights into the following questions with our simulations.

- How good is CARD's performance compared with other dispatching policies in the literature?
- Are there simple modifications of CARD that exhibit better performance in practice?
- CARD has three tunable parameters: c, α, and β. The recipe provided in Theorem 3.3 is optimal in heavy traffic, but are there rules of thumb that work well beyond heavy traffic? How sensitive is CARD's performance to these parameters?

In all of our simulations, we consider three benchmark policies: LWL, SITA-E[4] [17], and Dice [26]. Roughly, Dice lets the server with least work pick small jobs from the arrival stream, leaving the

[4]SITA-E is the version of SITA that splits the load equaly among all the servers. One can improve its performance very slightly by using an unbalanced load split, a policy known as SITA-O. But computing the optimal split is generally challenging [19], and in our initial experiments, SITA-O did not significantly improve upon SITA-E.



(a) 2 servers, cv = 1 (b) 2 servers, cv = 10 (c) 2 servers, cv = 100

Fig. 7.1. Normalized (relative to $\mathsf{E}[W_{\mathrm{M/G/1}}]$) mean response times for $n = 2$ servers.

large jobs for servers with more work. We refer interested readers to Appendix A for details.[5] Of course, there are many more dispatching policies. We pick LWL and SITA-E because they are extensively studied, and we pick Dice because among all heuristics for size- and state-aware dispatching, it has the best empirical performance at high load [26].

Our simulations include job size distributions with exponential and heavier tails. Heavy-tail distributions are common in computer systems and networks (e.g. [38]) and the high mean response times they incur make a good dispatching policy essential. Throughout this section, we consider three Weibull distributions with mean 1 and coefficients of variation (cv) 1, 10, and 100. We simulate 40 trials for each data point, with 10^7 arrivals per trial for cv = 1 and cv = 10, and 3×10^7 arrivals per trial for cv = 100. We show 95% confidence intervals when wider than the marker size.

7.1 Performance of CARD with Two Servers

Although CARD as introduced in Section 2.2 is heavy-traffic optimal, we can improve its performance under moderate traffic with one small modification: instead of statically deciding which server is short and which is long, dynamically treat whichever server has less work as the short server. We call this variant *Flexible CARD*, and call the original version *Rigid CARD* to disambiguate.

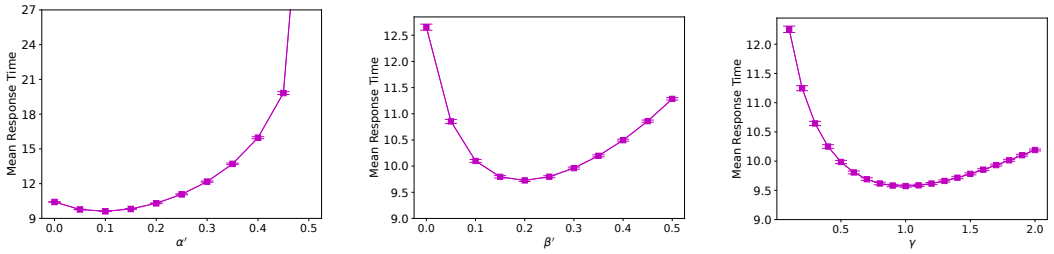
Figure 7.1 shows us that both CARD versions significantly outperform LWL and SITA-E, especially at high loads and with large coefficients of variation. For instance, with cv = 100 and $\rho = 0.98$, CARD gives a *93% reduction* compared to LWL, and a *61% reduction* compared to SITA-E. Flexible CARD is also almost tied with Dice at all loads simulated.

7.2 Calibrating the Parameters of Two-Server CARD

We now discuss how to calibrate parameters c, α, and β. In practice, α and β as prescribed in Theorem 3.3 are difficult to calibrate, because the ranges of α and β change as ρ increases. Therefore, we consider instead the parameters $\alpha' = \frac{1}{2} - \frac{\rho_s}{\rho}$ and $\beta' = \frac{1}{2} - \frac{\rho_\ell}{\rho}$. Adjusting α' can therefore be understood as adjusting the fraction of small jobs and adjusting β' can be understood as adjusting the fraction of large jobs.

After trying a few strategies for scaling c as a function of ρ, we found that thresholds of the form $c = \gamma \frac{1}{\sqrt{\varepsilon}} \log \frac{1}{\varepsilon}$, where γ depends on the distribution, yield decent performance.

[5]The version of Dice we simulate differs slightly from the original version in [26], where Dice have thresholds that do not vary with load. We notice that constant thresholds lead to suboptimal performance for either low or high loads. Therefore, we incorporate load-dependent thresholds for Dice that lead to good performance across all loads simulated. With two servers, we use a threshold of the form $\eta\varepsilon^{-1/3}$ for Dice, picking $\eta = 1.8, 5.2, 20$ for cv = 1, 10, 100, respectively. With ten servers, we use thresholds $2m_i\varepsilon^{-1/3}$.



(a) Varying α' ($\beta' = 0.15, \gamma = 0.6$) (b) Varying β' ($\alpha' = 0.15, \gamma = 0.6$) (c) Varying γ ($\alpha' = 0.15, \beta' = 0.15$)

Fig. 7.2. For each of the three plots, we fix two parameters and vary one parameter across a range of values. Size distribution simulated has cv = 10, and load is fixed at $\rho = 0.8$.

In general, for the three job size distributions we consider, mean response time under flexible CARD is not very sensitive to these parameters near the optima (see Figure 7.2). Any choice of parameters not too far from the optima yields decent performance. We found that $\alpha' = \beta' = 0.15$ for all three distributions and $\gamma = 0.3, 0.6, 2.5$ for cv = 1, 10, and 100, respectively, lead to decent performance. These are also the parameters we used in Figure 7.1.

7.3 Improving CARD's Performance for More than Two Servers

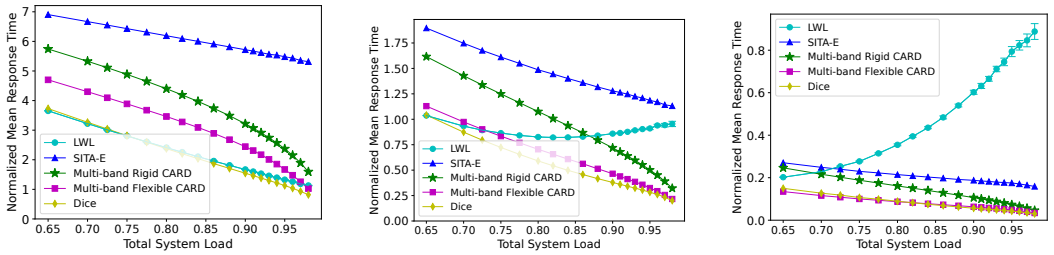
As the number of servers increases, flexible CARD with three parameters (γ, α', and β') no longer performs well for distributions with large coefficients of variation Figure C.1. Therefore, we propose another variant of CARD for n servers called multi-band CARD. We first present the general dispatching rules, then explain how multi-band rigid and flexible CARDs are defined.

- We divide the job size into $n+1$ small intervals such that each interval amounts to $\frac{1}{n}$ of the total load except for the first and last interval, each of which amounts to $\frac{1}{2n}$ of the total load. Denote the endpoints of these intervals as $0, m_1, \ldots m_n, \infty$.
- Server i except the last one has a threshold c_i, which can be different for different servers. When a job of size s arrives, it is dispatched according to the following general rules:
  - If s Ÿ m_1, it is dispatched to server 1.
  - If s ¡ m_n, it is dispatched to server n.
  - If $s \in [m_i, m_{i+1})$ for $i = 1, \ldots, n-1$, it is dispatched to server i if W_i 6 c_i. Otherwise, it is dispatched to server $i+1$.

Multi-band rigid CARD numbers the servers 1 to n and dispatches according to the rules outlined above. Server numbers do not change under rigid CARD. On the other hand, Multi-band flexible CARD sorts the servers in increasing work order when a job arrives so that W_1 6 W_2 6 $\cdots$ 6 W_n, then dispatch according to the general rules.

Since all the m_i's are fixed for each distribution, the tunable parameters are the c_i's. Our experiments show that we achieve good performance by setting $c_i = m_i/\sqrt{\varepsilon}$.

As we can see in Figure 7.3, multi-band CARDs significantly outperforms LWL and SITA-E at high loads and for job size with large coefficients of variation. When the job size distribution has cv=10, at $\rho = 0.98$, mean response time under flexible CARD is ~22% and ~19% of the mean response times under LWL and SITA-E, respectively. When the job size distribution has cv=100, at $\rho = 0.98$, mean response time under flexible CARD is ~4% and ~21% of the mean response times under LWL and SITA-E, respectively. Moreover, multi-band flexible CARD almost ties with Dice in all loads simulated.



(a) 10 servers, cv = 1 (b) 10 servers, cv = 10 (c) 10 servers, cv = 100

Fig. 7.3. Normalized (relative to $\mathsf{E}[W_{\mathrm{M/G/1}}]$) mean response times for $n = 10$ servers.

