Dispatching to Incentivize Fast Service in Multi-Server Queues

[Extended Abstract]

Sherwin Doroudi
Carnegie Mellon University
Tepper School of Business
sdoroudi@andrew.cmu.edu

Ragavendran Gopalakrishnan
California Institute of Tech.
ragad3@caltech.edu

Adam Wierman
California Institute of Tech.
adamw@caltech.edu

1. INTRODUCTION

As a field, queueing theory predominantly assumes that the arrival rate of jobs and the system parameters, e.g., service rates, are fixed exogenously, and then proceeds to design and analyze scheduling policies that provide efficient performance, e.g., small response time (sojourn time). However, in reality, the arrival rate and/or service rate may depend on the scheduling and, more generally, the performance of the system. For example, if arrivals are strategic then a decrease in the mean response time due to improved scheduling may result in an increase in the arrival rate.

Understanding the effect of such strategic interactions is the focus of "queueing games", which consider the interaction of classic queueing models and game theory. Typically, research on queueing games has focused on (i) strategic arrivals, which model jobs as strategic entities with utilities that depend on the performance received in the system, and (ii) profit-maximization, which allow the system to strategically price service (usually in the presence of strategic arrivals) in order to maximize profit. Examples of (i) include [11, 6, 8] and examples of (ii) include [14, 9, 12]. There is also an excellent survey available in [7].

In this work, we depart from the research cited above by considering a model where arrivals are not strategic, but where servers strategically choose their service rates. The motivating example for this work is call centers, where servers are people who have control over how quickly and efficiently they work. In this setting, a dispatch design that focuses only on efficiency may seek to send calls to the fastest and most efficient servers; however by doing so the dispatch policy is actually disincentivizing hard work by requiring the most effort from its best employees, which can hurt employee retention and job satisfaction. Resultantly, call center designs seek to ensure that call dispatching is "fair", in that servers have similar amounts of idle time [3, 4, 13]. This highlights the importance of paying attention to the "utility" of the servers; however such designs have not yet explicitly considered the strategic behavior of the servers.

In this work, we highlight that the strategic behavior of the servers has a fundamental impact on the design of dispatch policies. To do this, we focus on a simple model, an M/M/2 queue, where each server can strategically choose its service rate so as to maximize its utility, which is taken as the sum of a decreasing function of the chosen service rate and an increasing function of idle time experienced (which depends on the choice of service rate of the other server). In this setting, we consider the design of a non-preemptive, work-conserving dispatch policy, which decides which idle server to send the next job to, and our focus is on understanding the symmetric Nash equilibria (for the service rates) that emerges. Note that we focus on symmetric equilibria due to the importance of "fairness" in settings such as call centers.

In classic queueing theory, the most commonly proposed dispatch policies for this setting include Fastest Server First (FSF), Longest Idle Server First (LISF), and Random (which sends the job to each server with equal probability). When strategic servers are not considered, FSF is the natural choice for reducing the mean response time when forced to be work-conserving (though it is not optimal in general [5, 10]). However, we prove that FSF has no symmetric equilibria when servers are strategic. Further, we prove that LISF, a commonly suggested policy for call centers due to its fairness properties, has the same, unique, symmetric equilibrium as random dispatching. Thus, when strategic servers are considered, LISF does not even do better than the most naive dispatcher, Random. This highlights the importance of designing dispatch policies while being aware of the incentives they create.

With this in mind, one might suggest that Slowest Server First (SSF) would be a good dispatch policy, since it incentivizes servers to work fast; however, we prove that, like FSF, SSF has no symmetric equilibrium. But, by "softening" the bias placed by SSF toward slow servers we are able to give policies that are guaranteed to have a unique symmetric equilibrium and provide mean response times that are smaller than the response time at equilibrium under LISF and Random.

A key message provided by the results in this work is that dispatch policies must carefully balance two conflicting goals in the presence of strategic servers: they must make efficient use of the service capacity (e.g., by sending work to fast servers) while still incentivizing servers to work fast (e.g., by sending work to slow servers). While these two goals are inherently in conflict, it is possible to balance them in a way that provides improved performance over Random.

2. MODEL

Our motivating example throughout this abstract is call centers. Incoming jobs (calls) are served by one of many servers (agents). In the rest of this section, we describe the model in two parts—the queueing model and the game-theoretic model.

2.1 Queueing model

We assume that jobs arrive according to a Poisson process with rate normalized to 1, into a central, First Come First Served (FCFS) queue. The job sizes are independently exponentially distributed with rate normalized to 1. We assume that there is no abandonment, that is, every arrival is eventually served. In general, there are m servers that each choose the rates at which they work on the jobs, μ_i, $i = 1, \dots, m$, according to the game-theoretic model described in the next section. Let $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m)$ denote the vector of chosen service rates. In this abstract, we report results only for the case of $m = 2$. In other words, we have an $M/M/2/FCFS$ queue with an infinite buffer. The response time of a job is the amount of time it spends in the system, which is the sum of its waiting time (time spent in queue) and service time (time spent while being served). The performance objective for the system is to minimize the expected response time of a job, $\mathbb{E}[T]$.

