On a Class of Push and Pull Strategies with Single Migrations and Limited Probe Rate

Wouter Minnebo[a], Tim Hellemans[a], Benny Van Houdt[a,*]

[a]*Department of Mathematics and Computer Science, University of Antwerp - imec, Middelheimlaan 1, B-2020 Antwerp, Belgium*

Abstract

In this paper we introduce a general class of rate-based push and pull load balancing strategies, assuming there is no central dispatcher and nodes rely on probe messages for communication.

Under a pull strategy lightly loaded nodes send random probes in order to discover heavily loaded nodes, if such a node is found one task is transferred. Under a push strategy the heavily loaded nodes attempt to locate the lightly loaded nodes.

We show that by appropriately setting its parameters, rate-based strategies can be constructed that are equivalent with traditional or d-choices strategies.

Traditional strategies send a batch of L_p probes at task arrival (push) or completion times (pull), whereas rate-based strategies send probes according to an interrupted Poisson process. Under the centralized/distributed d-choices strategy, d or $d-1$ probes are sent in batch at arrival times and the task is transferred to the shortest queue discovered.

We derive expressions for the mean delay for all considered strategies assuming a homogeneous network with Poisson arrivals and exponential job durations under the infinite system model.

We compare the performance of all strategies given that the same overall probe rate is used. We find that a rate-based push variant outperforms d-choices in terms of mean delay, at the cost of being more complex. A simple pull strategy is superior for high loads.

Keywords: Performance analysis, Distributed computing, Processor scheduling, Load balancing

[*] Corresponding author

Email addresses: `wouter.minnebo@uantwerpen.be` (Wouter Minnebo), `tim.hellemans@uantwerpen.be` (Tim Hellemans), `benny.vanhoudt@uantwerpen.be` (Benny Van Houdt)

1. Introduction

Minimizing queuing delays of tasks in distributed networks is increasingly relevant due to the explosive growth of cloud computing. Cloud applications typically use a large number of servers, and even a small increase in delay can result in the loss of users and revenue [1].

Traditionally, distributed applications use a single load balancer to distribute incoming tasks among available servers. In this case join-the-shortest-queue is a straightforward strategy [2]. However, this requires that the load balancer is aware of all the queue lengths in the system. As the system grows in size this becomes impractical, especially if multiple load balancers use the same server pool. A practical solution when using multiple load balancers is join-the-idle queue [3], where idle servers inform a well chosen load balancer of their idle state. When there is an incoming task, the load balancer forwards it to an idle server if one is known at that time. This is closely related with the asymptotically optimal PULL proposed in [4], which uses a single load balancer.

Another approach using a centralized load balancer, called d-choices, lets the load balancer sample d queue lengths and forwards the task to the least loaded server. This policy does not require knowledge of all queue lengths at all times, and improves the queuing delays dramatically compared to randomized load balancing. This strategy is also known as the power-of-d-choices and is widely studied [5, 6, 7, 8]. When tasks arrive in batch, it is advantageous to sample multiple servers and distribute the batch over the discovered servers instead of treating each task separately [9].

In other systems tasks enter the network via the processing nodes themselves (e.g., [10, 11, 12, 13]) without an explicit load balancer. In such case, strategies to reduce the delay fall into two categories: pull and push. Under a pull strategy (or load stealing) the lightly loaded servers attempt to contact and migrate tasks from heavily loaded servers. Under a push strategy (or load sharing) it is the heavily loaded nodes that take the initiative to locate lightly loaded servers.

Nodes typically communicate via probe messages to exchange queue length information. In order to locate a target queue to migrate a tasks to/from, a random node is probed and its queue length will determine whether the transfer is allowed.

We further distinguish between traditional strategies which send a batch of probes at task arrival (push) or completion times (pull), and rate-based strategies which send probes periodically. We note that for some systems it is not feasible to migrate tasks after the initial server assignment. Therefore, the rate-based strategies are more suited for computational workloads where the cost of migration is small, as opposed to web services where TCP connections have to be migrated along with the task [3].

The performance of these classes of strategies has been studied by various authors. Results presented in [10, 14] compare several push and pull strategies for a homogeneous distributed system with Poisson arrivals and exponential job lengths, and extensions to heterogeneous systems are presented in [15, 16]. Load stealing is also commonly used in the context of shared-memory multiprocessor scheduling [17].