7.4 Tail Simulations

Although our paper focuses exclusively on mean response time analysis, metrics based on the *tail* of response time are often of interest in practice. As such, in this section, we conduct some simulations comparing the response time tails of CARD against the benchmark policies and provide some insights into the results.

We focus on a two-server system. The parameters of rigid CARD, flexible CARD, and Dice are the same as those in Section 7.2. The results are presented in Figure 7.4. We can see that for light-tail job size distribution, the tails of LWL and M/G/1/FCFS are better than those of rigid and flexible CARDs and Dice. On the other hand, for heavy-tail job size distribution, the tails of rigid and flexible CARDs and Dice are far better than those of LWL and M/G/1/FCFS up to 99-percentile of flexible CARD response time.

This result is not surprising. CARD and Dice both starve large jobs by making them wait in a long queue. For light-tail job size distributions, although giving a little priority to small jobs improves tail performance [15], starving large jobs in general only worsens tail performance [54]. For heavy-tail job size distributions, however, starving large jobs improves tail performance tremendously [54]. As is shown in Figure 7.4, CARD and Dice significantly outperform LWL and M/G/1/FCFS up to 99-percentile of flexible CARD response time for heavy-tail job size distributions.

7.5 Comparing CARD to Dice

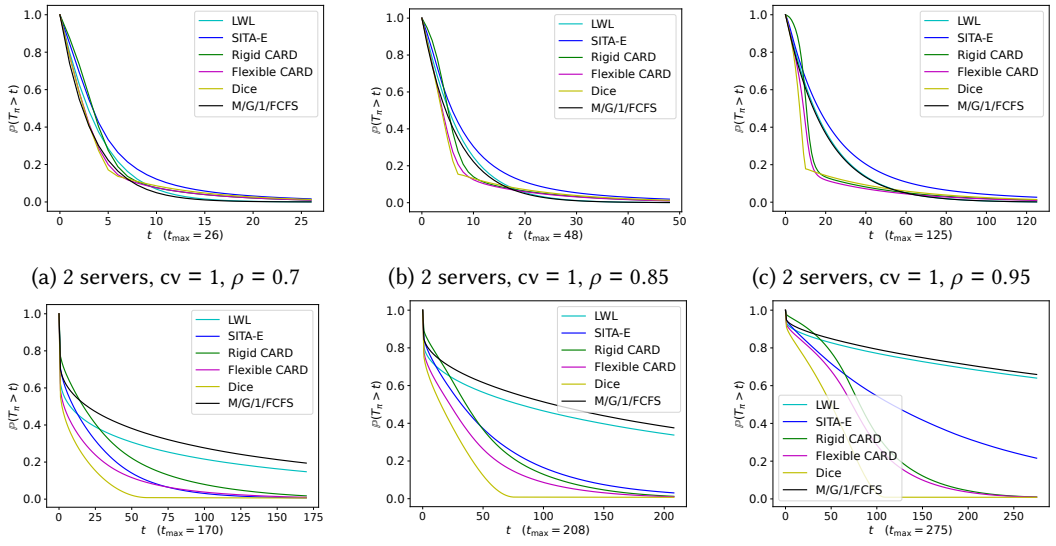
Given the excellent performance of Dice, we feel Dice warrants a more in-depth discussion. We refer interested readers to Appendix A, where we discuss the following questions:

- How complicated is Dice compared with CARD?
- Why does Dice perform so well in simulations?
- Is Dice heavy-traffic optimal?
- Why is it hard to analyze Dice?

The main takeaway is that Dice may *not* be heavy-traffic optimal, but it may be possible to modify Dice to make it heavy-traffic optimal. Dice remains a compelling option in practice that certainly deserves further study.

8 CONCLUSION

In this paper, we prove the first mean response time lower bound for FCFS servers. We design a new dispatching policy, called *CARD (Controlled Asymmetry Reduces Delay)*, and show that it is heavy-traffic optimal, thus making CARD the first proven heavy-traffic optimal size- and state-aware dispatching policy. CARD can thus serve as a new benchmark policy for future work in



(a) 2 servers, cv = 1, $\rho = 0.7$ (b) 2 servers, cv = 1, $\rho = 0.85$ (c) 2 servers, cv = 1, $\rho = 0.95$

(d) 2 servers, cv = 100, $\rho = 0.7$ (e) 2 servers, cv = 100, $\rho = 0.85$ (f) 2 servers, cv = 100, $\rho = 0.95$

Fig. 7.4. Response time tails for $n = 2$ servers shown above. The tails are plotted in $[0, t_{\max}]$, where $t_{\max}$ is chosen so that $\mathsf{P}[T_{\text{Flexible CARD}} \leq t_{\max}] \approx 0.99$.

dispatching or load-balancing for FCFS servers. Methodologically, our method of analyzing CARD using below-above cycles could be of independent interest, as it can be adapted to study other threshold-based policies.

Underlying our results is the insight that in the *size-aware* dispatching setting, it is helpful to have a significant imbalance between the amounts of work at each server. This insight has been made multiple times throughout the size-aware dispatching literature (e.g. Harchol-Balter et al. [17], Hyytiä et al. [24]). It is in contrast to the natural idea of always balancing the queues, which is helpful in size-oblivious dispatching [60].

In addition to minimizing mean response time, researchers today are also interested in tail performance. We conjecture that for job sizes with exponential tails and mean $1/\mu$, work under CARD asymptotically decays exponentially with rate $\frac{\mu-\lambda}{n}$ in an n-server system, which is worse (i.e. smaller) than the $\mu - \lambda$ decay rate of an M/M/1 under FCFS. An interesting follow-up question is how to balance the tradeoff between mean response time and response time decay rate, perhaps starting with the heavy-traffic regime. We leave this to future work.

ACKNOWLEDGMENTS

We thank Rhonda Righter and Esa Hyytiä for helpful discussions about optimal dispatching and the Dice policy. We also thank Onno Boxma and Ivo Adan for helpful pointers on the G/M/1, which featured in a previous version of our below-period analysis.

Isaac Grosof was supported by NSF grant no. CMMI-2307008, and a Tennenbaum Postdoctoral Fellowship at the Georgia Institute of Technology School of Industrial and Systems Engineering. Ziv Scully conducted this research in part while visiting the Simons Institute for the Theory of Computing, and in part while a FODSI postdoc at Harvard and MIT. He was supported by National Science Foundation grant nos. CMMI-2307008, DMS-2023528, and DMS-2022448.

REFERENCES

[1] Osman T Akgun, Rhonda Righter, and Ronald Wolff. 2013. Partial flexibility in routeing and scheduling. *Advances in Applied Probability* 45, 3 (2013), 673–691.

[2] Jonatha Anselmi. 2019. Combining size-based load balancing with round-robin for scalable low latency. *IEEE Transactions on Parallel and Distributed Systems* 31, 4 (2019), 886–896.

[3] Francois Baccelli and Pierre Brémaud. 2002. *Elements of queueing theory: Palm Martingale calculus and stochastic recurrences*. Vol. 26. Springer Science & Business Media.

[4] Yan Chen and Jing Dong. 2021. Scheduling with service-time information: The power of two priority classes. *arXiv preprint arXiv:2105.10499* (2021).

[5] DJ Daley. 1987. Certain optimality properties of the first-come first-served discipline for G/G/s queues. *Stochastic Processes and their Applications* 25 (1987), 301–308.

[6] Douglas G Down and Rong Wu. 2006. Multi-layered round robin routing for parallel servers. *Queueing Systems* 53 (2006), 177–188.

[7] Anthony Ephremides, Pravin Varaiya, and Jean Walrand. 1980. A simple dynamic routing problem. *IEEE transactions on Automatic Control* 25, 4 (1980), 690–693.

[8] Atilla Eryilmaz and Rayadurgam Srikant. 2012. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems* 72 (2012), 311–359.

[9] Hanhua Feng, Vishal Misra, and Dan Rubenstein. 2005. Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems. *Performance evaluation* 62, 1-4 (2005), 475–492.

[10] Sergey Foss, Seva Shneer, and Andrey Tyurlikov. 2012. Stability of a Markov-modulated Markov chain, with application to a wireless network governed by two protocols. *Stochastic Systems* 2, 1 (2012), 208–231.

[11] Sergei Georgievich Foss. 1980. Approximation of multichannel queueing systems. *Siberian Mathematical Journal* 21, 6 (1980), 851–857.

[12] Xinzhe Fu and Eytan Modiano. 2022. Joint Learning and Control in Stochastic Queueing Networks with Unknown Utilities. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–32.

[13] Steve W Fuhrmann and Robert B Cooper. 1985. Stochastic decompositions in the M/G/1 queue with generalized vacations. *Operations research* 33, 5 (1985), 1117–1129.

[14] Isaac Grosof, Ziv Scully, and Mor Harchol-Balter. 2019. Load balancing guardrails: keeping your heavy traffic on the road to low response times. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 2 (2019), 1–31.

[15] Isaac Grosof, Kunhe Yang, Ziv Scully, and Mor Harchol-Balter. 2021. Nudge: Stochastically improving upon FCFS. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 2 (2021), 1–29.

[16] Mor Harchol-Balter. 2013. *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press.