The focus of this work is on the *dispatch policy*, which assigns jobs to servers. We restrict our attention to non-preemptive and work-conserving dispatch policies that do not use job size information. At any given instant, if at least one server is idle, and the queue is non-empty, such a policy will always pick the next job in queue and assign it to one of the idle servers. So, the defining aspect of the dispatch policy is how to make the choice of which idle server gets the next job in queue.

Four commonly studied dispatch policies are FSF (Fastest Server First), SSF (Slowest Server First), Random, and LISF (Longest Idle Server First). As their names indicate, FSF assigns the next job in the queue to the fastest idle server, SSF to the slowest idle server, Random to any of the idle servers with equal probability, and LISF to the idle server that has been idle the longest. Among these dispatch policies, FSF minimizes the expected response time. However, it was shown in [5, 10] that even when there are two servers, FSF is not an optimal dispatch policy – there are non-work-conserving policies that improve upon it. But, in [1], it was shown that as the number of servers grows large and the arrival rate approaches the service capacity in the Halfin-Whitt regime, FSF is asymptotically optimal. A drawback of FSF is that prioritizing faster servers does not distribute idle time evenly between the servers, and such 'unfairness' to the faster servers could lower employee satisfaction and degrade performance [3, 4]. As such, LISF is a "fairer" policy than FSF in the sense that asymptotically, it shares the idle time among the servers in proportion to their service rates [2]. It is this "fairness" property that leads LISF to be commonly used in practice.

In addition to these dispatch policies, we study two broad classes of dispatch policies: rate-based and idle-time-based.

2.1.1 Rate-based dispatch policies

Let $\mathcal{I}(t)$ denote the set of idle servers at time t. In a rate-based dispatch policy, jobs are assigned to idle servers only based on the rate vector $\boldsymbol{\mu}$, restricted to $\mathcal{I}(t)$. We consider a parameterized class of rate-based dispatch policies that we term *r-dispatch policies* ($r \in \mathbb{R}$). Under these policies, at time t, the next job in queue is assigned to idle server $i \in \mathcal{I}(t)$ with probability

$$p_i(\boldsymbol{\mu}, t; r) = \frac{\mu_i^r}{\sum\limits_{j \in \mathcal{I}(t)} \mu_j^r}$$

Notice that for special values of the parameter r, we recover well-known policies. For example, setting $r = 0$ results in Random; as $r \to \infty$, it approaches FSF; and as $r \to -\infty$, it approaches SSF.

2.1.2 Idle-time-based dispatch policies

Let $\boldsymbol{s}(t) = (s_1, \ldots, s_{|\mathcal{I}(t)|})$ denote the ordered vector of idle servers at time t, where server s_j became idle before server s_k whenever $j < k$. Let $\mathcal{P}_n = \Delta(\{1, \ldots, n\})$ denote the set of all probability distributions over the set $\{1, \ldots, n\}$. An idle-time-based dispatch policy is defined by a vector of probability distributions $\boldsymbol{p} = (p_1, \ldots, p_m)$, such that $p_j \in \mathcal{P}_j$, $j = 1, \ldots, m$. Under this policy, at time t, the next job in queue is assigned to idle server $s_j \in \boldsymbol{s}(t)$ with probability $p_{|\mathcal{I}(t)|}(j)$. Examples of idle-time-based dispatch policies include Random, LISF, and SISF (Shortest Idle Server First).

2.2 Game-theoretic model

The novelty of our model is its game-theoretic aspect—servers act as strategic players in a noncooperative game. Specifically, each server chooses its service rate, μ_i, from its action set, given by $[\underline{\mu}, \infty)$, where $\underline{\mu}$ is a minimum required service rate. This captures the fact that agents in a call center are expected to perform above a minimum level of efficiency, failing which they could be fired. We set $\underline{\mu} = \frac{1}{m}$, which is enough to ensure stability. The decisions made by the servers constitute their joint action profile, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$.[1]

Given a dispatch policy Π, servers aim to selfishly maximize their utility functions, given by

$$U_i(\boldsymbol{\mu}; \Pi) = I_i(\boldsymbol{\mu}; \Pi) - c(\mu_i),$$

where $I_i(\boldsymbol{\mu}, \Pi)$ is the steady state fraction of time that server i is idle, $c(\mu)$ is an increasing function representing the cost incurred by a server to work at rate μ. We assume that servers are homogeneous, so they all have the same cost function. Our utility function captures the fact that servers value idle time, but have fatigue.[2] Note that we assume a fixed payment model where servers are paid a fixed periodic salary, and therefore the wages would just add a constant term to the utility function. We normalize the cost function, so that $c(\underline{\mu}) = 0$. We assume that c is convex, and satisfies $c'(\underline{\mu}) < \frac{5}{6}$, and $c'''(\mu) \geq 0$. In particular, the assumption that $c'(\underline{\mu}) < \frac{5}{6}$ ensures voluntary participation.