These studies showed that the pull strategy is superior under high load conditions, whereas the push strategy achieves a lower mean delay under low to moderate loads.

When comparing different strategies, one aspect to keep in mind is the number of probes required by the strategy. Clearly, allowing a strategy to send more probes should improve its performance. However, not all strategies can set their parameters as to match an arbitrary overall probe rate. Comparing strategies with a different overall probe rate can be biased, as sometimes the strategy with the higher probe rate is best [18].

In [18] rate-based pull and push variants are introduced that can match any predetermined probe rate R, allowing the comparison of pull and push strategies when they use the same overall probe rate. In these variants, probes are sent at a fixed rate r as long as the server is idle (for pull) or has jobs waiting (for push). The main result in [18] showed that the rate-based push strategy results in a lower mean delay if and only if

$$\lambda < \frac{\sqrt{(R+1)^2+4(R+1)}-(R+1)}{2},$$

under the so-called infinite system model, and that a hybrid pull/push strategy is always inferior to the pure pull or push strategy.

In [19] the model of [18] was extended to only allow highly loaded nodes to send probes, instead of all busy nodes. A node is considered highly loaded if it has more than T jobs. This allowed the construction of the max-push strategy that extended the range of λ values where the push variants outperformed the pull strategy.

In previous work tasks could only be migrated to an empty server. However, for higher loads it becomes harder to find an empty server. In this situation a migration to a server that is lightly loaded but not empty can further reduce the mean delay. Therefore, we extend both the traditional and rate-based model to allow transfers to lightly loaded nodes, in this case nodes with at most B jobs. Setting $B = 0$ only allows transfers to empty servers, reducing the models and closed form expressions to those found in previous work [19, 18, 13].

Furthermore, we develop several push models that achieve the same performance as the d-choice strategy when using the same number of probes, but without centralized load balancers.

This paper makes the following contributions:

1. We introduce a general class of push and pull strategies, and describe its evolution in an infinite system model. We identify several subclasses by restricting the model parameters. For these subclasses, we find the stationary queue length distribution, allowing us to express the mean delay explicitly. Furthermore, we state as conjecture an optimal pull and push strategy for this general class of strategies.
2. We show that rate-based strategies achieve the same level of performance compared to traditional strategies, when using the same overall probe rate.

Therefore, systems where it might be desirable to not send the probes at task arrival or completion instants are not at a disadvantage. In addition, rate-based strategies allow for more granular control over the overall probe rate, whereas the number of probes in a batch must be an integer for the traditional strategies.

3. We introduce several distributed versions of d-choices with an overall probe rate of $\lambda^2(1-\lambda^{d-1})/(1-\lambda)$, that are equivalent in performance compared to a centralized d-choices with d probes per task.
4. We show that a rate-based push variant has a lower mean delay than d-choices, and the pull strategy remains best for high loads.

The paper is structured as follows. In Section 2 we give an overview of the strategies considered in this paper. Section 3 presents the infinite system models for a general rate-based push and pull strategy, considers a subclass corresponding to a particular choice of parameters, and covers the max-push strategy. Section 4 analyses the traditional pull and push strategies, and shows the equivalence with rate-based strategies. This equivalence was shown in [18] for $T = 1$ and $B = 0$. In section 5 we introduce a distributed version of the d-choices strategy, and derive two rate-based variants that are equivalent to the original d-choices strategy with respect to their stationary distribution. In Section 6, the best performing rate-based pull and push strategies are compared to the d-choice strategy.

2. Problem Description and Overview of Strategies

We consider a continuous-time system consisting of N queues, where each queue consists of a single server with an infinite buffer. As in [10, 20, 15, 12], jobs arrive locally according to a Poisson process with rate $\lambda < 1$, and have an exponentially distributed duration with mean 1. Servers process jobs in first-come first-served order. Servers can send probe messages to each other to query for queue length information and to transfer jobs. We assume that the time required to transfer probe messages and jobs is sufficiently small in comparison with the processing time of a job, i.e., transfer times are considered zero. For a discussion on the impact of communication delay and transfer time we refer to [21, 22].

