Randomized Load Balancing with General Service Time Distributions

Maury Bramson
School of Mathematics
University of Minnesota
bramson@math.umn.edu

Yi Lu
Microsoft Research
ylu@microsoft.com

Balaji Prabhakar
Departments of EE and CS
Stanford University
balaji@stanford.edu

ABSTRACT

Randomized load balancing greatly improves the sharing of resources in a number of applications while being simple to implement. One model that has been extensively used to study randomized load balancing schemes is the supermarket model. In this model, jobs arrive according to a rate-$n\lambda$ Poisson process at a bank of n rate-1 exponential server queues. A notable result, due to Vvedenskaya *et.al.* (1996), showed that when each arriving job is assigned to the shortest of $d \geq 2$ randomly chosen queues, the equilibrium queue sizes decay doubly exponentially in the limit as $n \to \infty$. This is a substantial improvement over the case $d = 1$, where queue sizes decay exponentially.

The method of analysis used in the above paper and in the subsequent literature applies to jobs with exponential service time distributions and does not easily generalize. It is desirable to study load balancing models with more general, especially heavy-tailed, service time distributions since such service times occur widely in practice.

This paper describes a modularized program for treating randomized load balancing problems with general service time distributions and service disciplines. The program relies on an *ansatz* which asserts that any finite set of queues in a randomized load balancing scheme becomes independent as $n \to \infty$. This allows one to derive queue size distributions and other performance measures of interest. We establish the *ansatz* when the service discipline is FIFO and the service time distribution has a decreasing hazard rate (this includes heavy-tailed service times). Assuming the *ansatz*, we also obtain the following results: (i) as $n \to \infty$, the process of job arrivals at any fixed queue tends to a Poisson process whose rate depends on the size of the queue, (ii) when the service discipline at each server is processor sharing or LIFO with preemptive resume, the distribution of the number of jobs is insensitive to the service distribution, and (iii) the tail behavior of the queue-size distribution in terms of the service distribution for the FIFO service discipline.

SIGMETRICS'10, June 14–18, 2010, New York, New York, USA.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Queueing Theory, Stochastic Processes

General Terms

Theory, Performance

Keywords

Load Balancing, Randomized Algorithms, Asymptotic Independence

1. INTRODUCTION

Load balancing is a canonical method for efficiently sharing resources among different jobs. It is employed in a number of scenarios: for example, in hash tables, in distributed memory machines (DMMs) emulating a single shared memory system, for path selection in networks, and for request assignment at web servers. Randomized load balancing, where a job is assigned to a server from a small subset of randomly chosen servers, is very simple to implement and delivers surprisingly good performance in terms of reducing collisions, waiting times, backlogs, etc.

Two versions of the load balancing problem have been studied in the literature: static and dynamic. The static version was first analyzed by Azar *et. al.* [1] using the balls-and-bins model. There are n bins into which n balls are dropped sequentially. Each ball is placed in the least loaded of d, $d \geq 1$, bins chosen uniformly at random with replacement[1], and ties are broken randomly. For $d = 1$, the maximum load is $(1 + o(1)) \ln n / \ln \ln n$ whereas, for $d \geq 2$, the maximum load is only $\ln \ln n / \ln d + O(1)$ [1]. Thus, there is an exponential improvement in the performance with just a small increase in implementation complexity. This model was studied by Mitzenmacher using mean field theory [13]. Vöcking [17] observed that breaking ties always to the "left", say when $d = 2$, is better than breaking ties at random. This observation has led to the wide-spread use of the d-left family of load balancing schemes in practical hash table implementations. We refer the reader to the survey paper [14] by Mitzenmacher *et.al.* for a detailed description of the results.

[1]Throughout this paper, we assume sampling is done with replacement, although the distinction between sampling with and without replacement vanishes for large n.

We are interested here in the dynamic supermarket model operating under the SQ(d) policy. Jobs arrive at a bank of n servers according to a rate-$n\lambda$ Poisson process, with $\lambda < 1$. The servers all employ the same service discipline (e.g., FIFO, PS or SRPT). The service times are IID with arbitrary distributions with mean 1. The policy assigns each arrival to the shortest of d queues chosen independently and uniformly at random. Here, by the shortest queue we mean the queue with the least number of jobs. Ties are broken randomly.

Vvedenskaya *et.al.* [18] analyzed the supermarket model under the SQ(d) policy for service times with an exponential distribution. They found that, for $d \geq 2$, as the number of queues n goes to infinity, the probability that the number of jobs in a typical queue is at least k is $\lambda^{\frac{d^k-1}{d-1}}$. This is an exponential improvement over the case $d = 1$, where the corresponding probability is λ^k. The model was also studied by Mitzenmacher [12], and its path-space evolution was studied by Graham [4]. Luczak and McDiarmid [9] showed that the length of the longest queue scales as $\ln \ln n / \ln d + \mathcal{O}(1)$.

Certain generalizations of the supermarket model have also been explored. Martin and Suhov [10] studied the supermarket mall model where each node in a Jackson network is replaced by N parallel servers, and a job joins the shortest of d randomly chosen queues at the node to which it is directed. Luczak and McDiamid [8] studied the maximum queue length of the original supermarket model (where service times are exponential) when the service speed scales linearly with the number of jobs in the queue.

Existing work on the analysis of the supermarket model follows a common methodology, which we summarize as follows: (i) View the evolution of the system as a Markov process. For instance, under the SQ(d) policy with Poisson arrivals and exponential services, the joint queue-size process is Markov; on the other hand, with general service distributions, the residual service times of jobs in the system are needed to obtain a Markovian description. (ii) Demonstrate that the Markov process is positive recurrent (*i.e.,* the system is *stable*) and hence has an equilibrium distribution. (iii) Obtain a description of the limiting system using differential (or partial differential) equations. (iv) Establish the existence and uniqueness of solutions to the differential equations. (v) Obtain the equilibrium distribution of the queue-size process by solving for the fixed point of the differential equations.

A limitation of the above approach for the SQ(d) policy is this: Although the policy only looks at the *number of jobs* in each queue, however, the Markov process description in the case of general service distributions requires the *residual service times* as well. This considerably complicates the model and the explicit computation of asymptotic queue occupancies. Our approach, outlined in Section 2, allows us to re-interpret the system as a simpler Markov process by using an asymptotic independence property.

Our contributions

1. The main contribution of the paper is a framework for analyzing randomized load balancing systems with general service times and service disciplines. A key component of the framework is an *ansatz* that postulates the following asymptotic independence property: the queue size processes in a randomized load balancing system become asymptotically independent as the number of queues n goes to infinity. This allows, through a fixed point computation, the explicit determination of performance measures of interest.

2. A central piece of our framework is the "queue at the cavity". This object describes the behavior of a single queue in the limit as the system size grows to infinity. The queue at the cavity is interesting from a theoretical perspective and as an efficient simulation tool. Using the queue at the cavity, we theoretically evaluate the performance of different load balancing policies and service disciplines; some specific findings are listed in the next two items. As a simulation tool, the queue at the cavity greatly reduces simulation time from more than 3 hours for simulating a 500-queue load balancing system to just 2 minutes.

3. In the case of the processor sharing (PS) or LIFO-PR service disciplines and general service time distributions, we use this framework to show that a certain asymptotic *insensitivity* holds, namely, as $n \to \infty$, the probability that a typical queue has at least k jobs *equals* $\lambda^{\frac{d^k-1}{d-1}}$.

4. For the FIFO service discipline and power-law service times, we discover threshold phenomena for P_k, the asymptotic probability that a typical queue has at least k jobs. Specifically, as $n \to \infty$, the rate of decay of P_k depends on the number of samples d: P_k decays either polynomially, exponentially, or doubly exponentially depending on d and the exponent of the power-law. We obtain the values of d at which these transitions take place.