[17] Mor Harchol-Balter, Mark E Crovella, and Cristina D Murta. 1999. On choosing a task assignment policy for a distributed server system. *J. Parallel and Distrib. Comput.* 59, 2 (1999), 204–228.

[18] Mor Harchol-Balter, Alan Scheller-Wolf, and Andrew R Young. 2009. Surprising results on task assignment in server farms with high-variability workloads. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*. 287–298.

[19] Mor Harchol-Balter and Rein Vesilo. 2010. To balance or unbalance load in size-interval task allocation. *Probability in the Engineering and Informational Sciences* 24, 2 (2010), 219–244.

[20] Daniela Hurtado-Lange and Siva Theja Maguluri. 2022. A load balancing system in the many-server heavy-traffic asymptotics. *Queueing Systems* 101, 3-4 (2022), 353–391.

[21] Daniela Hurtado-Lange, Sushil Mahavir Varma, and Siva Theja Maguluri. 2022. Logarithmic heavy traffic error bounds in generalized switch and load balancing systems. *Journal of Applied Probability* 59, 3 (2022), 652–669.

[22] Esa Hyytiä. 2013. Lookahead actions in dispatching to parallel queues. *Performance Evaluation* 70, 10 (2013), 859–872.

[23] Esa Hyytiä, Peter Jacko, and Rhonda Righter. 2022. Routing with too much information? *Queueing Systems* 100, 3-4 (2022), 441–443.

[24] Esa Hyytiä, Aleksi Penttinen, and Samuli Aalto. 2012. Size-and state-aware dispatching problem with queue-specific job sizes. *European Journal of Operational Research* 217, 2 (2012), 357–370.

[25] Esa Hyytiä and Rhonda Righter. 2020. STAR and RATS: Multi-level dispatching policies. In *2020 32nd International Teletraffic Congress (ITC 32)*. IEEE, 81–89.

[26] Esa Hyytiä and Rhonda Righter. 2022. On Sequential Dispatching Policies. In *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 1–6.

[27] Esa Hyytiä and Rhonda Righter. 2023. On Dynamic Size-Aware Dispatching and Computation of the Optimal Actions. *SSRN* 4395052 (2023).

[28] Prakirt Raj Jhunjhunwala and Siva Theja Maguluri. 2021. Low-complexity switch scheduling algorithms: delay optimality in heavy traffic. *IEEE/ACM Transactions on Networking* 30, 1 (2021), 464–473.

[29] Prakirt Raj Jhunjhunwala and Siva Theja Maguluri. 2023. Heavy Traffic Queue Length Distribution without Resource Pooling in an Input-Queued Switch. *ACM SIGMETRICS Performance Evaluation Review* 50, 4 (2023), 26–28.

[30] Frank P Kelly and CN Laws. 1993. Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing systems* 13 (1993), 47–86.

[31] Gerhardus Martinus Koole. 1992. *On the optimality of FCFS for networks of multi-server queues*. Centre for Mathematics and Computer Science.

[32] Daniela Hurtado Lange and Siva Theja Maguluri. 2019. Heavy-traffic analysis of the generalized switch under multidimensional state space collapse. *ACM SIGMETRICS Performance Evaluation Review* 47, 2 (2019), 36–38.

[33] Xin Liu, Kang Gong, and Lei Ying. 2022. Steady-state analysis of load balancing with Coxian-2 distributed service times. *Naval Research Logistics (NRL)* 69, 1 (2022), 57–75.

[34] Zhen Liu and Rhonda Righter. 1998. Optimal load balancing on distributed homogeneous unreliable processors. *Operations Research* 46, 4 (1998), 563–573.

[35] Zhen Liu and Don Towsley. 1994. Optimality of the round-robin routing policy. *Journal of applied probability* 31, 2 (1994), 466–475.

[36] Siva Theja Maguluri, Sai Kiran Burle, and Rayadurgam Srikant. 2018. Optimal heavy-traffic queue length scaling in an incompletely saturated switch. *Queueing Systems* 88 (2018), 279–309.

[37] Siva Theja Maguluri and R Srikant. 2016. Heavy traffic queue length behavior in a switch under the MaxWeight algorithm. *Stochastic Systems* 6, 1 (2016), 211–250.

[38] Aniket Mahanti, Niklas Carlsson, Anirban Mahanti, Martin Arlitt, and Carey Williamson. 2013. A tale of the tails: Power-laws in internet measurements. *IEEE Network* 27, 1 (2013), 59–64.

[39] Sean P Meyn and Douglas Down. 1994. Stability of generalized Jackson networks. *The Annals of Applied Probability* (1994), 124–148.

[40] Sean P Meyn and Richard L Tweedie. 1993. Stability of Markovian processes II: Continuous-time processes and sampled chains. *Advances in Applied Probability* 25, 3 (1993), 487–517.

[41] Sean P Meyn and Richard L Tweedie. 1993. Stability of Markovian processes III: Foster–Lyapunov criteria for continuous-time processes. *Advances in Applied Probability* 25, 3 (1993), 518–548.

[42] Masakiyo Miyazawa. 1994. Rate conservation laws: a survey. *Queueing Systems* 15 (1994), 1–58.

[43] Sheldon M Ross. 1995. *Stochastic processes*. John Wiley & Sons.

[44] Sigurður Gauti Samúelsson and Esa Hyytiä. 2018. Applying reinforcement learning to basic routing problem. In *Queueing Theory and Network Applications: 13th International Conference, QTNA 2018, Tsukuba, Japan, July 25-27, 2018, Proceedings 13*. Springer, 238–249.

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

[46] Ziv Scully. 2022. *A New Toolbox for Scheduling Theory*. Ph. D. Dissertation. Carnegie Mellon University, Pittsburgh, PA. https://ziv.codes/pdf/scully-thesis.pdf

[47] Ziv Scully, Isaac Grosof, and Mor Harchol-Balter. 2020. The Gittins policy is nearly optimal in the M/G/k under extremely general conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3 (2020), 1–29.

[48] Yih-Choung Teh and Amy R Ward. 2002. Critical thresholds for dynamic routing in queueing networks. *Queueing Systems* 42 (2002), 297–316.

[49] Chang-Heng Wang, Siva Theja Maguluri, and Tara Javidi. 2017. Heavy traffic queue length behavior in switches with reconfiguration delay. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[50] Yinghui Wang and Douglas Down. 2014. On resource pooling in SITA-like parallel server systems. In *2014 26th International Teletraffic Congress (ITC)*. IEEE, 1–9.

[51] Richard R Weber. 1978. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability* 15, 2 (1978), 406–413.

[52] Wentao Weng, Xingyu Zhou, and R Srikant. 2020. Optimal load balancing with locality constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3 (2020), 1–37.

[53] Ward Whitt. 1993. Approximations for the GI/G/m queue. *Production and Operations Management* 2, 2 (1993), 114–161.

[54] Adam Wierman and Bert Zwart. 2012. Is tail-optimal scheduling possible? *Operations research* 60, 5 (2012), 1249–1257.

[55] Wayne Winston. 1977. Optimality of the shortest line discipline. *Journal of applied probability* 14, 1 (1977), 181–189.

[56] Ronald W Wolff. 1982. Poisson arrivals see time averages. *Operations research* 30, 2 (1982), 223–231.

[57] Runhan Xie and Ziv Scully. 2023. Reducing heavy-traffic response time with asymmetric dispatching. *ACM SIGMETRICS Performance Evaluation Review* 51, 2 (2023), 36–38.

[58] Xingyu Zhou, Jian Tan, and Ness Shroff. 2018. Flexible load balancing with multi-dimensional state-space collapse: Throughput and heavy-traffic delay optimality. *Performance Evaluation* 127-128 (2018), 176–193.
[59] Xingyu Zhou, Jian Tan, and Ness Shroff. 2018. Heavy-traffic delay optimality in pull-based load balancing systems: Necessary and sufficient conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 3 (2018), 1–33.
[60] Xingyu Zhou, Fei Wu, Jian Tan, Kannan Srinivasan, and Ness Shroff. 2018. Degree of queue imbalance: Overcoming the limitation of heavy-traffic delay optimality in load balancing systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 1–41.
[61] Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. 2017. Designing low-complexity heavy-traffic delay-optimal load balancing schemes: Theory to algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 2 (2017), 1–30.

A DICE

In this section, we give a brief introduction of Dice based on Hyytiä and Righter [26] and answer some questions readers may be curious about.

What is Dice? Let $\mathsf{W} = (W_1, \ldots, W_n)$ be the workload vector. Dice has $n-1$ threshold parameters $\tau_1, \ldots, \tau_{n-1}$ to set. When a job of size s arrives, Dice dispatches the job as follows:

- Sort the workload vector in increasing order. We assume that τ_i is the threshold on W_i.
- The dispatcher goes through W in increasing order and dispatches the job to the first server i that satisfies $\tau_i - W_i \ ¡ \ s$. If none of the first $n-1$ servers satisfy this condition, the job is dispatched to the last server.

How complicated is Dice compared with CARD? Dice is easier to implement than CARD. To implement Dice, one only needs to find decent threshold parameters $\tau_1, \ldots, \tau_{n-1}$. Successful implementation of CARD, however, requires tuning thresholds on job sizes (i.e. m_- and m_+ for two-server CARD and m_i's for multi-band CARD) in addition to the threshold parameters.