1. *Rate-based Push/Pull:* Whenever a server has i tasks it generates probe messages according to a Poisson process with rate r_i. The node with length j that is probed is selected at random and the transfer of a job from the server with the longest to the server with shortest queue length is allowed if $a_{i,j} = 1$, while no transfer takes place if $a_{i,j} = 0$. We will study several subclasses of this class.
2. *Traditional Push:* For every task arrival that would bring the queue length above T, the server first sends up to L_p probes in sequence. The task is forwarded to the first discovered server with queue length B or less. If no such server is found, the task is processed by the original server.

3. *Traditional Pull:* For each task completion that would bring the queue length to B or less, the server first sends up to L_p probes in sequence. A task is migrated from the first discovered server with queue length above T. If no such server is found, no further action is taken.
4. *Distributed d-Choices*: Nodes send $d-1$ probes on a task arrival instant and forward the job to the least loaded probed node, or process the task themselves if no shorter queue is found.
5. *Push-d-batch*: All servers that have tasks waiting generate probe events according to a Poisson process with rate r_i, where i is the queue length. During each probe event, a batch of $d-1$ probes is sent and a task is migrated to the least loaded probed node if its queue length is smaller than $i-1$.

We study the different strategies using an infinite system model, i.e. as the number of queues in the system (N) tends to infinity. In previous work [18, 19, 23] we observed that the infinite system model is an accurate approximation for the finite case. A relative error of a few percent or less was observed when predicting the mean delay for $N \geq 100$. We make similar observations in Sections 3.2 and 3.5 for the strategies introduced in this paper.

3. Rate-based Strategies

In this section we introduce the infinite system model to assess the performance of rate-based push and pull strategies. First let us define a general rate-based strategy belonging to the class $\mathcal{S}(\mathbf{r}, A)$, with $\mathbf{r}$ a vector $(r_0, r_1, \ldots)$ and A a binary matrix with elements $a_{i,j}$. The elements r_i of vector $\mathbf{r}$ indicate at which rate a queue with length i sends random probes. The elements $a_{i,j}$ of matrix A indicate whether a probe from a queue with length i to a queue with length j results in a task transfer. This class of strategies only allows the transfer of a single task per probe. We refer to a strategy as a pull strategy if $a_{i,j} = 1$ implies $i < j$, and as a push strategy if $a_{i,j} = 1$ implies $i > j$.

The evolution of the queue lengths under this general strategy is modeled by a set of ODEs denoted as $dx(t)/dt = D(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)) + \hat{\alpha} + \hat{\beta} - \hat{\gamma} - \hat{\delta} \tag{1}$$

with $x_0(t) = 1$, and

$$\hat{\alpha} = (x_{i-1}(t) - x_i(t)) \sum_{j=i+1}^{\infty} r_j (x_j(t) - x_{j+1}(t)) a_{j,i-1}$$

$$\hat{\beta} = r_{i-1}(x_{i-1}(t) - x_i(t)) \sum_{j=i+1}^{\infty} (x_j(t) - x_{j+1}(t)) a_{i-1,j}$$

$$\hat{\gamma} = r_i(x_i(t) - x_{i+1}(t)) \sum_{j=0}^{i-2} (x_j(t) - x_{j+1}(t)) a_{i,j}$$

$$\hat{\delta} = (x_i(t) - x_{i+1}(t) \sum_{j=0}^{i-2} (x_j(t) - x_{j+1}(t)) r_j a_{j,i}$$

The terms $\lambda(x_{i-1}(t) - x_i(t))$ and $(x_i(t) - x_{i+1}(t))$ indicate arrivals and completions, respectively. The term $\hat{\alpha}$ indicates incoming transfers to queues with length $i - 1$ resulting from push request by longer queues. The term $\hat{\beta}$ indicates incoming transfers to queues with length $i - 1$ resulting from pull requests by those queues. The term $\hat{\gamma}$ indicates outgoing transfers resulting from push requests made by queues with length i. The term $\hat{\delta}$ indicates outgoing transfers resulting from pull requests made by shorter queues to queues with length i.