Our choice of solution concept for this game is Nash equilibrium, which is a vector of service rates $\boldsymbol{\mu}^*$, such that for each server i, $U_i(\mu_i^*, \boldsymbol{\mu}_{-i}^*; \Pi) = \max_{\mu_i \geq \underline{\mu}} U_i(\mu_i, \boldsymbol{\mu}_{-i}^*; \Pi)$.[3] We restrict ourselves to *symmetric* Nash equilibria, since they best represent a "fair" outcome in our model with homogeneous servers. With a slight abuse of notation, for brevity, we say that μ^* is a symmetric Nash equilibrium if $(\mu^*, \ldots, \mu^*)$ is a Nash equilibrium.

3. RESULTS

Our interest in this work is on understanding how the choice of dispatch policy affects the system performance in the presence of strategic servers. We use the expected response time of a job, $\mathbb{E}[T]$, at symmetric equilibrium as the measure of system performance. The goal is to choose a dispatch policy that minimizes $\mathbb{E}[T]$ at symmetric equilibrium. To this end, we study the following questions:

- How do well known dispatch policies like FSF, SSF, Random, LISF perform in the presence of incentives?
- What dispatch policies admit a symmetric equilibrium? Do such policies admit unique symmetric equilibria?
- How do dispatch policies compare in terms of $\mathbb{E}[T]$ at symmetric equilibrium?

In the rest of this section, we state our answers to these questions for the case of two servers ($m = 2$), and the following assumptions on the cost function $c(\mu)$: (i) $c(\underline{\mu}) = 0$, (ii) $c'(\mu) > 0$, (iii) $c''(\mu) > 0$, (iv) $c'(\underline{\mu}) < \frac{5}{6}$, and (v) $c'''(\mu) \geq 0$. Note that we set $\underline{\mu} = \frac{1}{2}$.

[1] Even though the action profile $(\underline{\mu}, \ldots, \underline{\mu})$ is admissible, it can be shown that it is never an equilibrium, so there will be no stability issues.

[2] In general, any function that is increasing in $I_i(\boldsymbol{\mu}; \Pi)$ and decreasing in $c(\mu_i)$ could model this behavior, for example, $U_i(\boldsymbol{\mu}; \Pi) = -(1 - I_i(\boldsymbol{\mu}; \Pi))c(\mu_i)$.

[3] $\boldsymbol{\mu}_{-i}^* = (\mu_1, \ldots, \mu_{i-1}, \mu_{i+1}, \ldots, \mu_m)$ denotes the vector of service rates of all the servers except server i.

Our first result deals with the existence of symmetric equilibria, under rate-based r-dispatch policies. It asserts that, under mild assumptions on the cost function, there are policies that do not admit symmetric Nash equilibria.

Theorem 3.1. *There exists a bounded interval for r outside of which no r-dispatch policy admits a symmetric Nash equilibrium.*

Note that this result implies that well known policies FSF and SSF that have been used to construct asymptotically optimal and/or near-optimal dispatch policies in the classic setting [2] do not admit a symmetric equilibrium when incentives are considered. Intuitively, FSF does not admit a symmetric equilibrium because, given a symmetric action profile, decreasing its service rate slightly is a better response for either server, which would improve its idle time as well as cost. This dynamics would result in a progressive lowering of service rates, until there comes a point when the load becomes too high (and idle time too low), and so, working harder would significantly increase the idle time compared to the cost. A similar intuition holds for SSF as well. This highlights the importance of accounting for incentives while designing dispatch policies.

Our second result asserts that, there are some rate-based dispatch policies that admit unique symmetric equilibria.

Theorem 3.2. *Any r-dispatch policy with $r \in \{-2, -1, 0, 1\}$ admits a unique symmetric Nash equilibrium.*

Note that this result implies that Random admits a unique symmetric Nash equilibrium. This highlights the fact that in the presence of strategic servers, Random is "better" than FSF and SSF.

Our third result relates the class of idle-time-based dispatch policies to the class of rate-based dispatch policies.

Theorem 3.3. *All idle-time-based policies result in the same unique symmetric equilibrium as that of Random.*

Note that this result implies that no idle-time-based policy can perform better than Random (which is also a rate-based r-dispatch policy with $r = 0$). In particular, this highlights the fact that LISF does no better than Random.

Our final result provides a performance comparison among rate-based dispatch policies.