Why does Dice perform so well in simulations? Dice has the same spirit as CARD: it keeps one long queue populated mostly with large jobs and maintains short queues for small jobs to get through quickly, thereby increasing $\mathsf{E}[S_{\text{queue}}]$.

Is Dice heavy-traffic optimal? This is an open problem. We conjecture that Dice may *not* be heavy-traffic optimal. Consider a two-server system. Under Dice, only jobs that fit into the gap between current work at the shorter server and the threshold get dispatched to the shorter server. However, this will not guarantee that as $\varepsilon \downarrow 0$, all jobs with size less than m gets dispatched to the shorter server, as there is always some probability that the gap is too small for a job with size less than m.

Why is it hard to analyze Dice? Even with the tools we developed in this paper, Dice is more difficult to analyze than CARD. The main reason is that the dispatching policy under Dice changes continuously with the state of the shorter queue, which makes it challenging to directly apply our below-above cycle analysis. We expect that a refinement of our approach could work for Dice, but substantial extra work is needed.

B DEFERRED PROOFS

B.1 Suboptimality of LWL and SITA in heavy-traffic

In this section, we show that in a two-server system, neither LWL nor SITA is heavy-traffic optimal. Let

$$K_\pi = \lim_{\varepsilon \downarrow 0} \frac{\mathsf{E}[T_\pi]}{\mathsf{E}[W_{\text{M/G/1}}]},$$

be the *heavy-traffic constant* of policy π. In Theorem B.1 below, we show that K_{LWL} and $K_{\text{SITA-E}}$ are strictly greater than K_{CARD} as defined in (2.1).

However, this does not finish the story for SITA, because while SITA-E is the version of SITA that splits the load equally, it is possible to improve SITA's performance by using an unequal load split. The version of SITA that uses the optimal load split is known as SITA-O. Surprisingly, SITA-O can have a significantly better heavy-traffic constant than SITA-E, even though there is very little flexibility in the amount of load each server can receive. In Theorem B.2, we sketch a computation of $K_{\text{SITA-O}}$, showing that it, too, is strictly greater than K_{CARD}.

For simplicity of computations, we focus on $n = 2$ servers, but the results should generalize to more servers. Similarly, while we continue to assume continuous job size distribution S (Section 2.1) for simplicity of defining SITA, the results should hold for any job size distribution for which m is well-defined.[6]

Theorem B.1. *In a system with $n = 2$ servers and and a continuous job size distribution S, we have $K_{\text{LWL}} > K_{\text{CARD}}$ and $K_{\text{SITA-E}} > 2K_{\text{CARD}}$.*

Proof. It is known that $K_{\text{LWL}} = 1$ (see e.g. [53]). To compute $K_{\text{SITA-E}}$, we first note that under SITA-E, the system decouples to two independent M/G/1 queues. For any $\varepsilon \in (0, 1)$, we have

$$\begin{aligned}
\mathsf{E}[T_{\text{SITA-E}}] &= 2\,\frac{\lambda \mathsf{P}[S \ddot{Y} m]\mathsf{E}[S^2 \mid S \ddot{Y} m]}{\varepsilon}\mathsf{P}[S \ddot{Y} m] + \frac{\lambda \mathsf{P}[S > m]\mathsf{E}[S^2 \mid S > m]}{\varepsilon}\mathsf{P}[S > m] \quad \text{(B.1)}\\
&\quad + 2\mathsf{E}[S]\\
&= 2\,\frac{\lambda \mathsf{E}[S^2 \mathcal{I}(S \ddot{Y} m)]}{\varepsilon}\mathsf{P}[S \ddot{Y} m] + \frac{\lambda \mathsf{E}[S^2 \mathcal{I}(S > m)]}{\varepsilon}\mathsf{P}[S > m] + \mathsf{E}[S]\\
&\overset{\text{(a)}}{>} \frac{2\lambda \mathsf{E}[S^2]}{\varepsilon}\mathsf{P}[S > m] + 2\mathsf{E}[S]\\
&= 4\mathsf{P}[S > m]\mathsf{E}[W_{\text{M/G/1}}] + 2\mathsf{E}[S],
\end{aligned}$$

where (a) follows from the fact that $\mathsf{P}[S \ddot{Y} m] > \mathsf{P}[S > m]$. Looking at the $\varepsilon \downarrow 0$ limit, we have $K_{\text{SITA-E}} > 2K_{\text{CARD}}$.

Theorem B.2. *In a system with $n = 2$ servers and a continuous job size distribution S, we have $K_{\text{SITA-O}} > K_{\text{CARD}}$.*

Proof sketch. SITA-O works like SITA-E, except instead of using size threshold m to split jobs between the servers, it uses a different size threshold m'. The key insight is that in the $\varepsilon \downarrow 0$ limit, we must have $m' - m \leq O(\varepsilon)$, beause otherwise we would overload one of the servers. This means, roughly speaking, that SITA-O can affect the denominators in (B.1), but it cannot significantly affect the numerators. Specifically, there exists $x \in (-1, 1)$ such that

$$\begin{aligned}
\mathsf{E}[T_{\text{SITA-O}}] &= 2\,\frac{\lambda \mathsf{P}[S \ddot{Y} m]\mathsf{E}[S^2 \mid S \ddot{Y} m]}{\varepsilon(1-x)}\mathsf{P}[S \ddot{Y} m] + \frac{\lambda \mathsf{P}[S > m]\mathsf{E}[S^2 \mid S > m]}{\varepsilon(1+x)}\mathsf{P}[S > m] \pm O(1)\\
&= \frac{2\lambda}{\varepsilon}\,\frac{\mathsf{E}[S^2 \mathcal{I}(S \ddot{Y} m)]\mathsf{P}[S \ddot{Y} m]}{1-x} + \frac{\mathsf{E}[S^2 \mathcal{I}(S > m)]\mathsf{P}[S > m]}{1+x} \pm O(1).
\end{aligned}$$

[6]When the distribution has an atom at m, there are corner cases where $K_{\text{SITA-O}} = K_{\text{CARD}}$. One example is when $S \in \{a, b\}$ with probability 1 such that $a\mathsf{P}[S = a] = b\mathsf{P}[S = b]$ [57].

Optimizing over the value of x yields

$$\begin{aligned} K_{\text{SITA-O}} &= \frac{2}{\mathsf{E}[S^2]}\left(\sqrt{\mathsf{E}[S^2\mathcal{I}(S \ddot{Y} m)]\mathsf{P}[S \ddot{Y} m]} + \sqrt{\mathsf{E}[S^2\mathcal{I}(S > m)]\mathsf{P}[S > m]}\right)^2 \\ &\overset{(a)}{>} \frac{2\mathsf{P}[S > m]}{\mathsf{E}[S^2]}\left(\sqrt{\mathsf{E}[S^2\mathcal{I}(S \ddot{Y} m)]} + \sqrt{\mathsf{E}[S^2\mathcal{I}(S > m)]}\right)^2 \\ &= K_{\text{CARD}}\left(1 + \frac{2\sqrt{\mathsf{E}[S^2\mathcal{I}(S \ddot{Y} m)]\mathsf{E}[S^2\mathcal{I}(S > m)]}}{\mathsf{E}[S^2]}\right), \end{aligned}$$

where (a) follows from the fact that $\mathsf{P}[S \ddot{Y} m] > \mathsf{P}[S > m]$.

B.2 Stability

As outlined in Section 5, we begin by showing that the short server is stable for any threshold $c > 0$. Our main tool is a continuous-time Foster-Lyapunov theorem developed in Meyn and Tweedie [41]. A key component of the theorem is the *infinitesimal generators* for Markov processes. Let $X(t)$ be a Markov process, its infinitesimal generator, $\mathscr{A}$, is the operator defined by

$$\mathscr{A}V(x) = \lim_{t \downarrow 0} \frac{\mathsf{E}[V(X(t)) \mid X(0) = x] - V(x)}{t}.$$

The domain of $\mathscr{A}$ is all functions V for which the limit on the right exists for all x in the state space. Since work at the short server, $W_s(t)$, is a Markov process, for a function V with left derivative, we may explicitly derive the infinitesimal generator of $W_s(t)$ under CARD:

$$\begin{aligned} \mathscr{A}V(w_s) = -\tfrac{1}{2}V'(w_s)\mathcal{I}(w_s \ddot{i} 0) + \lambda p_s \mathsf{E}_{S_s}[V(w_s + S_s) - V(w_s)] \\ + \mathcal{I}(w_s \ddot{6} c)\lambda p_m \mathsf{E}_{S_m}[V(w_s + S_m) - V(w_s)], \end{aligned}$$

where $p_s = \mathsf{P}[S \ddot{6} m]$, $p_m = \mathsf{P}[m_- \ddot{Y} S \ddot{Y} m_+]$, $\mathsf{E}_{S_s}[\cdot]$ is the expectation over the distribution of small jobs (i.e. $S \mid S \ddot{6} m_-$) and $\mathsf{E}_{S_m}[\cdot]$ is the expectation over the distribution of medium jobs (i.e. $S \mid m_- \ddot{Y} S \ddot{Y} m_+$), $\mathcal{I}(\cdot)$ is the indicator function, and V' is the left derivative.

