# Heavy-Traffic Optimal Size- and State-Aware Dispatching

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
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

## ABSTRACT

We study the problem of dispatching jobs to multiple FCFS (First-Come, First-Served) queues. 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. In this work, we propose the first size- and state-aware dispatching policy, called *CARD (Controlled Asymmetry Reduces Delay)*, that provably minimizes mean delay in heavy traffic. This abstract summarizes our full paper [13].

**ACM Reference Format:**
Runhan Xie, Isaac Grosof, and Ziv Scully. 2024. Heavy-Traffic Optimal Size- and State-Aware Dispatching. In *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE Abstracts '24), June 10–14, 2024, Venice, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3652963.3655059

## 1 PROBLEM: DISPATCHING TO FCFS QUEUES

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 this, we ask: *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 work with M/G arrivals, a typical stochastic arrival model.

Despite the extensive literature on dispatching in queueing theory [1–3, 6, 8, 11, 12, 14, 15], optimal size- and state-aware dispatching is an open problem, as highlighted by Hyytiä et al. [7]. The problem is a Markov decision process (MDP), so it can in principle be approximately solved numerically [10]. 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.

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

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

Our bounds match in the heavy-traffic limit as load $\rho$ approaches 1, the maximum load capacity. Specifically, we find a constant $K$ such that the dominant term of both bounds is $\frac{K}{1-\rho}$. Characterizing $K$ (see (2.1)) is another contribution of our work.

## 2 OPTIMAL DISPATCHING VIA ASYMMETRY

Below, we describe the intuition behind two-server CARD, illustrated in Figure 2.1. See our paper [13] for the $n$-server version.

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 [4, 5]. Under SITA, due to Poisson splitting, the dispatching system reduces to two independent M/G/1 systems. SITA can sometimes perform very well, but it can sometimes be much worse than simple LWL (Least Work Left) dispatching [5].

CARD uses SITA's design as a starting point, but makes one significant improvement. Where in SITA's design is there an 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

---

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

---

*SIGMETRICS/PERFORMANCE Abstracts '24, June 10–14, 2024, Venice, Italy*

ACM ISBN 979-8-4007-0624-0/24/06
https://doi.org/10.1145/3652963.3655059

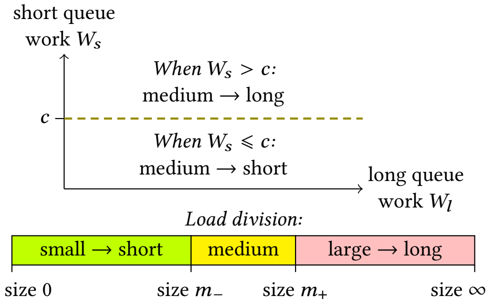

**Figure 2.1: Sketch of the CARD policy for two servers. Small and large jobs are always dispatched to the short or long server, respectively. Medium jobs are dispatched based on whether $W_s$, the amount of work at the short server, exceeds a threshold $c$. The size cutoffs $m_-$ and $m_+$ are chosen to be close to $m$ from (2.1) so that small and large jobs each constitute slightly less than half the load.**

a threshold which serves as a target amount of work at the short server. Whenever a medium job arrives, CARD 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.

We also study CARD in simulation across a wide range of loads, with Figure 2.2 showing one example. We find empirically that CARD has good performance outside of heavy traffic, but slightly modifying CARD can significantly improves performance. Both versions of CARD improve upon LWL and SITA, sometimes by an order of magnitude. The modified version is competitive with the Dice policy of Hyytiä and Righter [9], the best known heuristic for the size- and state-aware setting.

Our paper [13] presents three main theoretical results:

- A lower bound on the mean response time of any policy.
- An upper bound on CARD's mean response time which implies its heavy-traffic optimality.
- Stability of the system under CARD.

We summarize the first two results below. We consider a system with M/G arrivals with arrival rate $\lambda$, job size distribution $S$, and $n$ servers. We use the convention that each server completes work at speed $1/n$, so the load $\rho = \lambda \mathbb{E}[S]$ is the utilization.

Our lower and upper bounds imply that in the heavy-traffic limit, namely as $\rho \uparrow 1$, the mean response time $\mathbb{E}[T]$ of both the optimal policy and CARD scale as $\mathbb{E}[T] \sim \frac{K}{1-\rho}$, for the same constant $K$, and thus CARD is *heavy-traffic optimal* for mean response time. The constant $K$ is determined by solving the following for $m$:

$$K = \frac{\mathbb{E}[S^2]}{2\,\mathbb{E}[S \mid S \geqslant m]} = n\,\mathbb{P}[S \geqslant m] \cdot \frac{\mathbb{E}[S^2]}{2\,\mathbb{E}[S]}. \tag{2.1}$$

One can view $m$ as the value such that jobs of size $m$ and larger contribute a $1/n$ fraction of the load. As explained in our paper [13], the jobs CARD treats as "medium" are those of size close to $m$.

## ACKNOWLEDGMENTS

Isaac Grosof was supported by NSF grant no. CMMI-2307008 and a Tennenbaum Postdoctoral Fellowship at Georgia Tech. Ziv Scully was supported by NSF grant nos. CMMI-2307008, DMS-2023528, and DMS-2022448. He conducted this research in part while visiting the Simons Institute at UC Berkeley, and in part while an NSF FODSI postdoc at Harvard and MIT.

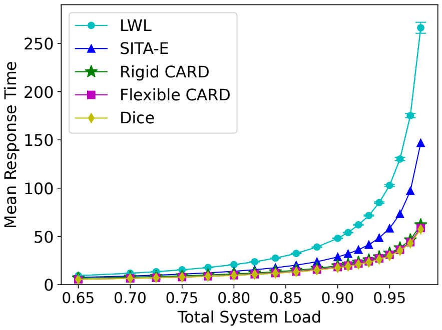

**Figure 2.2: Mean response time as a function of load for several policies, including two versions of CARD. *Rigid CARD* is the version we theoretically analyze, while *Flexible CARD* is modified slightly to improve empirical performance. The job size distribution has coefficient of variation $\mathrm{cv} = 10$. See Section 7 of our paper [13] for details.**

## REFERENCES

[1] Douglas G Down and Rong Wu. 2006. Multi-layered round robin routing for parallel servers. *Queueing Systems* 53 (2006), 177–188.
[2] 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.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] Esa Hyytiä, Peter Jacko, and Rhonda Righter. 2022. Routing with too much information? *Queueing Systems* 100, 3-4 (2022), 441–443.
[8] 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.
[9] Esa Hyytiä and Rhonda Righter. 2022. On Sequential Dispatching Policies. In *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC).* IEEE, 1–6.
[10] Esa Hyytiä and Rhonda Righter. 2023. On Dynamic Size-Aware Dispatching and Computation of the Optimal Actions. *SSRN* 4395052 (2023).
[11] 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.
[12] 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.
[13] Runhan Xie, Isaac Grosof, and Ziv Scully. 2024. Heavy-Traffic Optimal Size-and State-Aware Dispatching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8, 1 (2024), 1–36.
[14] Runhan Xie and Ziv Scully. 2023. Reducing heavy-traffic response time with asymmetric dispatching. *ACM SIGMETRICS Performance Evaluation Review* 51, 2 (2023), 36–38.
[15] 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.