5. An outline of the proof of the *ansatz* for the case where the queues are FIFO and the service times have a decreasing hazard rate (DHR).[2] This generalizes the earlier work for exponential service times. The family of random variables with decreasing hazard rate includes power-law distributions and, hence, is of great practical significance.

Organization of the paper. Section 2 describes our framework for analyzing randomized load balancing systems and the corresponding *ansatz.* Section 3 describes the queue at the cavity and the cavity map. Section 4 explores the consequences of the framework through a series of examples. Section 5 studies various routing policies through simulation and demonstrates the use of the queue at the cavity as an efficient simulation tool. Section 6 establishes the framework, *i.e.*, outlines a proof of the *ansatz*, for load balancing systems operating under the SQ(d) policy with DHR service distributions and the FIFO service discipline.

Remark 1. In this paper, we restrict our attention to the SQ(d) policies. Another family of policies consists of the LL(d) policies, which always assign the arriving job to the least loaded queue, *i.e.*, the queue with the least unfinished work. Analysis of the LL(d) policies is considerably easier than the SQ(d) policies. In particular, the analog of the *ansatz* will hold for all service distributions, irrespective of the service discipline. This can be shown using the same argument employed for the SQ(d) policies with the FIFO service discipline and DHR service times, and gives strong evidence in support of the *ansatz.*

[2]Recall that a random variable, $S \geq 0$, is said to have a decreasing hazard rate if the function $h(x) = \lim_{\delta \downarrow 0} P(S \in (x, x+\delta) | S > x)/\delta$ is decreasing in x.

2. A FRAMEWORK FOR ANALYSIS

We present a modularized program for analyzing load balancing systems after introducing some notation. Fix a service discipline and distribution and suppose that the system operates under the SQ(d) policy. Also, fix the value of d. Let $Q^n(t) = (q^{1,n}(t), ..., q^{n,n}(t))$ denote the joint queue-size process at time t. For $1 \leq i \leq n$, let $r^{i,n}(t)$ be the vector of *residual service times* of the $q^{i,n}(t)$ jobs at server i at time t and let $R^n(t) = (r^{1,n}(t), ..., r^{n,n}(t))$. Under the Poisson arrival and IID service assumptions, the process $(Q^n(t), R^n(t); t \geq 0)$ is Markov. Let $(\Pi^n(t), \Gamma^n(t))$ be the distribution of $(Q^n(t), R^n(t))$.

DEFINITION 1. *The service discipline at server i is said to be* local *if it only depends on $(q^{i,n}(\cdot), r^{i,n}(\cdot))$. Specifically, the service discipline at server i cannot depend on the queue sizes or residual service times of jobs at other servers. A load balancing system is* local *if the service discipline at each of its servers is local.*

Examples of local service disciplines include FIFO, PS, LIFO-PR, SRPT (shortest remaining processing time), LAS (least attained service) and SJF (shortest job first).

A modularized program

We now describe a program which will allow us to compute performance measures under the SQ(d) policy. The program consists of the following steps.

a. Uniform stability. Under the condition $\lambda < 1$, prove that $(Q^n(t), R^n(t); t \geq 0)$ is positive recurrent, and, hence, has a unique equilibrium distribution (Π^n, Γ^n). Show these equilibrium distributions are *uniformly stable*. That is, let

$$w^{1,n}(t) = \sum_{j=1}^{q^{1,n}(t)} r_j^{1,n}(t)$$

be the workload in queue 1 at time t. Define the norm:

$$\left|x^{1,n}(t)\right| = q^{1,n}(t) + w^{1,n}(t).$$

Then uniform stability means

$$\sup_{n \geq 1} P_{(\Pi^n, \Gamma^n)} \left(\left|x^{1,n}(t)\right| > M \right) \to 0 \text{ as } M \to \infty. \qquad (2.1)$$

Note that if the queue at each server has capacity $B < \infty$, then uniform stability is automatic.

Under the SQ(1) policy, the system decomposes into n stable, independent M/GI/1 queues. It is therefore tempting to infer that the system must be stable under the SQ(d) policies, for $d \geq 2$. However, establishing the stability of the SQ(d) policies is more difficult. It has recently been shown by Bramson [3] that uniform stability holds under the SQ(d) policies for all service disciplines.

In the following steps the service disciplines are assumed to be local.

b. Asymptotic independence. Demonstrate $(\Pi^n) \to (\Pi)$ as $n \to \infty$, where Π is a stationary and ergodic measure on Z_+^∞. Show that the limit Π is *unique*, depending only on the service distribution, service discipline and load balancing rule. Let $\Pi^{(k)}$ be the restriction of Π to its first k coordinates, with $\pi = \Pi^{(1)}$ being the one-dimensional marginal of Π. Show that, for every k,

$$\Pi^{(k)} = \bigotimes_{i=1}^{k} \pi.$$

That is, in the large n limit, any finite number of queues become *independent*.

Note that, for any initial state,

$$\pi_j \stackrel{def}{=} \lim_n \lim_t P(q^{1,n}(t) = j)$$

is the asymptotic equilibrium probability that queue 1 (and by symmetry, any queue) has j jobs. Here, π_j is also the asymptotic *fraction* of queues having j jobs.

c. Isolation of one queue, the queue at the cavity.[3] The above independence of the equilibrium distribution yields: In the large n limit, queue 1 (or any given queue) has state-dependent Poisson arrivals. More formally, let A_p^n be the process of *potential arrivals* to queue 1 in the n-system. These are arrivals that have queue 1 as one of their d choices. Note that, for each n, A_p^n is a Poisson process. Now, let A_a^n be the process of *actual arrivals* to queue 1, *i.e.*, those arrivals that join queue 1. Show that $A_a^n \to A_a$ in distribution as $n \to \infty$, where A_a is a state-dependent Poisson arrival process whose rate depends on the number of jobs in queue 1 prior to a potential arrival. Denote by λ_k the arrival rate of A_a when queue 1 has k jobs, and set $\Lambda = \{\lambda_k,\, k \geq 0\}$.

For a given service distribution and service discipline at queue 1, the previous statement allows one to obtain performance measures of interest, as described next.

d. Calculations. Given Λ, analyze queue 1 in the large n limit using queueing techniques to express π as a function of Λ,

$$\pi = F(\Lambda). \qquad (2.2)$$

The given load balancing policy routes jobs to queue 1 depending on the load of the rest of the infinite system, *i.e.*, queues 2, 3,.... Since π_j is the fraction of queues with j jobs in the rest of the system, one can also express Λ as a function of π,

$$\Lambda = G(\pi). \qquad (2.3)$$

Solve these two "fixed point" equations for π and Λ to obtain explicit distributions for the queue size.

The program we have outlined has a *structural* component (parts a–c) and a *computational* component (part d). The key steps that allow the explicit computation of the distribution of queue lengths are parts b and c. We formulate part b as an *ansatz*, which will be employed in Section 4.

An Ansatz

Consider a load balancing system operating under the SQ(d) policy, with $\lambda < 1$, and a given local service discipline. The jobs are assumed to have an arbitrary service time distribution with mean 1. Then part b of the modularized program is valid for this load-balancing system. That is, in the large

[3]This step and the next are reminiscent of the Cavity Method in Spin Glass Theory, a connection we elaborate in Section 3.

n limit, there is a unique equilibrium distribution. Moreover, under this distribution, any finite number of queues are independent.