We now present the continuous-time Foster-Lyapunov theorem [41, Theorem 4.4] below for easy reference.

Theorem B.3. *Suppose that a Markov process Φ is a non-explosive right process. If there exists constants $c, d \ddot{i} 0$, a function $f > 1$, a closed petite set C, and a function $V > 0$ that is bounded on C such that for all $x \in O_m$ and $m \in \mathsf{Z}$,*

$$\mathscr{A}_m V(x) \ddot{6} -\eta f(x) + d\mathcal{I}_C(x),$$

then Φ is positive Harris recurrent.

Here, O_m is a family of precompact sets that increases to the entire state space as $m \to \infty$ and $\mathscr{A}_m$ is the generator for the truncated process restricted to O_m. This restriction is in place mainly to handle possibly explosive processes. Our process $\mathsf{W}(t)$ is not explosive. More importantly, the Lyapunov function V we consider in Lemma 5.1 is increasing and differentiable. It follows that $\mathscr{A}_m V(x) \ddot{6} \mathscr{A}V(x)$ for all x. It therefore suffices for us to apply theorem Theorem B.3 with $\mathscr{A}V(x)$ instead.

Lemma 5.1. *W_s is Harris ergodic for any $\varepsilon \ddot{i} 0$.*

Proof. We first check the preconditions of Theorem B.3 hold for W_s under CARD for any $\varepsilon \ddot{i} 0$. W_s is obviously non-explosive. Let $V(W_s) = W_s$, $C = \{W_s : W_s \ddot{6} c\}$, and $f(w_s) \equiv 1$. Then we have

$$\mathscr{A}V(W_s) = \beta\mathcal{I}(W_s \ddot{6} c) - \alpha\mathcal{I}(W_s \ddot{i} c)$$

Since $\alpha > 0$ for any $\varepsilon > 0$, positive Harris recurrence of $W_s(t)$ follows from Theorem B.3 if C is a closed petite set. We now check this.

It follows from Meyn and Tweedie [41, Theorem 3.1] that $W_s(t)$ is non-evanescent. By Meyn and Tweedie [40, Theorem 4.2], the K_a-chain of $W_s(t)$ is an irreducible T-process with everywhere nontrivial continuous component. By Meyn and Tweedie [40, Theorem 4.1(i)], C is a petite set.

Given positive recurrence of $W_s(t)$ and Meyn and Tweedie [40, Theorem 4.2], we conclude from Meyn and Down [39, Theorem 3.2] that $W_s(t)$ is also ergodic.

Lemma 5.2. *Suppose $\theta > 0$ satisfies $(S^{\check{Z}}_{s,m})_e(\theta) < \frac{1}{n(n-1)\beta+n-1}$, where $(S^{\check{Z}}_{s,m})_e(\cdot)$ is the Laplace transform of the equilibium distribution of the size of small and medium jobs, $S_{s,m} = (S \mid S < m_+)$. Then for all $x \in [0, c]$,*

$$\mathsf{P}[W_s < c - x] \le \frac{(n(n-1)\beta + n - 1)(S^{\check{Z}}_{s,m})_e(\theta)}{(n(n-1)\beta + n - 1)(S^{\check{Z}}_{s,m})_e(\theta) - 1} e^{-\theta x}.$$

Proof. Define $V(w_s) = (c - w_s)^+$ and fix some $\theta > 0$. Since W_s has a stationary distribution, we can apply the rate conservation law [42] to $e^{\theta V(W_s)}$, which yields

$$\frac{\theta}{n}\mathsf{E}_\pi[e^{\theta V(W_s)}\mathcal{I}(V(W_s) < c)] + \lambda_{s.m}\mathsf{E}_{\pi,S_{s,m}}[e^{\theta V(W_s+)} - e^{\theta V(W_s)}] = 0.$$

Here, π is the stationary distribution of W_s and $\lambda_{s,m}$ is the arrival rate into a short server from small and medium jobs, and $V(W_s+)$ is the value of $V(W_s)$ immediately after a job arrival of size $S_{s,m}$. Rearranging yields

$$\frac{\theta}{n}\mathsf{E}_\pi[e^{\theta V(W_s)}] + \lambda_{s,m}\mathsf{E}_{\pi,S_{s,m}}[e^{\theta V(W_s+)} - e^{\theta V(W_s)}] = \frac{\theta}{n}\mathsf{E}_\pi[e^{\theta V(W_s)}\mathcal{I}(V(W_s) = c)].$$

We drop the RHS and work with

$$\frac{\theta}{n}\mathsf{E}_\pi[e^{\theta V(W_s)}] + \lambda_{s,m}\mathsf{E}_{\pi,S_{s,m}}[e^{\theta V(W_s+)} - e^{\theta V(W_s)}] > 0. \tag{B.2}$$

We first analyze the second term on the LHS. Conditioning on a given state W_s, we have

$$\begin{aligned}
&\mathsf{E}_{S_{s,m}}\left[e^{\theta V(W_s+)} - e^{\theta V(W_s)} \mid W_s\right] \\
&\overset{(a)}{=} \mathsf{E}_{S_{s,m}}\left[e^{\theta(V(W_s)-S_{s,m})^+} - e^{\theta V(W_s)} \mid W_s\right] \\
&= \mathsf{E}_{S_{s,m}}\left[e^{\theta(V(W_s)-S_{s,m})^+} - e^{\theta(V(W_s)-S_{s,m})} + e^{\theta(V(W_s)-S_{s,m})} - e^{\theta V(W_s)} \mid W_s\right] \\
&\overset{(b)}{=} \mathsf{E}_{S_{s,m}}\left[e^{\theta(V(W_s)-S_{s,m})^+} - e^{\theta(V(W_s)-S_{s,m})} \mid W_s\right] + e^{\theta V(W_s)}\mathsf{E}_{S_{s,m}}\left[(e^{-\theta S_{s,m}} - 1)\right] \\
&\overset{(c)}{=} \mathsf{E}_{S_{s,m}}\left[(1 - e^{\theta(V(W_s)-S_{s,m})})\,\mathcal{I}(V(W_s) < S_{s,m}) \mid W_s\right] + e^{\theta V(W_s)}(S_{s,m}(\theta) - 1),
\end{aligned} \tag{B.3}$$

where

(a) follows from $V(W_s+) = (c - W_s - S_{s,m})^+ = (V(W_s) - S_{s,m})^+$,
(b) follows from the independence of the arriving job size and $V(W_s)$ for any given $W_s \le c$, and
(c) follows from the definition of LST and the fact that

$$e^{\theta(V(W_s)-S_{s,m})^+} - e^{\theta(V(W_s)-S_{s,m})} = \begin{cases} 0, & \text{if } V(W_s) > S_{s,m} \\ 1 - e^{\theta(V(W_s)-S_{s,m})}, & \text{if } V(W_s) < S_{s,m}. \end{cases}$$

Taking expectation over π on both sides of (B.3) and substituting into (B.2) gives

$$\begin{aligned}\frac{\theta}{n}\mathsf{E}_\pi[e^{\theta V(W_s)}] &> \lambda_{s,m}\mathsf{E}_\pi\left[e^{\theta V(W_s)}\left(1-S_{s,m}^{\mathsf{G}}(\theta)\right)\right]-\lambda_{s,m}\mathsf{E}_{\pi,S_{s,m}}\left[\left(1-e^{\theta(V(W_s)-S_{s,m})}\right)\mathcal{I}(V(W_s)\ddot{Y}S_{s,m})\right]\\ &\overset{(d)}{=}\theta\left(\frac{n-1}{n}+(n-1)\beta\right)\mathsf{E}_\pi\left[e^{\theta V(W_s)}\right](S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)\\ &\quad-\theta\left(\frac{n-1}{n}+(n-1)\beta\right)\mathsf{E}_{\pi,S_{s,m}}\left[\frac{1-e^{\theta(V(W_s)-S_{s,m})}}{\theta\mathsf{E}[S_{s,m}]}\mathcal{I}(V(W_s)\ddot{Y}S_{s,m})\right],\end{aligned}\tag{B.4}$$

where (d) follows from $\frac{n-1}{n}+(n-1)\beta=\lambda_{s,m}\mathsf{E}[S_{s,m}]$, which is the load of the short and medium jobs, and the fact that

$$(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)=\frac{1-S_{s,m}^{\mathsf{G}}(\theta)}{\theta\mathsf{E}[S_{s,m}]},$$

which holds for a general job size distribution (with or without a density function). See e.g. Ross [43].