An interesting question regarding the infinite system model is whether it corresponds to the limit as N tends to infinity of the sequence of rescaled Markov processes, where process N corresponds to a system consisting of N servers. This question is typically answered in two steps: (1) does the set of ODEs describes the proper limit process of the corresponding finite systems for any finite time horizon $[0, T]$ and (2) does the convergence extend to the stationary regime? For the fixed rate pull and push strategies introduced in the next subsection with $B = 0$ and $T \geq 0$, both these questions were answered affirmatively in [18, 19]. In Appendix A and B we shown that this is also the case for $B > 0$. While the main line of reasoning in Appendix B is similar to [18, 19], the proof methodology in Appendix A is not and relies for the most part on the approach taken in [6]. It may be possible to further generalize this result, but we have not pursued this further.

In the next sections we simplify Equation (1) by restricting the choice of $\mathbf{r}$ and A, resulting in explicit expressions for the unique fixed point and mean delay.

3.1. Fixed Rate Push and Pull

To restrict the set of pull and push strategies we state that a queue is long if it contains more than T tasks and that a queue is short if it has at most B tasks, with $B < T$. In addition, only one group of queues (be it long or short) sends probes independently of the queue length with rate r. We do not consider hybrid strategies where both the long and short queues transmits probes in this section. In fact, Theorem 5 from [18] shows that a pure pull or push strategy is superior to any hybrid strategy when $B = 0$ and $T = 1$. Whether this result extends to $B > 0$ and/or $T > 1$ is an interesting open problem. We allow only transfers from long queues to short queues. In other words, for the fixed rate pull strategy

$$\begin{cases} r_i = 0 & \text{if } i > B \\ r_i = r & \text{if } i \leq B \end{cases}$$

and $a_{i,j}$ is one if $i \leq B$ and $j > T$ and zero otherwise. Likewise, for the fixed rate push strategy

$$\begin{cases} r_i = 0 & \text{if } i \leq T \\ r_i = r & \text{if } i > T \end{cases}$$

and $a_{i,j}$ is one if $i > T$ and $j \leq B$ and zero otherwise.

The evolution of both the fixed rate pull and push strategy is modeled by a set of ODEs denoted as $dx(t)/dt = F(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. This is a simplification of Equation (1), and the ODEs can be written as

$$\frac{dx_i(t)}{dt} = (\lambda + r x_{T+1}(t))(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{2}$$

for $1 \leq i \leq B+1$ with $x_0(t) = 1$, and

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{3}$$

for $B + 2 \leq i \leq T$, and

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)) - r(1 - x_{B+1}(t))(x_i(t) - x_{i+1}(t)) \tag{4}$$

for $i > T$.

In the next Theorem we express the fixed point for this set of ODEs in $E = \{(x_i)_{i\geq 0} | 1 = x_0 \geq x_1 \geq \ldots \geq 0, \sum_{i\geq 1} x_i < \infty\}$.

Theorem 1. *The set of ODEs given by (2-4) has a unique fixed point $\pi = (\pi_1, \pi_2, \ldots) \in E$ for $\lambda < 1$. Let $\eta_i = \pi_i - \pi_{i+1}$ and $\eta_0 = 1 - \lambda$, then the fixed point can be expressed as*

$$\eta_i = (1-\lambda)(\lambda + r\pi_{T+1})^i, \qquad 1 \leq i \leq B+1 \tag{5}$$

$$\eta_i = \eta_{B+1}\lambda^{i-(B+1)}, \qquad B+2 \leq i \leq T \tag{6}$$

$$\eta_i = \eta_T \left(\frac{\lambda}{1 + r(1 - \pi_{B+1})}\right)^{i-T}, \qquad i > T. \tag{7}$$

Further π_{T+1} is the unique root on $(0, \lambda^{T+1})$ of the ascending function

$$f(x) = (x-1) + (1-\lambda)\sum_{i=0}^{B} u(x)^i + (1 - \lambda^{T-B})u(x)^{B+1},$$

where $u(x) = \lambda + rx$ and $1 - \pi_{B+1} = (1-\lambda)\sum_{i=0}^{B} u(\pi_{T+1})^i$.

Proof. The expressions for η_i readily follow from setting $dx_i(t)/dt = 0$ in Equations (2-4), and observing that $\pi_1 = \lambda$ due to the requirement $\sum_{i\geq 1} \pi_i < \infty$.