Theorem 3.4. *Any r-dispatch policy that admits a symmetric Nash equilibrium admits a unique symmetric Nash equilibrium. Further, among all such policies, $\mathbb{E}[T]$ at symmetric equilibrium is increasing in r.*

Note that this result (along with Theorem 3.2) implies that an r-dispatch policy with $r = -2$ outperforms Random and all idle-time-based policies. This highlights the fact that choosing a dispatch policy while paying attention to server incentives can lead to better performance.

Our results suggest that using smaller r values lead to better performance by "softening" the bias placed by SSF toward slow servers, but, beyond a certain limit, symmetric equilibria cease to exist. While $r = -2$ is the best performing parameter for which we could prove the existence of a symmetric equilibrium for all admissible cost functions, individual cost functions may allow for even smaller values. For example, Figure 1 shows the performance of a 2-server system, for two cost functions, $c_1(\mu) = \frac{\mu^2}{20} - \frac{1}{80}$, and $c_2(\mu) = \frac{2\mu^2}{3} - \frac{1}{6}$. In both cases, we see that moving r down to -10 still yields a symmetric equilibrium.

4. FINAL REMARKS

This abstract summarizes a first step toward understanding the interaction of strategic servers and dispatch policy design. We are working to extend the work in many directions. Most obviously, it would be interesting to extend our results to $m > 2$ servers. Other promising directions include exploring heterogeneous agents, asymmetric information, alternate payment models, alternate utility functions, more general queueing models, and broader classes of dispatch policies.

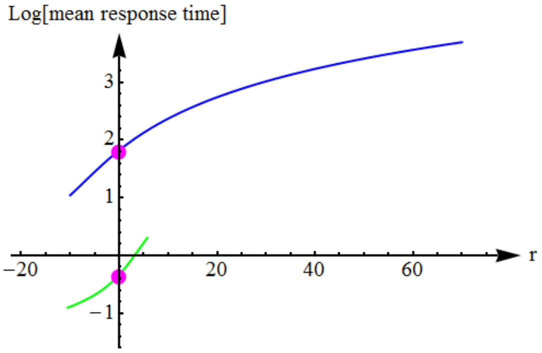

Figure 1: Mean response time (log scale) at symmetric equilibrium as a function of policy parameter r, for two cost functions c_1 (blue) and c_2 (green). The purple dots indicate the mean response time at symmetric equilibrium for Random and any idle-time-based policy.

5. REFERENCES

[1] M. Armony. Dynamic routing in large-scale service systems with heterogeneous servers. *Queueing Syst. Theory Appl.*, 51:287–329, 2005.

[2] M. Armony and A. R. Ward. Fair dynamic routing in large-scale heterogeneous-server systems. *Oper. Res.*, 58:624–637, 2010.

[3] Y. Cohen-Charash and P. E. Spector. The role of justice in organizations: A meta-analysis. *Organizational Behavior and Human Decision Processes*, 86(2):278–321, 2001.

[4] J. A. Colquitt, D. E. Conlon, M. J. Wesson, C. O. L. H. Porter, and K. Y. Ng. Justice at the millennium: A meta-analytic review of 25 years of organizational justice research. *Applied Psych.*, 86(3):425–445, 2001.

[5] F. de Véricourt and Y.-P. Zhou. Managing response time in a call-routing problem with service failure. *Oper. Res.*, 53:968–981, 2005.

[6] R. Hassin. Consumer information in markets with random product quality: The case of queues and balking. *Econometrica*, 54(5):1185–1195, 1986.

[7] R. Hassin and M. Haviv. *To Queue or Not to Queue: Equilibrium Behavior in Queueing Systems.* 2003.

[8] M. Haviv and T. Roughgarden. The price of anarchy in an exponential multi-server. *Oper. Res. Letters*, 35(4):421–426, 2007.

[9] N. C. Knudsen. Individual and social optimization in a multiserver queue with a general cost-benefit structure. *Econometrica*, 40(3):515–528, 1972.

[10] W. Lin and P. Kumar. Optimal control of a queueing system with two heterogeneous servers. *Automatic Control, IEEE Trans. on*, 29(8):696–703, 1984.

[11] P. Naor. The regulation of queue size by levying tolls. *Econometrica*, 37(1):15–24, 1969.

[12] R. C. Rue and M. Rosenshine. Some properties of optimal control policies for entry to an M/M/1 queue. *Naval Res. Logistics Quarterly*, 28(4):525–532, 1981.

[13] W. Whitt. The impact of increased employee retention on performance in a customer contact center. *Mfg. Service Oper. Mgmt.*, 8(3):235–252, 2006.

[14] U. Yechiali. On optimal balking rules and toll charges in the GI/M/1 queuing process. *Oper. Res.*, 19(2):349–370, 1971.