Since

$$1-e^{-\theta\left(S_{s,m}-V(W_s)\right)}\ \mathcal{I}(V(W_s)\ddot{Y}S_{s,m})\le 1-e^{-\theta S_{s,m}},$$

we have

$$\mathsf{E}_{\pi,S_{s,m}}\left[\frac{1-e^{\theta(V(W_s)-S_{s,m})}}{\theta\mathsf{E}[S_{s,m}]}\mathcal{I}(V(W_s)\ddot{Y}S_{s,m})\right]\le(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta).$$

Since $\theta>0$ is chosen so that $(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)<\frac{1}{n(n-1)\beta+n-1}$, we have $\frac{n-1}{n}+(n-1)\beta\,(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)-\frac{1}{n}<0$. Thus, we rearrange (B.4) to obtain

$$\mathsf{E}_\pi[e^{\theta V(W_s)}]\le\frac{(n(n-1)\beta+n-1)(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)}{(n(n-1)\beta+n-1)(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)-1}.$$

Markov's inequality then gives

$$\mathsf{P}[V(W_s)\ddot{Y}x]\le\frac{(n(n-1)\beta+n-1)(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)}{(n(n-1)\beta+n-1)(S_{s,m}^{\check{\mathsf{Z}}})_e(\theta)-1}e^{-\theta x},$$

and the lemma follows.

Theorem 3.2. *Let $\delta>0$, and consider CARD with threshold*

$$c=\frac{n(n-1)m_+}{\beta}\log\frac{n+1}{n\beta\delta}.$$

Then,

(a) Each short server satisfies $\mathsf{P}[W_s=0]\le\delta$.
(b) If $\delta\ddot{<}\frac{n}{n-1}\varepsilon$, then the system is stable. Specifically, the set $\{(0,\dots,0)\}$ is positive recurrent for the process $\mathsf{W}(t)=(W_1(t),\dots,W_n(t))$.

Proof. Part (a) is a corollary of Lemma 5.3. For part (b), we first establish the result for $n=2$ servers, then show how it generalizes to $n>2$ servers. For $n=2$, we denote the state as $\mathsf{W}(t)=(W_s(t),W_\ell(t))$.

To establish (b), we first apply Foss et al. [10, Theorem 1] to the pre-jump chain $\{\mathsf{W}(T_n-)\}$, where T_n is the arrival time of the nth job. Conditions A1-A3 in Theorem 1 are fulfilled by Lemma 5.1. Define

$$L_2(w_\ell)=\lambda w_\ell,\qquad f(w_s)=-\tfrac{1}{2}+\rho_\ell+\rho_m\mathcal{I}(w_s>c),\qquad h(x)=\tfrac{1}{2}e^{-x}.$$

We now verify conditions B1 and B2 are met with above choices of L_2, f, and h.

Condition B1:

$$\sup_{w_s, w_\ell} \mathsf{E}[|L_2(W_\ell(T_1-)) - L_2(w_\ell)| \mid W_\ell(0-) = w_\ell, W_s(0-) = w_s, \text{Arrival at time } 0] \leq 1.$$

Condition B2: Let π_s be the stationary distribution of $W_s(t)$. By PASTA, π_s is also the stationary distribution of $\{W_s(T_n-)\}$. We have

$$\begin{aligned}\mathsf{E}_{\pi_s}[f(W_s)] &= -\tfrac{1}{2} + \rho_\ell + \rho_m \mathsf{P}[W_s > c] \\ &\overset{(a)}{\leq} -\tfrac{1}{2} + \rho_\ell + \rho_m \frac{(\rho_m + \rho_s) - \frac{1}{2} + \frac{\delta}{2}}{\rho_m} \\ &= -1 + \rho + \frac{\delta}{2} = -\varepsilon + \frac{\delta}{2} < 0,\end{aligned}$$

where (a) comes from Lemma 6.10 and PASTA. We then compute[7]

$$\begin{aligned}&\mathsf{E}[L_2(W_\ell(T_1-)) - L_2(w_\ell)) \mid W_\ell(0-) = w_\ell, W_s(0-) = w_s, \text{Arrival at time } 0] \\ &\quad = \lambda \mathsf{E}\left[\int_0^{T_1-} -\tfrac{1}{2}\mathcal{I}(W_\ell(t) > 0)\,\mathrm{d}t \;\middle|\; W_\ell(0-) = w_\ell, W_s(0-) = w_s, \text{Arrival at time } 0\right] \\ &\qquad + \rho_\ell + \rho_m \mathcal{I}(w_s > c) \\ &\quad = f(w_s) + \frac{\lambda}{2}\mathsf{E}\left[(T_1 - (w_\ell + S))^+ \;\middle|\; W_\ell(0-) = w_\ell, W_s(0-) = w_s, \text{Arrival at time } 0\right] \\ &\quad \leq f(w_s) + \frac{\lambda}{2}\mathsf{E}\left[(T_1 - w_\ell)^+\right] = f(w_s) + \frac{\lambda}{2}\frac{1}{\lambda}e^{-\lambda w_\ell} = f(w_s) + h(L_2(w_\ell)).\end{aligned}$$

It now follows from Foss et al. [10, Theorem 1] that the embedded pre-jump chain $\{\mathbf{W}(T_n-)\}$ is positive Harris recurrent.

Since $\{\mathbf{W}(T_n-)\}$ is positive Harris recurrent and easily seen to be $\{(0,0)\}$-irreducible, the expected number of steps until returning to $(0,0)$ is finite from any starting state. The time between steps is exponentially distributed with mean $1/\lambda$, so we conclude from Wald's equation that the expected return time of the original process $\mathbf{W}(t)$ to state $(0,0)$ is also finite. Positive Harris recurrence of $\mathbf{W}(t)$ immediately follows. We now generalize the above proof to $n > 2$ servers. To begin with, we define a vector-valued process $\mathbf{W}_{\text{short servers}}(t) = (W_{s_1}(t), \ldots, W_{s_{n-1}}(t))$. Under multiserver CARD, $W_{\text{short servers}}(t)$ has the following properties:

- $\mathbf{W}_{\text{short servers}}(t)$ is a Markov process of its own and is Harris ergodic.
- Since stationary distribution of the short servers are i.i.d., the stationary distribution of $\mathbf{W}_{\text{short servers}}$ is the product of stationary distributions of the short servers in isolation.

With these two properties in hand, the argument for $n = 2$ servers as presented above works for $n > 2$ servers with the same functions h, L_2, and the following f:

$$f(\mathbf{w}_{\text{short servers}}) = -\frac{1}{n} + \rho_\ell + \frac{\rho_m}{n-1}\sum_{i=1}^{n-1} \mathcal{I}(w_{s_i} > c).$$

B.3 Response Time Analysis

Lemma 6.4.

$$\mathsf{E}[W_s - c \mid W_s > c] \leq \frac{m_+}{4\alpha} \qquad \text{and} \qquad \mathsf{E}[(W_s - c)^2 \mid W_s > c] \leq \frac{m_+^2}{8\alpha^2}$$

[7]The conditional probabilities below are a slight abuse of notation. They should be understood as referring to the probability measure induced by the pre-jump Markov chain starting from state (w_s, w_ℓ).

Proof. As stated in the proof sketch, $(W_s - c \mid W_s \ i \ c)$ has the same distribution as an M/G/1 with vacations.

- The job size distribution is $S_s = (S \mid S \ddot{Y} m_-)$. In particular, using the fact that S_s is stochastically dominated by m_+, one can show that $(S_s)_e$ is stochastically dominated by a uniform distribution on $[0, m_+]$.[8]
- The load is $1 - 2\alpha$, and so the slackness is 2α.
  - The reason we use 2α instead of α is because the server operates at speed $\frac{1}{2}$. By "doubling the clock speed", the server speed becomes 1, and the distribution of $(W_s - c \mid W_s \ i \ c)$ is unaffected. This makes it easy to apply standard results about the M/G/1 with vacations.
- Let U denote the vacation length distribution. It is hard to characterize exactly, but because $W_s - c \ 6 \ m_+$ at the start of an above period, U_e is stochastically dominated by a uniform distribution on $[0, m_+]$.

The desired bounds follow from the work decomposition formula for the M/G/1 with vacations [13]. Specifically, for an M/G/1 with vacations, we can write its steady-state work $W_{\mathrm{M/G/1/vac}}$ as an independent sum of random variables with distributions $W_{\mathrm{M/G/1}}$ and U_e. This means

$$\mathsf{E}[W_s - c \mid W_s \ i \ c] = \mathsf{E}[W_{\mathrm{M/G/1/vac}}] = \mathsf{E}[W_{\mathrm{M/G/1}}] + \mathsf{E}[U_e],$$
$$\mathsf{E}[(W_s - c)^2 \mid W_s \ i \ c] = \mathsf{E}[W^2_{\mathrm{M/G/1/vac}}] = \mathsf{E}[W^2_{\mathrm{M/G/1}}] + 2\mathsf{E}[W_{\mathrm{M/G/1}}]\mathsf{E}[U_e] + \mathsf{E}[U_e^2].$$

Applying the PK formula with the relevant parameters, we obtain

$$\mathsf{E}[W_{\mathrm{M/G/1}}] = \frac{(1-2\alpha)\mathsf{E}[(S_s)_e]}{2\alpha} \ 6 \ \frac{(1-2\alpha)m_+}{4\alpha},$$
$$\mathsf{E}[W^2_{\mathrm{M/G/1}}] \ 6 \ \frac{(3 - 4\alpha(2-\alpha))m_+^2}{24\alpha^2}.$$

The result then follows from $\mathsf{E}[U_e] \ 6 \ \frac{m_+}{2}$ and $\mathsf{E}[U_e^2] \ 6 \ \frac{m_+^3}{3}$.