3. THE QUEUE AT THE CAVITY

The Cavity Method of Spin Glass Theory (see [11] and [16]) is used for analyzing the behavior of n-particle systems in the limit as $n \to \infty$. At a high-level, the Cavity Method compares an n- and an $(n+1)$-particle system; the removal of a particle from the $(n+1)$-particle system results in the n-particle system, creating a "cavity." Equations are obtained for quantifying the effect of the particle at the cavity on the remaining n particles, *the environment*, and vice versa. By solving these equations one gets the behavior of the entire system in the large n limit.[4]

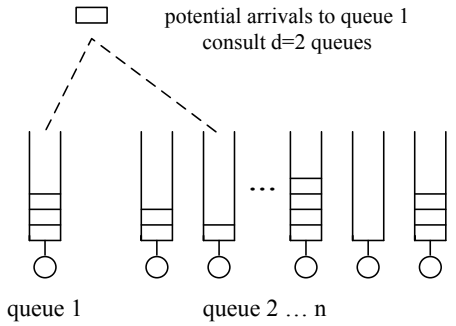

Figure 1: A bank of n servers with the SQ(2) routing policy. Potential arrivals to queue 1 are jobs that have queue 1 among the d choices.

Consider the n-queue system in Figure 1 and the dynamics at queue 1. As $n \to \infty$, this will be *the queue at the cavity.* For ease of exposition, we shall consider the SQ(2) policy and indicate the corresponding for the SQ(d) policy, $d > 2$. Recall that the potential arrival process at queue 1 consists of jobs that have queue 1 as one of their two choices. For any n, this process is Poisson with rate (recall sampling is with replacement)

$$\frac{2n-1}{n^2} n\lambda \to 2\lambda \text{ as } n \to \infty.$$

For SQ(d), the potential arrival process is Poisson with asymptotic rate $d\lambda$.

At any time t, let the empirical distribution of queues 2,...,n be denoted by $\mu^n(t) = \{\mu_k^n(t), k = 0, 1, ...\}$, where $\mu_k^n(t)$ is the fraction of queues 2,...,n having k jobs at time t. Thus, $\mu^n(t)$ is a random probability measure that reflects the distribution of the environment at time t. Let μ^n denote this measure when the n-queue system is in equilibrium. Note the following: for each finite n, (i) μ^n is a random probability measure and (ii) at any time, the size of queue 1 is dependent on μ^n at that time. However, by the *ansatz*, as $n \to \infty$, μ^n converges to a fixed (deterministic) limiting measure μ that is independent of the size of queue 1. We will refer to μ^n, respectively μ, as the *background distribution.*

Suppose an n-queue system is in equilibrium and that queue 1 has k jobs at some time t. Suppose also that there is a potential arrival to queue 1 at this time. This arrival samples a value, say j, from the background distribution μ^n. In the original SQ(2) system, this is equivalent to a potential arrival at queue 1 sampling a queue that has j jobs. Now, the potential arrival becomes an actual arrival (that is, joins queue 1) if $k < j$ or, if $k = j$, it becomes an actual arrival with probability 0.5. Since μ^n is not independent of the size of queue 1 for any finite n, the actual arrival process at queue 1 is *not* a Poisson process. However, as $n \to \infty$, the background distribution becomes independent of queue 1 and the actual arrival process is a Poisson process with a rate that depends only on the number of jobs in queue 1.

As $n \to \infty$, we refer to queue 1 as the queue at the cavity. See Figure 2. The queue at the cavity has the same service discipline and service time distribution as prescribed in the n-queue system. Potential arrivals occur at the queue as a rate-2λ Poisson process, with actual arrivals occurring according to a state-dependent Poisson process of rate λ_k when the size of the queue is k. For $M_k = \sum_{j \geq k} \mu_j$, one can check that

$$\lambda_k = \lambda \left(\frac{(M_k)^d - (M_{k+1})^d}{M_k - M_{k+1}} \right). \tag{3.1}$$

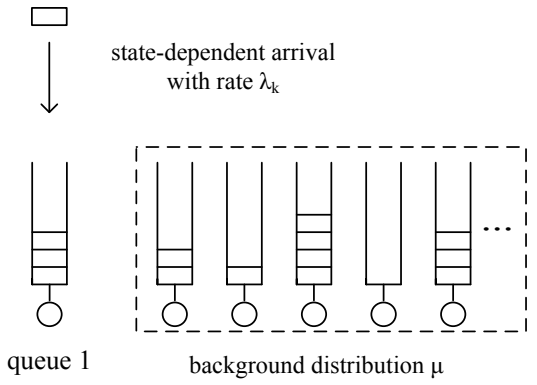

Figure 2: Asymptotic independence allows us to isolate queue 1 and treat the rest of the system as a background distribution μ. The actual arrival process to queue 1 is a state-dependent Poisson process with rate λ_k, where k is the number of jobs in the queue.

Now suppose that for a given background distribution ν that does not change over time, the queue at the cavity evolves according to the dynamics specified by ν. Assuming stability, let ν' denote the equilibrium distribution of the queue size. The map $\nu' = \mathcal{T}(\nu)$, of the set of probability measures on $\{0, 1, ...\}$ into itself, is called the *cavity map.*

When $\pi = \mathcal{T}(\pi)$, π is a fixed point of the cavity map. Such a π, if it exists and is unique, is the distribution of the size of queue 1 in the large n limit. It is also the asymptotic fraction of queues with a given number of jobs. For performing computations, we need to first obtain the maps in equations

[4]While the Cavity Method has proved to be a powerful tool in the analysis of a variety of complex systems (spin glasses, matching problems, coding systems, satisfiability problems, etc.), our use in the networking context does not need the full power of the method. However, the nature of problems in large complex networks suggests that there are likely to be several challenging applications for the Cavity Method in networking.

(2.2) and (2.3), since the map $\mathcal{T}$ is the composition of the maps F and G, that is, $\mathcal{T}(\pi) = F(G(\pi))$.

We outline the existence and uniqueness of the fixed points of $\mathcal{T}$ for FIFO queues with DHR service time distributions in Section 6. In the next section, we solve the cavity equations for some important scenarios.

4. COMPUTATION OF EQUILIBRIUM DISTRIBUTIONS

We now apply the *ansatz* and the Cavity Method to compute the equilibrium distributions for three scenarios.

1. Exponential service times

In [18], the stability of the SQ(d) policy with exponential service times was obtained through a coupling of the SQ(d) policy, $d \geq 2$, and the SQ(1) policy, with the asymptotic independence of the queue sizes being established in [4]. Here we rederive this result via the fixed point computation in part d of Section 2.

Consider queue 1 in the large n limit. Let $P_k = \sum_{j \geq k} \pi_j$ be the tail of the equilibrium queue-size distribution at queue 1 in the limit and note that it is also equal to the asymptotic fraction of queues with at least k jobs.

Since queue 1, with a state-dependent Poisson arrival process, is a simple birth-death chain, the flow balance equations give

$$\pi_{k+1} = \lambda_k \pi_k \;\Leftrightarrow P_{k+1} - P_{k+2} = \lambda_k (P_k - P_{k+1}). \quad (4.1)$$

At the fixed point, the background distribution is π; hence

$$\lambda_k = \lambda \left(\frac{(P_k)^d - (P_{k+1})^d}{P_k - P_{k+1}} \right). \quad (4.2)$$

Solving equations (4.1) and (4.2) yields

$$P_k = \lambda^{\frac{d^k - 1}{d-1}},$$

which is the result in [18].

2. General service times: Insensitivity

The insensitivity of a policy refers to its indifference to the distribution of job service times, *i.e.*, performance measures such as the queue size distribution depend on the service time distribution only through its mean. For example, an M/GI/1 queue operating under the PS or the LIFO-PR service discipline is well-known to be insensitive [6]. The insensitivity property is appealing since it allows the development of engineering rules without knowing precise traffic statistics.

Bonald and Proutière studied the insensitivity property of adaptive routing policies for load balancing flows on n parallel links ([2], 8.2). Flows of mean size $1/\mu$ arrive according to a rate-λ Poisson process. An arriving flow is routed to one of the links according to a routing polity. Each link has unit capacity, which it shares equally among all the flows passing through it. Link i can serve a maximum of N_i flows simultaneously. This system is equivalent to the supermarket model having queues with finite buffers and with the PS service discipline. Figure 3 shows link 1 with two flows and link n with 1 flow.