The requirement $\sum_{i\geq 1} \pi_i < \infty$ implies that $\sum_{i\geq 1} \eta_i = 1$, which can be restated as

$$(1-\lambda)\sum_{i=0}^{B} u(\pi_{T+1})^i + u(\pi_{T+1})^{B+1}(1-\lambda^{T-B}) + \pi_{T+1} = 1.$$

Thus, $f(\pi_{T+1}) = 0$ and $f(x)$ is increasing on $(0,1)$ as $u(x)$ is increasing on $(0,1)$. Finally, $f(0) = -\lambda^{T+1}$ and $f(\lambda^{T+1}) \geq 0$ as $u(\lambda^{T+1}) \geq \lambda$.

□

When $B = 0$ the root of $f(x)$ correspond to the root of a linear equation and therefore π_{T+1} has a simple explicit form, i.e., $\pi_{T+1} = \lambda^{T+1}/(1+r(1-\lambda^T))$. For small B, e.g., $B = 1$, it is still possible to find an explicit expression for π_{T+1}, but this expression does not appear to be very elegant. Instead we suggest to use any root finding algorithm on $(0, \lambda^{T+1})$ to determine π_{T+1} when $B > 0$.

We can now express the main performance measures of these push and pull strategies. First, we note that the overall probe rate for push strategies equals

$$R_{push} = r_{push}\pi_{T+1}, \tag{8}$$

as all queues with length $T+1$ or more send probes with rate r_{push}. Similarly, the overall probe rate of the pull strategy equals

$$R_{pull} = r_{pull}(1-\pi_{B+1}), \tag{9}$$

as all queues with length B or less send probes with rate r_{pull}. The behavior of R_{Push} and R_{Pull} for a varying r with fixed λ is shown in Figures 1 and 2. Furthermore, the total migration rate is

$$M = r(1-\pi_{B+1})\pi_{T+1}.$$

From a push perspective a fraction of nodes (π_{T+1}) sends probes at rate r, succeeding with probability $(1-\pi_{B+1})$. From a pull perspective the roles of senders and receivers are reversed. Now we can formulate the mean delay:

Theorem 2. *The mean delay D of a job under the fixed rate push or pull strategy equals*

$$D_{both} = \frac{1}{1-\lambda}\left(1 - \frac{M}{\lambda}\gamma\right),$$

with

$$\gamma = T - B + \alpha + \delta, \qquad \alpha = \sum_{i=T+2}^{\infty} \frac{(i-(T+1))\eta_i}{\pi_{T+1}} = \frac{\lambda}{1-\lambda+r(1-\pi_{B+1})},$$

$$\begin{aligned}\delta &= \sum_{i=0}^{B-1} \frac{(B-i)\eta_i}{1-\pi_{B+1}} \\ &= \frac{(1-\lambda)(B(1-\lambda-r\pi_{T+1}) - (\lambda+r\pi_{T+1})(1-(\lambda+r\pi_{T+1})^B))}{(1-\pi_{B+1})(1-\lambda-r\pi_{T+1})^2}.\end{aligned}$$

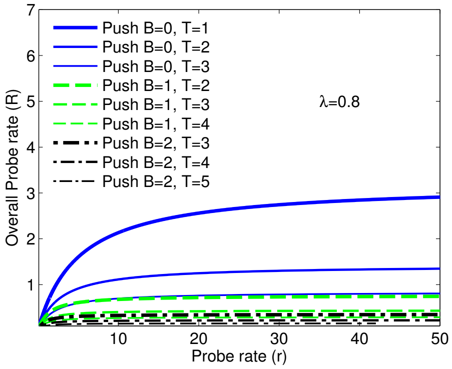

Figure 1: Resulting overall probe rate R when varying the individual probe rate r for push strategies with different settings of B and T, for a fixed load λ.

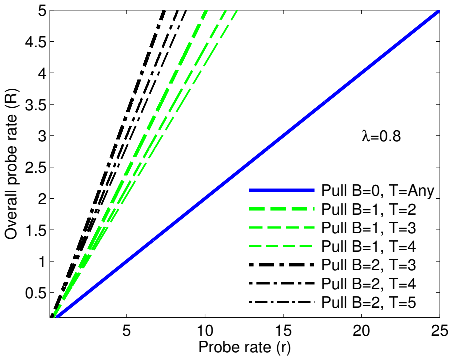

Figure 2: Resulting overall probe rate R when varying the individual probe rate r for pull strategies with different settings of B and T, for a fixed load λ.