Lemma 6.7.

$$\mathsf{E}[W_\ell \mathcal{I}(W_s \ i \ c)] - q_A \mathsf{E}^0_c[W_\ell(0)] \ 6 \ q_A \mathsf{E}[A_e] + 2\sqrt{q_A \mathsf{E}[A_e] \, q_B \mathsf{E}[B_e]} \ 6 \ \frac{q_A m_+}{4\alpha^2} + \frac{\sqrt{2 q_A q_B m_+ c}}{\alpha \sqrt{\beta}}.$$

Proof. The proof is very similar to that of Lemma 6.6, so we give only the key steps. Applying the Palm inversion formula [3] to $W_\ell \mathcal{I}(W_s \ i \ c)$ gives

$$\mathsf{E}[W_\ell \mathcal{I}(W_s \ i \ c)] = \frac{1}{\mathsf{E}[A+B]} \mathsf{E}^0_c \left[\int_B^{A+B} W_\ell(t) \, \mathrm{d}t \right],$$

where we can start the integral at B and remove the indicator because $W_s(t) \ i \ c$ exactly during above periods, which corresponds to $t \in [B, A+B)$. Expanding this using (6.6) and noting the independence of $W_\ell(0)$ from the below-above cycle, we obtain

$$\mathsf{E}[W_\ell \mathcal{I}(W_s \ i \ c)] - \frac{\mathsf{E}[A]}{\mathsf{E}[A+B]} \mathsf{E}^0_c[W_\ell(0)]$$
$$= \frac{1}{\mathsf{E}[A+B]} \mathsf{E}^0_c \left[\int_B^{A+B} \max\left(-\Delta_\ell(0,t) + \Sigma^m_\ell(0,t) + \Sigma^\ell_\ell(0,t), -W_\ell(0) \right) \mathrm{d}t \right].$$

[8] It is not in general true that S being dominated by R implies S_e is dominated by R_e. This is specific to the case that R is a deterministic constant.

Applying (6.5) to the left-hand side, we see it suffices to give bounds on the right-hand side. The same reasoning as the proof Lemma 6.6 yields

$$\mathsf{E}[W_\ell \mathcal{I}(W_s < c)] - q_A \mathsf{E}_c^0[W_\ell(0)] \le \frac{1}{\mathsf{E}[A+B]} \mathsf{E}_c^0 \left[\int_B^{A+B} t \, dt \right] = \frac{\mathsf{E}[(A+B)^2 - B^2]}{2\mathsf{E}[A+B]}.$$

The result then follows from a computation similar to the end of the proof of Lemma 6.6.

Theorem 6.11. *In a system with $n = 2$ servers, if $\delta \le \varepsilon < \frac{1}{2}$ and $\beta > 2\delta$, then by setting c according to Theorem 3.2, CARD achieves mean response time bounded by*

$$\mathsf{E}[T_{\mathrm{CARD}}] \le K_{\mathrm{CARD}} + \frac{4\beta}{\alpha+\beta}\left(1 + \frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + 2\mathsf{E}[S] + 44 m_+ \max\left(\frac{\beta}{\alpha^2(\alpha+\beta)}, \sqrt{\frac{\beta}{\alpha^2\varepsilon(\alpha+\beta)}}, \frac{\log\frac{3}{2\beta\delta}}{\beta(\alpha+\beta)}, \sqrt{\frac{\log\frac{3}{2\beta\delta}}{\beta}}, \frac{\sqrt{\delta}\log\frac{3}{2\beta\delta}}{\alpha^2\beta^2\varepsilon} \right).$$

Proof. Consider a tagged job arriving to the system. Recall from (6.1) that

$$\mathsf{E}[T_{\mathrm{CARD}}] - 2\mathsf{E}[S] \le 2(p_s + p_m)\mathsf{E}[W_s] + 2p_m \mathsf{E}[W_\ell \mathcal{I}(W_s < c)] + 2p_\ell \mathsf{E}[W_\ell].$$

We now bound the work expectations and probabilities in the last line.

- Lemma 6.4 implies $\mathsf{E}[W_s] \le c + q_A \mathsf{E}[W_s - c \mid W_s < c] \le c + \frac{q_A m_+}{\alpha}$.
- Lemmas 6.6 and 6.7 imply, after some simplification,

$$\mathsf{E}[W_\ell \mathcal{I}(W_s < c)] \le q_A \mathsf{E}[W_\ell] + \frac{q_A m_+}{\alpha^2} + \frac{4 q_A q_B c}{\beta} + \frac{\sqrt{2 q_A q_B m_+ c}}{\alpha\sqrt{\beta}}.$$

- Lemma 6.9 bounds $\mathsf{E}[W_\ell]$.

From these bounds and some simplification, using facts like $p_s + p_m + p_\ell = 1$ and $\alpha \le 1$, we obtain

$$\mathsf{E}[T_{\mathrm{CARD}}] - 2\mathsf{E}[S] \le 2(p_\ell + q_A)\left(1 + \frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + \frac{4 q_A m_+}{\alpha^2} + \frac{m_+\sqrt{q_A}}{\alpha\sqrt{\varepsilon}} + 6c + \frac{8 q_A q_B c}{\beta} + \frac{2\sqrt{2 q_A q_B m_+ c}}{\alpha\sqrt{\beta}} + \frac{8c\sqrt{\delta}}{\alpha^2\beta\varepsilon}.$$

We now use Lemma 6.10 to express as much as possible on the right-hand side in terms of α, β, δ, and ε. After some simplification, including using the preconditions of the theorem, we obtain

$$\mathsf{E}[T_{\mathrm{CARD}}] - 2\mathsf{E}[S] \le 2\left(p_\ell + \frac{2\beta}{\alpha+\beta}\right)\left(1 + \frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + \frac{8 m_+ \beta}{\alpha^2(\alpha+\beta)} + \frac{m_+\sqrt{2\beta}}{\alpha\sqrt{\varepsilon(\alpha+\beta)}} + \left(\frac{12 m_+}{\beta} + \frac{32 m_+ \alpha}{\beta(\alpha+\beta)^2}\right)\log\frac{3}{2\beta\delta} + \frac{4 m_+}{\alpha+\beta}\sqrt{\frac{2}{\alpha\beta}\log\frac{3}{2\beta\delta}} + \frac{16 m_+\sqrt{\delta}}{\alpha^2\beta^2\varepsilon}\log\frac{3}{2\beta\delta}.$$

Finally, we observe that $p_\ell = \mathsf{P}[S < m_+] \le \mathsf{P}[S < m]$ and simplify further.

B.4 Extension to Any Number of Servers

Theorem 3.3. *For any fixed number of servers $n > 2$, if CARD's parameters are set such that*

$$\alpha = \Theta(1), \qquad \beta = \Theta\left(\varepsilon^{1/3}\left(\log\frac{1}{\varepsilon}\right)^{2/3}\right), \qquad \text{and} \qquad c = \frac{n(n-1)m_+}{\beta}\log\frac{n+1}{n\beta\delta},$$

in the $\varepsilon \downarrow 0$ limit, then CARD achieves mean response time bounded by

$$\mathsf{E}[T_{\text{CARD}}] \le K_{\text{CARD}}\mathsf{E}[W_{\text{M/G/1}}] + O\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)^{1/3}.$$

In particular, CARD is heavy-traffic optimal: $\limsup_{\varepsilon\downarrow 0} \frac{\mathsf{E}[T_{\text{CARD}}]}{\mathsf{E}[T_\pi]} \le 1$ *for any dispatching policy* π.

Proof for $n > 2$ servers. Fix a short server s_i. Notice that under multi-server CARD, W_{s_i} are i.i.d. Thus, the analysis applies to any short server. Let A and B be the above and below periods of W_{s_i}. Using a similar proof as that of Lemma 6.6, we obtain

$$\mathsf{E}[W_\ell] - \mathsf{E}_c^0[W_\ell(0)] \le \left(\sqrt{q_A\mathsf{E}[A_e]} + \sqrt{q_B\mathsf{E}[B_e]}\right)^2 \le \frac{q_A m_+}{2\alpha^2} + \frac{4q_B c}{\beta}.$$

For any short server i, we have

$$q_A = \mathsf{P}[W_{s_i} < c] \le \frac{\beta + \frac{1}{n}\delta}{\alpha+\beta} \le \frac{2\beta}{\alpha+\beta} \qquad \text{and} \qquad q_B = \mathsf{P}[W_{s_i} \le c] \le \frac{\alpha}{\alpha+\beta}.$$

The proof is similar to that of Lemma 6.10. Note that q_A and q_B are the same for all short servers because W_{s_i} are i.i.d. in steady state. Lemmas 6.3 and 6.5 follow from the same arguments as the two-server case. We would like to obtain a counterpart of Lemma 6.9. To this end, we use a multi-server version of Theorem 6.1. Note that we have

$$IW_{\text{all}} = \frac{1}{n}\sum_{i=1}^{n-1} \mathcal{I}(W_{s_i} = 0)W_\ell + \frac{1}{n}\sum_{i=1}^{n-1} \mathcal{I}(W_\ell = 0)W_{s_i} + \frac{1}{n}\sum_{k<j} \mathcal{I}(W_{s_k} = 0)W_{s_j}$$

We bound these three terms separately.