It was found in [2] that there exists a *unique* insensitive routing policy characterized as follows: Let $x = (x_1, \cdots, x_n)$,

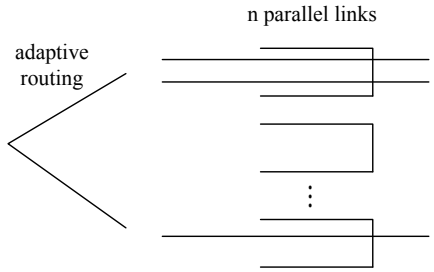

Figure 3: Adaptive routing on n parallel links with finite capacity.

where x_i is the number of flows on link i, and let $\nu_i(x)$ denote the actual arrival rate at link i when the network is in state x. The routing algorithm will be insensitive if and only if

$$\nu_i(x) = \frac{N_i - x_i}{\sum_{j=1}^{n} (N_j - x_j)} \lambda. \quad (4.3)$$

Therefore, the greedy routing policy, which directs an arriving flow to the link with the most "room," *i.e.,* the link $J = \arg\max\{N_j - x_j\}$, is *not* insensitive.

Note that under the greedy routing policy and the assumption $N_i = N$ for all i, the model for the routing problem is equivalent to the supermarket model operating under the Join the Shortest Queue load balancing policy and where the servers use the PS service discipline. This is because the queue with the most room will be the one with smallest number of jobs. By condition (4.3), the SQ(d) policy for this system is also *not insensitive* for any finite n.

However, the situation changes completely when $n \to \infty$. Suppose that flows arrive according to a rate-$n\lambda$ Poisson process at the system of n parallel links, each link with a capacity N. Suppose that flows are assigned to the links according to the SQ(d) policy. Assuming the *ansatz*, the backlogs at each of the links become independent and the actual arrival process at link 1 becomes a state-dependent Poisson process. Therefore, by the argument in [6] (pg. 72–80), the flow backlog process at link 1 is *reversible* and the distribution of the number of flows passing through it is *insensitive* to the service distribution. Specifically,

$$P_k = \frac{\lambda^{\frac{d^k - 1}{d-1}}}{\sum_{i=0}^{N} \lambda^{\frac{d^i - 1}{d-1}}}, \text{ for } k \leq N \text{ and } P_k = 0 \text{ for } k > N.$$

Thus, even though important properties such as reversibility and insensitivity do not hold for any finite n, they emerge in the limit as $n \to \infty$.

It should be noted that insensitivity holds also in a number of other cases, for example, with the LIFO-PR and other symmetric service disciplines. It also holds for the following randomized version of the policy due to Bonald and Proutière [2]:

BP(d). Consider the supermarket model where each queue has a buffer of capacity N. The service time distribution is arbitrary and the service discipline is PS. The BP(d) load balancing policy works as follows: For each arrival, d queues are sampled. Let $x = (x_1, \cdots, x_d)$ be the number of jobs in each of these queues. The flow is assigned to the i^{th} sampled

queue with probability

$$p_i = \frac{N - x_i}{\sum_{j=1}^{d}(N - x_j)},$$

i.e., the job is assigned to a queue with a probability proportional to the "room" available in the queue.
The performance of the BP(d) policy is hard to obtain via the standard approach. Indeed, the differential equations describing the proportion of queues with a given load are difficult to solve explicitly, especially for arbitrary service distributions. However, with our approach, the model is easy to analyze using the queue at the cavity and the insensitivity of the PS service discipline. The performance of the BP(2) algorithm is studied in Section 5.2.

3. General service times with the SQ(d) policy and the FIFO service discipline: Threshold phenomena and generalized Fibonacci sequences

In the previous examples we have seen that with exponential service times or with symmetric service disciplines, the queue size distribution decays super-exponentially for all $d \geq 2$; there is no real improvement to be obtained from sampling more than two queues: the "power of two choices" holds. The example here will show that the number of samples matters crucially when the service times are power-law and the service discipline is FIFO. In particular, we will obtain super-exponentially decaying queue sizes when d exceeds a threshold value. Theorem 1 specifies this value of d in terms of the tail of the service distribution.

Once again, we assume that the *ansatz* holds. Therefore, assuming the background distribution is π, the arrival process at queue 1 is a state-dependent Poisson process with rates given by equation (4.2). To obtain the reverse relationship between $\pi(\cdot)$ and $\lambda(\cdot)$, we need to solve the queueing equations corresponding to queue 1. That is, we need to determine the queue-size distribution of a FIFO queue with state-dependent Poisson arrival process with arrival rates $\{\lambda_k\}$ and power-law, IID service times. The result of this computation is stated below as Theorem 1.

DEFINITION 2. *Let $\lfloor\beta\rfloor$ be the largest integer not exceeding β and let $\hat{\beta} = \beta - \lfloor\beta\rfloor$. The* β-generalized Fibonacci sequence *is given by*

$$F_{\beta,k} = 1 \text{ for } 0 \leq k \leq \lfloor\beta\rfloor - 1, \text{ and}$$

$$F_{\beta,k} = \sum_{i=k-\lfloor\beta\rfloor+1}^{k-1} F_{\beta,i} + \hat{\beta} F_{\beta,k-\lfloor\beta\rfloor} \quad \text{for } k \geq \lfloor\beta\rfloor.$$

For a fixed integer $d \geq 2$, let

$$\alpha = \lim_{k\to\infty} \frac{\log_d\left((d-1)F_{\beta,k}\right)}{k}$$

be the growth rate *of the sequence.*

One can check that $0 < \alpha < \infty$. Note that for $\beta = 3$ and $d = 2$, we get the familiar Fibonacci sequence 1, 1, 2, 3, 5,..., with growth rate $\alpha = \log_2\left(\frac{1+\sqrt{5}}{2}\right)$.

In the next two theorems, we consider the SQ(d) load balancing policy for a fixed $d \geq 2$ and the FIFO service discipline. Theorem 1 considers power-law service times with asymptotic "shape parameter" β and gives a relationship between d and β, depending upon which, the equilibrium queue sizes decay doubly exponentially, exponentially, or just polynomially. Theorem 2 states that when service times have exponential moments, the queue sizes decay doubly exponentially.

THEOREM 1. *Let $\beta^* = \frac{d}{d-1}$ and suppose the service time S satisfies $P(S > x) = \Theta(x^{-\beta})$.*

(1) For $\beta > \beta^*$, $\log_d \log \frac{1}{P_k} = (\alpha + o(1))k$. That is, if the tail of the service times decays faster than β^*, then the queue size decays doubly exponentially.

(2) For $\beta = \beta^*$, $\log \frac{1}{P_k} = \Theta(k)$. That is, at the critical value β^*, the number of samples just suffices to ensure that the queue size decays exponentially.

(3) For $1 < \beta < \beta^*$, $P_k = k^{-\frac{\beta-1}{1-(d-1)(\beta-1)}+o(1)}$. That is, the number of samples only suffices for the queue size to decay polynomially.

THEOREM 2. *Suppose $E(e^{\theta S}) < \infty$ for some $\theta > 0$. Then, for all $d \geq 2$, $\log_d \log \frac{1}{P_k} = (1 + o(1))k$.*

Remark 2. Under the SQ(1) policy, queue 1 becomes an M/GI/1 queue and it is well-known that when $P(S > x) = \Theta(x^{-\beta})$, the queue-size has only a $(\beta - 1)$ moment (see [7], pg. 191-196). Thus, in case (3) of Theorem 1, the queue-size does not even have a finite first moment.

However, as Theorem 1 shows, the SQ(d) policy for $d \geq 2$ not only gives finite moments for all $\beta > 1$, but it can give exponential or even double exponential moments when d is chosen correctly. Theorem 2 shows that double exponential moments are the best one can get.