Proof. We use a similar argument as in [23], which showed that the improvement over the mean delay of an M/M/1 queue can be formulated as a migration frequency (M/λ) and migration gain (γ). The migration frequency denotes how many migrations per job take place on average and the migration gain quantifies the number of places in the queue the migrating job skips. Therefore, the total improvement is given by how many migrations take place on average per task multiplied by how many places in the queue a migrating task skips.

Migrating tasks skips on average γ places in the queue. All tasks skip $T-B$ places by construction of the strategy. Tasks can skip more places depending on the length of the queue sending the task, accounting for α places on average. We note this equals the average number of customers in an M/M/1 queue with service rate $1+r(1-\pi_{B+1})$. Tasks can also skip more places depending on the length of the queue receiving the task, accounting for δ places on average. □

When comparing the pull and push strategy for a fixed R, we need to set r such that R attains the target value. For the pull strategy this is trivial, one simply sets $r = R/(1-\lambda)$. For the push strategy this problem can be solved by substituting $r\pi_{T+1}$ by R and computing the fixed point directly from (5-7). However, when R is relatively large this will result in a negative value for π_{T+1}. This indicates that queues can send probes at an infinite rate without exceeding the overall probe limit R, thereby instantly finding migration targets for all tasks from queues with length $T+1$ or more, and reducing π_{T+1} to zero. This observation is in agreement with Figure 1 where we observe that for the push strategy R does not appear to become infinitely large as r tends to infinity. This is further illustrated in Figure 3, where the load at which π_{T+1} reaches zero is marked with a dot. For all loads lower than this point the substitution we performed (using R instead of $r\pi_{T+1}$) is no longer valid, and computing π_{T+1} yields a negative result. In this case the push strategy with the current B, T and λ parameters uses less probes than allowed by the overall probe limit R, as

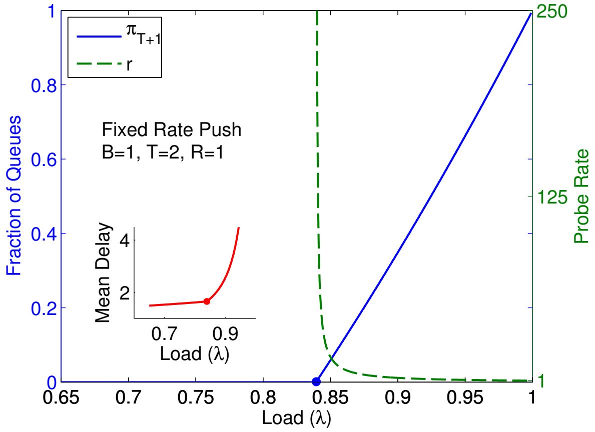

Figure 3: The probe rate of individual queues (r) and the fraction of queues allowed to send probes (π_{T+1}), shown for the fixed rate push strategy with $B = 1, T = 2$ and $R = 1$. The probe rate r goes to infinity as the fraction of queues with at least $T+1$ tasks (π_{T+1}) reaches zero. The load λ at which this occurs is marked with a dot. This is also the point where the behavior of the mean delay changes, as shown in the inset plot. Increasing R results in a larger r and smaller π_{T+1} for any given load λ, so π_{T+1} reaches zero at a higher load. The converse is true for decreasing R.

all tasks that are eligible to migrate are instantly exhausted. The behavior of a push strategy with infinite r is equivalent with the max-push strategy with $r_{mp} = 0$, covered in Section 3.4.

Conjecture 1. *The optimal choice for a rate-based pull strategy in class $\mathcal{S}(\mathbf{r}, A)$ given an overall probe rate R is a fixed rate pull strategy with $B = 0$ and $T = 1$.*

In [19, Theorem 5] it was shown that if $B = 0$, setting $T = 1$ is optimal. Intuitively, increasing T makes it less likely that a probe is successful. Similarly, a non-empty server is just as likely to locate a queue with length at least T than an empty server. And the tasks can skip more places in the queue if the request was sent by the empty server. Therefore, we expect that setting $B = 0$ and $T = 1$ is optimal for the rate-based pull strategy. Figure 4 illustrates that setting $B = 0$ and $T = 1$ is indeed superior to some other choices for B and T when $R = 1$.