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n-1} \mathsf{E}[\mathcal{I}(W_{s_i} = 0)W_\ell] &\overset{(a)}{=} \frac{n-1}{n}\mathsf{E}[W_\ell \mathcal{I}(W_{s_1} = 0)] \\
&\le \frac{n-1}{n}\left(\delta\,\mathsf{E}[W_{\text{all}}] + \frac{q_A m_+}{4\alpha^2} + \frac{4q_B c}{\beta} + \frac{2c\sqrt{2\delta}}{\beta}\right), \\
\frac{1}{n}\sum_{i=1}^{n-1} \mathsf{E}[\mathcal{I}(W_\ell = 0)W_{s_i}] &\overset{(b)}{=} \frac{n-1}{n}\mathsf{E}[\mathcal{I}(W_\ell = 0)W_{s_1}] \le \frac{n-1}{n}\left(n\varepsilon c + \frac{m_+\sqrt{q_A n\varepsilon}}{2\sqrt{2}\alpha}\right), \\
\frac{1}{n}\sum_{k<j} \mathsf{E}[\mathcal{I}(W_{s_k} = 0)W_{s_j}] &\overset{(c)}{\le} \frac{(n-1)(n-2)}{n}\mathsf{P}[W_{s_k} = 0]\mathsf{E}[W_j] \le (n-1)\left(c + \frac{m_+ q_A}{\alpha}\right)\delta,
\end{aligned}$$

where (a), (b), and (c) all follow from the fact that $W_{s_1}, \ldots, W_{s_{n-1}}$ are i.i.d. in steady state. Proof of the other bounds are similar to their counterparts in Lemmas 6.8 and 6.9. Theorem 6.1 gives

$$\begin{aligned}
\mathsf{E}[W_\ell] \le \mathsf{E}[W_{\text{all}}] &\le \left(1 + (n-1)\frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\text{M/G/1}}] \\
&\quad + n(n-1)c + \sqrt{n}(n-1)\frac{m_+\sqrt{q_A}}{2\sqrt{2}\alpha\sqrt{\varepsilon}} + \frac{8}{n}\frac{c\sqrt{\delta}}{\alpha^2\beta\varepsilon} + \frac{n(n-1)}{\varepsilon}\left(c + \frac{m_+ q_A}{\alpha}\right)\delta.
\end{aligned}$$

from Lemma 6.10. Since $W_{s_1}, \ldots, W_{s_{n-1}}$ are i.i.d. in steady state, we have, by PASTA,

$$\mathsf{P}[\text{A medium job joins a short server queue}] = \mathsf{P}[W_{s_1} \le c] = q_B$$

Therefore, using an argument similar to that for Theorem 6.11, we have

$$\begin{aligned}\mathsf{E}[T_{\mathrm{CARD}}] - n\mathsf{E}[S] \;\leq\; & n(p_\ell + q_A)\left(1 + (n-1)\frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + \frac{2nq_Am_+}{\alpha^2} \\ & + n(n-1)\sqrt{n}\frac{m_+\sqrt{q_A}}{2\sqrt{2}\alpha\sqrt{\varepsilon}} + \left(n^2(n-1) + n\right)\left(c + \frac{4nq_Aq_Bc}{\beta} + \frac{n\sqrt{2q_Aq_Bm_+c}}{\alpha\sqrt{\beta}}\right) \\ & + \frac{8c\sqrt{\delta}}{\alpha^2\beta\varepsilon} + n^2(n-1)\left(c + \frac{m_+q_A}{\alpha}\right)\frac{\delta}{\varepsilon}.\end{aligned}$$

This can be further expanded using the bounds for q_A and q_B, as well as the expression of c.

$$\begin{aligned}\mathsf{E}[T_{\mathrm{CARD}}] - n\mathsf{E}[S] \;\leq\; & n\left(p_\ell + \frac{2\beta}{\beta+\alpha}\right)\left(1 + (n-1)\frac{\delta}{\varepsilon}\right)\mathsf{E}[W_{\mathrm{M/G/1}}] + \frac{4n\beta m_+}{\alpha^2(\alpha+\beta)} + n(n-1)\frac{m_+\sqrt{n\beta}}{2\alpha\sqrt{\varepsilon(\alpha+\beta)}} \\ & + \frac{n(n-1)(n^2(n-1)+n)m_+}{\beta} + \frac{8n^2(n-1)m_+\alpha}{\beta(\alpha+\beta)^2}\log\frac{n+1}{n\beta\delta} \\ & + \frac{2n\sqrt{n(n-1)m_+}}{\alpha+\beta}\sqrt{\frac{1}{\alpha\beta}\log\frac{n+1}{n\beta\delta}} + \frac{8n(n-1)m_+\sqrt{\delta}}{\alpha^2\beta^2\varepsilon}\log\frac{n+1}{n\beta\delta} \\ & + n^2(n-1)\underbrace{\left(\frac{n(n-1)m_+}{\beta}\log\frac{n+1}{n\beta\delta} + \frac{2m_+\beta}{\alpha(\alpha+\beta)}\right)\frac{\delta}{\varepsilon}}_{\mathcal{T}}\end{aligned}$$

At this point, we note that the upper bound for $\mathsf{E}[T_{\mathrm{CARD}}] - n\mathsf{E}[S]$ is, after letting $n = 2$, the same as that in Theorem 6.11, except for $\mathcal{T}$. Thus, setting

$$\alpha = \Theta(1), \qquad \beta = \Theta\left(\varepsilon^{1/3}\log\frac{1}{\varepsilon}\right)^{2/3}, \qquad \text{and} \qquad c = \frac{n(n-1)m_+}{\beta}\log\frac{n+1}{n\beta\delta},$$

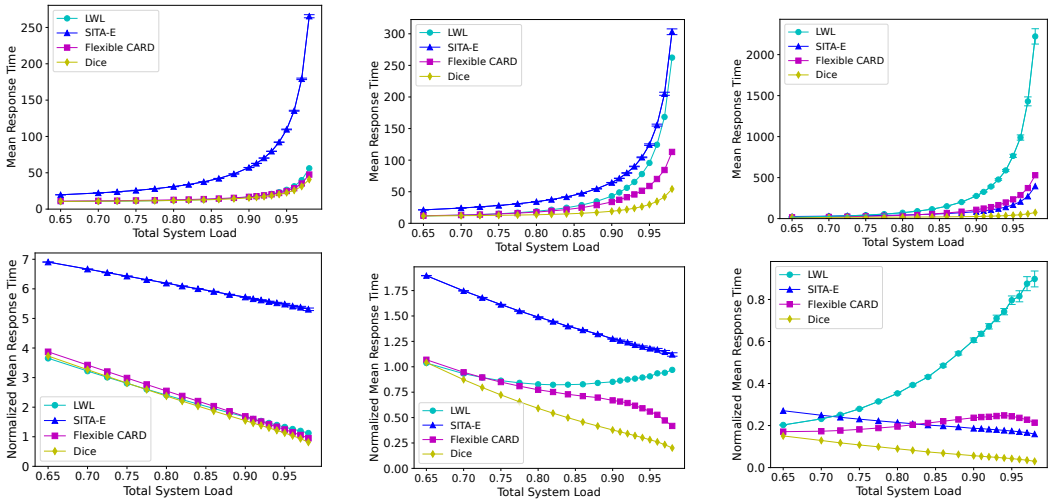
and noting that $\mathcal{T} \to 0$ as $\varepsilon \downarrow 0$, we conclude that the bound yields the same heavy-traffic scaling as that in Theorem 6.11. Finally, we note that K_{CARD} emerges because

$$\lim_{\varepsilon\downarrow 0} np_\ell = n\mathsf{P}[S \;i\; m] = K_{\mathrm{CARD}}.$$

C ADDITIONAL SIMULATIONS

Our additional simulations applies flexible CARD with three parameters to $n = 10$ servers. We simulate 40 trials for each data point, with 10^7 arrivals per trial for cv = 1 and cv = 10 and 3×10^7 job arrivals per trial for cv = 100. We show 95% confidence intervals when wider than the marker size.

Figure C.1 show that, for $n = 10$ servers, flexible CARD has decent performance when the coefficient of variation is small. However, for large coefficients of variation, flexible CARD does not perform well, even if we use LWL to dispatch small and medium jobs among the short servers. Specifically, when cv=10, flexible CARD deviates from Dice, although still better than LWL and SITA-E. When cv=100, flexible CARD performs worse than SITA-E at high loads. The unsatisfactory performance of flexible CARD for $n = 10$ servers motivates us to design multi-band CARD.



(a) 10 servers, cv = 1 (b) 10 servers, cv = 10 (c) 10 servers, cv = 100

Fig. C.1. Plots for the mean response times under the aforementioned policies for $n = 10$ servers. On the top row are plots of mean response times of the policies. On the bottom row are plots for mean response times normalized by the mean response time of a resource-pooled M/G/1 queue. We use LWL, instead of random, dispatching to short servers when a small or medium job arrives.

Received October 2023; revised January 2024; accepted January 2024