Proof outline of Theorem 1. We compute accurate upper and lower bounds on the equilibrium distribution of an isolated queue. Since there are three regions for β, this gives six cases in all to consider.

We start by rephrasing the problem in terms of the tail of the return time for the corresponding Markov process, whose transition probabilities are given by the (unknown) background distribution. Associated with this Markov process is the discrete time Markov chain given by the number of jobs after each succeeding departure from the queue; bounds for the tail of the return time for the chain can be reinterpreted in terms of the tail for the Markov process.

The arguments associated with the three regions of β are different. The arguments for the upper and lower bounds when $\beta > \beta^*$ are the most natural, and because of the very rapid decrease of the tail here, estimates can be crude. Elementary combinatoric arguments together with Borel-Cantelli estimates over events associated with the size of upward jumps for the Markov chain give both directions.

The arguments associated with the other two regions are more delicate. When $\beta < \beta^*$, one first obtains a weak bound on the power of the tail distribution, which one improves by successive iteration. When $\beta = \beta^*$, one employs the upper bounds obtained from the previous case, together with direct computation, again employing various Borel-Cantelli estimates.

3.1 A robust sampling algorithm

Theorem 1 shows that two samples may not suffice, in general, for obtaining a doubly exponential tail for the queue size. For this reason and the fact that service distributions

(or file sizes) may not be knowable or may change over time, we formulate an algorithm, called *d-adaptive*, which chooses the right number of samples in an adaptive fashion.

The d-adaptive algorithm: Let $f(k)$ be a positive integer-valued, non-decreasing function of k. For every arrival, do the following:
1. Choose a queue at random.
2. Suppose this queue has k jobs. Sample $f(k)$ additional queues so that the total number of samples equals $f(k)+1$.
3. Send the arrival to the shortest of these $f(k)+1$ queues, breaking ties at random.

Some choices for $f(\cdot)$ are $f(k)=k$ and $f(k)=k^2$. The correct choice involves a trade-off between the desired degree of load balancing and the complexity of sampling several queues. When $f(k)=k$, the arrival rate at queue 1 asymptotically equals

$$\lambda_k = \lambda\left(\frac{(P_k)^{k+1}-(P_{k+1})^{k+1}}{P_k-P_{k+1}}\right).$$

In Section 5, we will simulate the d-adaptive algorithm using $f(k)=k$.

5. SIMULATION

There are two parts to this section. In 5.1, we simulate a large system of n queues with various service disciplines and distributions, under the SQ(d) and the d-adaptive policies. We demonstrate the insensitivity property of the PS and LIFO-PR disciplines under the SQ(d) policy in 5.1.1. In 5.1.2, we study the tail behavior of queue sizes under the FIFO service discipline for different service distributions. We pay particular attention to the threshold phenomenon. In 5.1.3, we compare the performance of the d-adaptive algorithm to the SQ(d) algorithm when the queue is FIFO and the service distribution is heavy-tailed.

In 5.2, we simulate the queue at the cavity with a fixed background distribution and iterate the cavity map $\nu' = \Gamma(\nu)$ to obtain its fixed point, π. (Recall that π gives us the tail behavior of the queue sizes in the limit as $n\to\infty$.) Simulating the cavity map is a much faster method for studying the performance of the load balancing system for large n than actually simulating a large n system. Thus, it is an efficient simulation tool for studying various policies for large randomized load balancing systems. We demonstrate the convergence of the cavity map corresponding to the SQ(d) policy when the FIFO service discipline is used in 5.2.1. We investigate the insensitivity property of the BP(2) randomized routing policy, defined in Section 4, in 5.2.2.

Simulation parameters

Throughout this section, we use the following four service distributions. To be able to compare the results, we normalize by setting the traffic intensity $\rho=\lambda E(S)=0.6$ in all cases.
1. *Constant services.* $S=1$ with probability 1, $\lambda=0.6$.
2. *Exponential services.* $P(S>x)=e^{-x}$, $\lambda=0.6$.
3. *Power-law,* $\beta=3$. $P(S>x)=x^{-3}$, $x\geq 1$, $\lambda=0.4$.
3. *Power-law,* $\beta=1.5$. $P(S>x)=x^{-1.5}$, $x\geq 1$, $\lambda=0.2$.

5.1 Large System of Queues

In this section, we obtain the following empirical measure of the queue size distribution in a large randomized load balancing system:

$$P_k^n(t)=\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(q^{i,n}(t)\geq k),$$

where t is measured in the number of arrivals. By the PASTA property [7], this is equal to the empirical queue size distribution at an arbitrary time instant. We start with $q^{i,n}(0)=0$ for $1\leq i\leq n$ and a large value of t so that the system is close to equilibrium. We find that $T=10$ million suffices. We record

$$P_k^n \stackrel{def}{=} \frac{1}{T}\sum_{t=1}^{T}P_k^n(t).$$

5.1.1 Asymptotic Insensitivity of PS and LIFO-PR

We consider the SQ(2) load balancing system for $n=500$ and $\rho=0.6$. Table 1 shows the empirical distribution of the tail of the queue sizes with the PS and LIFO-PR service disciplines. The first column shows the putative limiting distribution $P_k=\rho^{2^k-1}$. The 2nd and 3rd columns give the empirical distribution of the tail of the queue sizes when the service times are constant, equal to 1. The 4th and 5th columns give the corresponding values when the service times are power-law with shape parameter $\beta=1.5$.

As can be seen from the table, the queue-size distribution under the PS and LIFO-PR service distributions is very close to the corresponding values for the exponential service time distribution, supporting the claimed insensitivity.

	ρ^{2^k-1}	constant		power law ($\beta=1.5$)	
		PS	LIFO-PR	PS	LIFO-PR
P_0	1	1	1	1	1
P_1	0.6	0.601	0.601	0.599	0.599
P_2	0.216	0.217	0.217	0.215	0.214
P_3	0.0280	0.00282	0.0282	0.0276	0.0276
P_4	0.0005	0.0005	0.0005	0.0005	0.0005
P_5	10^{-7}	10^{-7}	10^{-7}	10^{-7}	10^{-7}

Table 1: Asymptotic insensitivity of PS and LIFO-PR service disciplines. Service time distributions are constant or power law with $\beta=1.5$.

Next, we study the behavior of the system as n varies. Figure 4 plots the difference between the empirical distribution for a given value of n and the limiting distribution for the PS service discipline, that is, it plots

$$\sigma=\sum_{k=1}^{5}|P_k^n-\rho^{2^k-1}|$$

against n, with $\rho=0.6$. We use four service time distributions for this experiment: constant, exponential, power-law with $\beta=3$ and $\beta=1.5$. The figure shows that the empirical queue size distribution converges quickly to the limiting distribution as n increases. Note that the plot is in log-log scale. The plot for the LIFO-PR service discipline is similar.

5.1.2 FIFO under SQ(2)

We simulate four service time distributions with FIFO under SQ(2): constant, exponential, power law with $\beta=3$ and $\beta=1.5$. The constant distribution is an example of

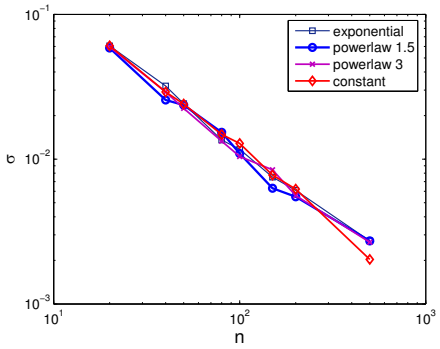

Figure 4: Convergence of equilibrium distributions for the PS discipline. The y-axis is the distance between the empirical distribution and the computed limiting distribution, which is the same for all four service distributions.

distributions with an increasing hazard rate, and the power law distributions are examples of distributions with an decreasing hazard rate. The exponential distribution has a constant hazard rate. We tabulate the results for $n = 500$ and $\rho = 0.6$.

	constant	exponential	power law ($\beta = 3$)	power law ($\beta = 1.5$)
P_0	1	1	1	1
P_1	0.601	0.601	0.601	0.600
P_2	0.121	0.217	0.144	0.270
P_3	0.0032	0.028	0.0084	0.130
P_4	1.5×10^{-6}	0.0005	7.5×10^{-5}	0.0762
P_5	0	10^{-7}	0	0.0517
P_6	0	0	0	0.0383
P_7	0	0	0	0.0299
P_8	0	0	0	0.0244
P_9	0	0	0	0.0204

Table 2: Empirical distribution for $n = 500$ and FIFO service discipline under SQ(2), with four different service time distributions.