For the push strategy setting $B = 0$ is not optimal as shown in Figure 5. Increasing B improves the performance of the push under moderate to high loads. We observed that increasing T higher than $B + 2$ is not beneficial, as setting the parameter B to $B + 1$ yields a lower mean delay for that load. Therefore, such settings are not shown.

3.2. Numerical Validation of Fixed Rate Push

In this section we present validation results for the fixed rate push strategy with $B \geq 1$ as the model for push and pull strategies with $B = 0$ was already

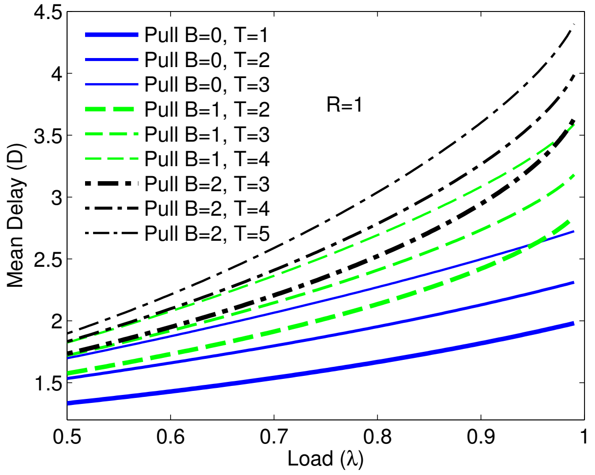

Figure 4: The mean delay of the pull strategy with $R = 1$ for different settings of B and T. Increasing either B or T results in a higher mean delay.

validated in [19] and we conjecture that the mean delay of the pull strategy is minimized for $B = 0$ and $T = 1$. The infinite system model and simulation setup only differ in the system size. The rate r_{push} in the simulation experiments is independent of N and was determined by λ and R using the expression for R_{push} in (8), we choose $R = 1$ in all experiments. Each entry in the tables represents the average value of 25 simulation runs. Each run has a length of 10^6 (where the service time is exponentially distributed with mean 1) and a warm-up period of length $10^6/3$.

Table 1 shows the relative error in mean delay, observed when comparing a finite system with size N to the infinite system model. As expected, the error decreases as the system grows in size, with at most a few percent relative error as the system reaches 100 nodes. Changing values for T and B can either increase or decrease the error. For example taking $B = 1$ and $\lambda = 0.90$, increasing T from 2 to 3 decreases the error, but with $\lambda = 0.95$ the same change increases the error. The error also increases with the load. The infinite system model is optimistic, underestimating the observed mean delay.

We should note that the actual overall probe rate observed in the finite system exceeds the requested R, as shown in Table 2. In other words, the relation between R_{push} and r_{push} given by (8) is not very accurate for small N values as the infinite model is optimistic with respect to the queue length distribution. However, as the finite system grows in size, the actual overall probe rate converges to the one requested.

3.3. Limiting the Individual Probe Rate (r)

In the previous sections we compared strategies by limiting the overall probe rate (R). However, another factor to take into consideration is the rate at

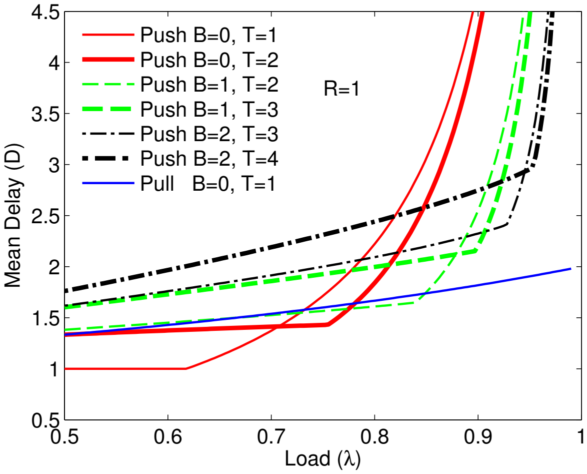

Figure 5: Mean delay of the push strategy, with $T = B + 1$ and $T = B + 2$ for $B = 0, 1, 2$. Also shown is the pull strategy with $B = 0$ and $T = 1$. All strategies use $R = 1$.

which individual servers send probes (r), as in a practical setting the individual servers might also have a maximum probe rate in addition to the overall probe rate constraint. As Equations (8) and (9) imply, r can be much higher than R. In this section we study the impact on the strategies' performance when introducing a maximum probe rate limit (r_{max}).