The queue size distribution for the constant service times has the fastest decreasing tail among the four. The queue size for exponential service times has a distribution close to $P_k = \rho^{2^k-1}$. From Theorem 1, the threshold for the two power law distributions are: $d = 3$ for $\beta = 1.5$, and $d = 1.5$ for $\beta = 3$. Thus, under the SQ(2) policy, the power law distribution with $\beta = 3$ is expected to have a doubly-exponential tail and that with $\beta = 1.5$ is expected to have a polynomial tail. The numbers in Table 2 support the threshold phenomenon: the queue size distribution for $\beta = 1.5$ decays much more slowly than for $\beta = 3$.

5.1.3 FIFO and the d-adaptive Algorithm

We simulate the d-adaptive algorithm using $f(k) = k$ when the service times have a power law with $\beta = 1.5$ and compare the result to that of SQ(d), $d = 2, 3, 4$. From Theorem 1, the tail of the queue size decays polynomially, exponentially or doubly exponentially as $d = 2$, 3 or 4, respectively. The experiment uses $n = 500$ and $\rho = 0.6$. Table 3 gives a comparison of the performance.

	SQ(2)	SQ(3)	SQ(4)	d-adaptive
P_0	1	1	1	1
P_1	0.601	0.602	0.602	0.601
P_2	0.270	0.202	0.155	0.238
P_3	0.130	0.057	0.019	0.061
P_4	0.0762	0.017	0.001	0.0044
P_5	0.0517	0.006	0	0
P_6	0.0383	0.002	0	0
P_7	0.0299	0.0007	0	0
P_8	0.0244	0.0002	0	0
P_9	0.0204	10^{-6}	0	0

Table 3: Comparison of the d-adaptive and SQ(d) algorithm, for $d = 2, 3, 4$, under FIFO. The service times are power law with $\beta = 1.5$.

First of all, this experiment shows the threshold phenomenon in sharp relief: the tail of the queue sizes decays very slowly when $d = 2$ when compared with $d = 3$. In turn, the decay is much slower for $d = 3$ when compared with $d = 4$.

Secondly, the tail of the queue sizes under the d-adaptive algorithm decays much faster than under the SQ(2) and SQ(3) policies, and is very comparable to the decay under the SQ(4) policy. On the other hand, the d-adaptive policy obtains much fewer samples than the SQ(4) policy; indeed, it obtains a total of 3 samples or fewer for 94% of the arrivals.

5.2 Simulating the Queue at the Cavity

In this section, we study the queue at the cavity. There are two ways to proceed:

(i) Obtain the cavity map $\mathcal{T}(\cdot)$ for the given load balancing policy, service discipline and service distribution. Take an initial background distribution ν_0 and obtain the iterates $\nu_K = \mathcal{T}^K(\nu_0)$.[5] Hence obtain the fixed point $\pi = \mathcal{T}(\pi)$.

(ii) Simulate the queue at the cavity. That is, take an empty single server queue with a rate-2λ Poisson process according to which potential arrivals occur and a given background distribution ν_0. Use the load balancing policy, the service discipline and service distribution to simulate the queue as described in Section 3 until it is close to equilibrium (assuming equilibrium exists). Obtain the distribution of the resulting queue size and denote it ν_1. Repeat this procedure to obtain ν_K and π.

We use both methods, taking $\nu_0 = (1, 0, 0, ...)$. Method (ii) will be easier when the queueing equations are hard to write down explicitly, making it difficult to obtain the map $\mathcal{T}$.

5.2.1 FIFO under SQ(2)

We use method (ii) to iterate the cavity map corresponding to a queue which uses FIFO service discipline and the SQ(2) policy. Table 4 shows the evolution of

$$P_{K,k} = \sum_{j \geq k} \nu_{K,k},$$

[5]The iterates can be obtained numerically if it is difficult to obtain them analytically.

that is, the tail of the queue size distribution ν_K. For simplicity, we shorten $P_{K,k}$ to P_K and tabulate the results for some selected iterations K, as indicated in the table. The service distribution is exponential with mean 1. Table 5 shows the corresponding results when the service distribution is power law with $\beta = 3$.

As mentioned earlier, the queue at the cavity is a much more efficient way of simulating large load balancing systems. For example, both the above-mentioned experiments finish within 2 minutes, which is approximately 100 times faster than the time it takes to simulate a load balancing system with 500 queues to a comparable precision.

	Iter 1	Iter 2	Iter 5	Iter 10	ρ^{2^k-1}
P_0	1	1	1	1	1
P_1	0.374	0.502	0.588	0.600	0.6
P_2	0	0.092	0.199	0.215	0.216
P_3	0	0	0.021	0.0277	0.0280
P_4	0	0	0.0002	0.0005	0.0005
P_5	0	0	0	10^{-7}	10^{-7}
P_6	0	0	0	0	0

Table 4: Iteration of the cavity map for FIFO under SQ(2). The service time distribution is exponential. The queue size distribution after 1, 2, 5 and 10 iterations are shown. The total variation distance between the background and the queue size distributions at 10^{th} iteration is 0.0005.

	Iter 1	Iter 2	Iter 5	Iter 10	Table 2
P_0	1	1	1	1	1
P_1	0.375	0.507	0.591	0.600	0.601
P_2	0	0.064	0.139	0.143	0.144
P_3	0	0	0.0074	0.0082	0.0084
P_4	0	0	0.00005	6×10^{-5}	7.5×10^{-5}
P_5	0	0	0	0	0

Table 5: Iteration of the cavity map for FIFO under SQ(2). The service time distribution is power law with $\beta = 3$. The queue size distribution after 1, 2, 5 and 10 iterations are shown. The total variation distance between the background and the queue size distributions at 10^{th} iteration is 0.00054.

Table 4 traces the evolution of the iterates of ν_K for $K = 1, 2, 5$ and 10 when the service times have an exponential distribution and compares this with the theoretical limiting distribution $P_k = \rho^{2^k-1}$, which is tabulated in the last column. As can be seen, the convergence is pretty rapid: ν_{10} is very close to the right answer.

Table 5 compares the iterates of ν_K when the services have a power law distribution with $\beta = 3$. This is compared with the distribution of the tail of the queue size obtained from simulating a system of 500 queues, which was presented in Table 2. We again observe a close match.

5.2.2 BP(2): A randomized routing policy

We consider the BP(2) randomized routing policy described in Example 2 of Section 4. Recall that the supermarket model under consideration has a finite buffer size N. The service time distribution is arbitrary and the service discipline is PS. For each arrival, 2 queues are sampled. Let (x_1, x_2) be the number of jobs in each queue. The flow is assigned to the i^{th} sampled queue, $i = 1, 2$, with probability

$$p_i = \frac{N - x_i}{\sum_{j=1}^{2}(N - x_j)}.$$

Note that, by the result in [2], the randomized version of the routing algorithm is not insensitive for any finite n. Nevertheless, due to our *ansatz*, it is insensitive in the limit as $n \to \infty$. We verify this via simulations.

First, we need to obtain the limiting distribution of the queue sizes. Since we have assumed the *ansatz* and since the PS service discipline is insensitive, we can obtain the limiting distribution of the queue sizes by analyzing the cavity map for the exponential service distribution.

	Iter 1	Iter 2	Iter 3	Iter 5	Exponential
P_0	1	1	1	1	1
P_1	0.561	0.594	0.599	0.600	0.600
P_2	0.297	0.335	0.341	0.342	0.342
P_3	0.147	0.178	0.182	0.183	0.183
P_4	0.067	0.087	0.0905	0.091	0.091
P_5	0.028	0.039	0.041	0.041	0.041
P_6	0.010	0.016	0.017	0.017	0.017
P_7	0.0033	0.0054	0.058	0.0058	0.0058
P_8	0.0008	0.0015	0.0016	0.0016	0.0017
P_9	0.0001	0.0003	0.0003	0.0003	0.0003

Table 6: Iteration of the cavity map for PS under the BP(2) routing policy. The service time distribution is power law with $\beta = 3$. The queue size distribution after 1, 2, 3 and 5 iterations is compared with the fixed point for the exponential service time distribution (last column). The closeness of the last two columns supports the insensitivity of the BP(2) routing policy.

Accordingly, let ν_0 be a probability measure on $\{0, 1, ..., N\}$ and let it be the background distribution to the queue at the cavity. Let $\nu_{0,j}$ be the probability of obtaining j when sampling from ν. According to the routing policy BP(2), the arrival rate at the queue at the cavity when it has k jobs is given by

$$\lambda_k = \sum_{j=0}^{N} \frac{N - k}{N - k + N - j} \nu_{0,j}. \tag{5.1}$$

When the services are exponential, the queue size process of the queue at the cavity is a birth-death chain. Its distribution ν_1 is given by

$$\lambda_k \nu_{1,k} = \nu_{1,k+1}, \text{ for } k = 0, \cdots, N-1. \tag{5.2}$$

One can solve equations (5.1) and (5.2) numerically to obtain ν_1. Repeating the procedure, one obtains ν_K for $K \geq 2$ and the fixed point π. The last column in Table 6 contains the values of the fixed point π for exponential services.

By simulating the queue at the cavity, one can also obtain the iterates ν_K when the service time distribution h is power law with $\beta = 3$. The iterates are shown in Table 6. As can be seen by comparing the last two columns, the queue size distributions are virtually identical for the exponential

service distribution and the power law distribution, strongly supporting the claimed insensitivity of the BP(2) policy in the limit as $n \to \infty$.

6. PROOF OF THE ANSATZ FOR FIFO AND DHR SERVICES

We outline the proof of the *ansatz* when the service discipline is FIFO and the services have a decreasing hazard rate (DHR). These assumptions will be in force throughout this section, unless stated otherwise. The proof has several steps, which, due to a shortage of space, we only sketch here. We highlight the crucial role played by the FIFO and DHR assumptions.

Consider an n-queue load balancing system and let $Q^n(t) = (q^{1,n}(t), ..., q^{n,n}(t))$ be the queue size process and $E^n(t) = (e^{1,n}(t), ..., e^{n,n}(t))$ be the *elapsed* service time of the first job in each of the n queues at time t. The state of the Markov process corresponding to the load balancing system is given by $(Q^n(t), E^n(t))$. Note that this is different from the state representation $(Q^n(t), R^n(t))$ used in Section 2. It is crucial for facilitating some coupling arguments that we consider the elapsed time of a job rather than its residual time.

For each n, the Markov process $(Q^n(t), E^n(t))$ is positive recurrent; let $(\Pi^n, \mathcal{E}^n)$ be its equilibrium distribution and $\Pi^{n,(k)}$ be the restriction of Π^n to its first k co-ordinates. In order to discuss the convergence of distributions in the rest of the section, we need to introduce the following metric.

Fix an n and let $x^n = (q^n, e^n)$ denote a point in $Z_+^n \times R_n^n$. Such an x^n denotes the state of the Markov process $(Q^n(t), E^n(t))$. Let $h(s)$ denote the hazard rate of the service time distribution, that is,

$$h(s) = \lim_{\delta \downarrow 0} \frac{P(S \in (s, s+\delta))}{\delta}.$$

Denote by
(i) s_∞ the smallest value of s at which $\inf_s h(s)$ is attained. This may be ∞ as happens, for example, if $P(S > x) = x^{-\beta}$ or $P(S > x) = e^{-x}$.
(ii) $r(s)$ the mean residual time of a job which has already received s units of service.

For defining the metric below, we require that $e^{i,n} \leq s_\infty$. Define the norm on $Z_+^n \times R_n^n$ as follows:

$$||x^n|| = \sum_{i=1}^{n} ||x^{i,n}||, \tag{6.1}$$

where

$$||x^{i,n}|| = m(q^{i,n} - 1)_+ + r(e^{i,n}).$$

Here m is the mean service time, which we take to be 1. The term $m(q^{i,n} - 1)_+$ measures the mean service time still required for all completely unserved jobs in queue i and $r^{i,n}$ measures the mean residual time of the job at the head of queue i.

Similarly, for two points x_1 and x_2 on $Z_+^n \times R_n^n$, we set

$$d^n(x_1, x_2) = \sum_{i=1}^{n} d^{i,n}(x_1, x_2), \tag{6.2}$$

where

$$\begin{aligned} d^{i,n}(x_1, x_2) &= m\left|(q_2^{i,n} - 1)_+ - (q_1^{i,n} - 1)_+\right| \\ &\quad + \left|r\left(e_2^{i,n}\right) - r\left(e_1^{i,n}\right)\right|. \end{aligned}$$

The convergence of measures in the theorems below is with respect to the above metric.

THEOREM 3. *Consider a load balancing system operating under the SQ(d) policy with FIFO service discipline and DHR service times.*
(a) Then $(\Pi^{(k)}, \mathcal{E}^{(k)}) = \lim_{n\to\infty}(\Pi^{n,(k)}, \mathcal{E}^{n,(k)})$ *exists.*
(b) Let $\pi = \Pi^{(1)}$. *Then* $\Pi^{(k)} = \bigotimes_{i=1}^{k} \pi$, *that is,* $\Pi^{(k)}$ *is IID with marginal distribution* π.

The proof of Theorem 3 will follow from Theorem 4.

THEOREM 4. *Assume the system is empty at time 0,* i.e., $(Q^n(0), E^n(0)) = (\mathbf{0}, \mathbf{0})$. *At any time* $t \geq 0$, *let* $(\Pi^n(t), \mathcal{E}^n(t))$ *be the distribution of* $(Q^n(t), E^n(t))$. *Then*

$$(\Pi^{(k)}, \mathcal{E}^{(k)}) = \lim_{t,n\to\infty} (\Pi^{n,(k)}(t), \mathcal{E}^{n,(k)}(t)). \tag{6.3}$$

Moreover, $\Pi^{(k)} = \bigotimes_{i=1}^{k} \pi$, *for* π *as in Theorem 3.*

Proof outline of Theorem 4.
Consider the process $(Q^n(t), E^n(t))$. It is a function of the initial condition $(Q^n(0), E^n(0))$, the arrival process to the load balancing system in $[0, t)$ and the services rendered to the jobs in the system during this time. Now, by the hypothesis of Theorem 4, $(Q^n(0), E^n(0))$ is IID, being the all zeros vector. Moreover, the service times of all jobs are IID. Therefore, in order to prove that the first k co-ordinates of $(Q^n(t), E^n(t))$ become independent as $n \to \infty$, we need to establish and use the fact that the actual arrival processes to the first k queues under the SQ(d) policy become asymptotically independent as $n \to \infty$. This is done using the branching process argument outlined in Section 6.2. This style of argument is referred to as "propagation of chaos" in the literature; see [5, 4, 15], for example. Note that this step only relies on the randomized load balancing policy SQ(d) and not on the DHR and FIFO assumptions.

Now, to generalize from the IID initial condition to the more general setting where $(Q^n(0), E^n(0))$ can be arbitrarily distributed, we use the monotonicity argument in Section 6.1. Essentially, this step consists of observing that the evolution of the load balancing system with *any* non-zero initial condition *stochastically dominates* the evolution of the same system with the all-zeros initial condition. This step uses the DHR and FIFO assumptions. Given the monotonicity property, uniform stability is used to show that the distance between the two evolutions of the load balancing system monotonically decreases with time, and uniformly in n, in the metric defined in (6.2). This is used to conclude that for n and t large enough, the evolution of the load balancing system under the arbitrary initial condition is close to being IID. The complete proof will be presented in subsequent publications.

6.1 Monotonicity

Consider two FIFO queues serving jobs with a common DHR service distribution. Suppose that, at a given time, neither queue is empty. Denote by e_i and d_i, $i = 1, 2$, the

elapsed and departure times of the first job at queue i. The following lemma is a consequence of DHR service times.

LEMMA 1. *Suppose $e_1 > e_2$. Then there exists a coupling such that $d_1 > d_2$.*

DEFINITION 3. *Let (Q_1, E_1) and (Q_2, E_2) be two random vectors taking values in $Z_+^n \times R_+^n$. Then $(Q_1, E_1) \geq_{st} (Q_2, E_2)$ means there exists a coupling under which every entry of (Q_1, E_1) is larger than or equal to the corresponding entry of (Q_2, E_2).*

The following lemma establishes an important monotonicity property under the FIFO and DHR assumptions.

LEMMA 2. *Consider two n-queue load balancing systems that have common arrivals, with each arriving job having the same service time. If $(Q_1(s), E_1(s)) \geq_{st} (Q_2(s), E_2(s))$ for some time s, then $(Q_1(t), E_1(t)) \geq_{st} (Q_2(t), E_2(t))$ for all $t > s$.*

Remark 3: The proof proceeds similarly to a well-known argument in the case of exponential service times. Lemma 1 is a key component of the proof, since it ensures that departure times do not violate the inequality.

COROLLARY 1. *Consider an n-queue load balancing system. Suppose the system is empty at time 0. Then, for $s < t$,*

$$(Q^n(s), E^n(s)) \leq_{st} (Q^n(t), E^n(t)).$$

Corollary 1 states that the measures $(\Pi^n(t), \mathcal{E}^n(t))$ are monotonically increasing in t. The corollary follows from Lemma 2 since $(Q^n(t-s), E^n(t-s)) \geq_{st} (Q^n(0), E^n(0))$.

6.2 Independence via Branching Process

To simplify matters somewhat, we do this for $k = 2$; accordingly, we fix a $t > 0$ and consider $q^{1,n}(t)$ and $q^{2,n}(t)$. Recall that the system is empty at time 0. It will be useful to employ the terminology *selection set* of an arrival to denote the set of d queues that it chooses.

The basic idea is to consider a random "influence set", consisting of queues, that governs the state of queues 1 and 2 at time t. The influence set for queue i is rooted at queue i at time t and increases monotonically, going backwards in time. It increases when the selection set of an arrival intersects with the influence set, at which time the influence set increases to include the entire selection set. One can show that the size of the influence set is dominated by an appropriate d-ary branching process. Since the branching process increases exponentially in time, by time t, it has of the order e^{Ct} nodes. Therefore, for large enough n, the probability will be arbitrarily small that the influence sets corresponding to queues 1 and 2 ever intersect. Moreover, the probability that a selection ever intersects the influence set at more than one queue will be arbitrarily small. Consequently with overwhelming probability, the influence set will be a tree, whose law does not depend on n.

Since $q^{1,n}(t)$ is a function of the arrivals and their service times that occur at the queues belonging to its influence set, by the preceding discussion, $q^{i,n}(t)$, $i = 1, 2$, converges in distribution as $n \to \infty$. Moreover, $q^{1,n}(t)$ and $q^{2,n}(t)$ become independent as $n \to \infty$.

7. CONCLUSION AND FURTHER WORK

The paper presented a modularized program for the analysis of randomized load balancing systems with general service time distributions. This program, particularly the *ansatz*, significantly enhances the analyzability of the popular supermarket model. The queue at the cavity bridges the structural and computational parts of the program. It facilitates the analysis of complicated randomized load balancing policies under general service disciplines and service distributions. In this paper, the program has been used to show that under the SQ(d) policy the PS and LIFO-PR service disciplines are insensitive in the large n limit. Moreover, it has been used to discover threshold phenomena for the FIFO service discipline: The tail of the queue size decays polynomially, exponentially or doubly exponentially depending on the tail of the service distribution and the number of the samples, d.

We have only obtained a proof of the *ansatz* for a load balancing system operating under the SQ(d) policy with the FIFO service discipline and DHR services. However, simulations show that it appears to be true under very general assumptions.

Some further work: Obviously, a proof of the *ansatz* for the SQ(d) policies would be good to obtain and this may yield a new technical insight or a new approach. As mentioned, we have only used the most basic form of the Cavity Method in this problem. Generalizations of the load balancing model to problems where the queues are vertices in a graph (say the D-regular graph) would need a stronger form of the Cavity Method. Other scheduling problems in networking, where the system size increases, would also make candidates for an application of the Cavity Method, for example, load balancing with migration penalties or cycle stealing.

8. REFERENCES

[1] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced allocations. In *SIAM Journal on Computing*, pages 593–602, 1994.

[2] T. Bonald and A. Proutiere. Insensitivity in processor-sharing networks. *Performance Evaluation*, 49:193–209, 2002.

[3] M. Bramson. Stability of join the shortest queue networks. *submitted to the Annals of Probability.*

[4] C. Graham. Chaoticity on path space for a queueing network with selection of the shortest queue among several. *Journal of Appl. Prob.*, 37:198–211, 2000.

[5] C. Graham and S. Méléard. Chaos hypothesis for a system interacting through shared resources. *Probab. Theory Relat. Fields*, 100:157–173, 1994.

[6] F. P. Kelly. *Reversiblity and Stochastic Networks.* John Wiley and Sons Ltd, 1979.

[7] L. Kleinrock. *Queueing Systems. Volume1: Theory.* Wiley-Interscience, 1975.

[8] M. Luczak and C. McDiarmid. On the power of two choices: Balls and bins in continuous time. *The Annals of Applied Probability*, 15(3):1733–1764, 2005.

[9] M. Luczak and C. McDiarmid. On the maximum queue length in the supermarket model. *The Annals of Probability*, 34(2):493–527, 2006.

[10] J. B. Martin and Y. M. Suhov. Fast jackson networks. *Ann. Appl. Prob.*, 9(4):840–854, 1999.

[11] M. Mezard, G. Parisi, and M. Virasoro. Spin glass theory and beyond. *Physics Today*, 41:109, 1988.

[12] M. Mitzenmacher. The power of two choices in randomized load balancing. *Ph.D. thesis, Berkeley*, 1996.

[13] M. Mitzenmacher. Studying balanced allocations with differential equations. *Combin. Probab. Comput.*, 8:473–482, 1999.

[14] M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: A survey of techniques and results. *Handbook of Randomized Computing: volume 1. edited by P. Pardalos, S. Rajasekaran, and J. Rolim*, pages 255–312, 2001.

[15] A. S. Sznitman. Propagation of chaos. *Ecole d'ete Saint-Flour. Lect. Notes Math.*, 1464:165–251, 1989.

[16] M. Talagrand. *Spin Glasses: A Challenge for Mathematicians. Cavity and Mean Field Models.* Springer, 2000.

[17] B. Vocking. How asymmetry helps load balancing. In *IEEE Symp. Found. Comp. Sci*, pages 131–140, 1999.

[18] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Probl. Inf. Transm*, 32(1):20–34, 1996.