In Figures 6 and 7 the mean delay achieved by push strategies is shown when setting r_{max} to 10 and 50, respectively. These figures suggest that the relative loss in performance due to limiting r decreases with both B and T. For $B = 0$ and $T = 1$ we can derive a bound on the performance loss due to limiting r as follows. In [18, Corollary 1] the mean delay for the push strategy with $B = 0, T = 1$ is expressed as

$$D(\lambda, r) = 1 + \frac{\lambda}{(1-\lambda)(1+r)},$$

and the highest load λ_{D1} where this mean delay equals one is determined by

$$\lambda_{D1} = \frac{1}{2}\sqrt{R^2 + 4R} - \frac{R}{2}.$$

At this load the relative loss in performance is the highest, so

$$\frac{D(\lambda_{D1}, r_{max}) - 1}{1} = \frac{R + \sqrt{R(R+4)}}{2r_{max} + 2}$$

is an upper bound for the relative loss in performance when $B = 0$ and $T = 1$.

In Figures 8 and 9 the mean delay achieved by pull strategies is shown when setting r_{max} to 10 and 50, respectively. As soon as the individual probe limit is reached, the performance quickly declines compared to the case where r is

N	$B = 1$					$B = 2$
	$T = 2$			$T = 3$		
	$\lambda = 0.85$	$\lambda = 0.90$	$\lambda = 0.95$	$\lambda = 0.90$	$\lambda = 0.95$	$\lambda = 0.95$
25	4.15e-2	8.78e-2	1.17e-1	5.42e-2	1.28e-1	1.43e-1
50	1.70e-2	4.21e-2	5.77e-2	2.00e-2	6.28e-2	6.60e-2
100	7.68e-3	2.07e-2	2.92e-2	7.95e-3	3.12e-2	3.17e-2
200	3.60e-3	1.04e-2	1.50e-2	3.49e-3	1.58e-2	1.52e-2
400	1.76e-3	5.07e-3	7.19e-3	1.62e-3	7.68e-3	7.54e-3
800	8.74e-4	2.53e-3	3.76e-3	7.94e-4	3.88e-3	3.76e-3
1600	4.22e-4	1.25e-3	1.79e-3	3.96e-4	2.04e-3	1.94e-3

Table 1: The relative error of the mean delay D in a finite system with size N using the fixed rate push strategy, compared to the infinite system model. We note that the infinite system model is optimistic with respect to the performance of the finite system.

N	$B = 1$					$B = 2$
	$T = 2$			$T = 3$		
	$\lambda = 0.85$	$\lambda = 0.90$	$\lambda = 0.95$	$\lambda = 0.90$	$\lambda = 0.95$	$\lambda = 0.95$
25	4.75e-1	1.05e-1	2.80e-2	1.45e+0	7.13e-2	2.67e-1
50	1.98e-1	5.36e-2	1.40e-2	5.27e-1	3.69e-2	1.41e-1
100	8.75e-2	2.74e-2	7.37e-3	1.96e-1	1.88e-2	7.30e-2
200	4.07e-2	1.40e-2	3.87e-3	8.20e-2	9.85e-3	3.65e-2
400	1.98e-2	6.86e-3	1.80e-3	3.73e-2	4.71e-3	1.85e-2
800	9.75e-3	3.42e-3	9.82e-4	1.79e-2	2.39e-3	9.32e-3
1600	4.78e-3	1.71e-3	4.62e-4	8.80e-3	1.28e-3	4.81e-3

Table 2: The relative error of the overall probe rate R in a finite system with size N using the fixed rate push strategy, compared to the infinite system model. We note that when using the r as derived from the infinite system model, the finite system produces a higher overall probe rate than requested.

not limited. Interestingly, the setting $B = 0, T = 1$ is no longer optimal. For $\lambda > \frac{r_{max}-R}{r_{max}}$ the pull strategy with $B = 0, T = 1$ is not able to reach the overall probe limit since it is constrained by the individual probe limit. In this case an alternative strategy could be formulated, where queues with length at most B probe at rate r_{max} and the queues with length $B+1$ probe at the highest rate the overall probe limit allows, probes would result in a task transfer if a server with at least $B+2$ tasks is found. We conjecture that such a strategy achieves a mean delay that connects the (λ, D) points on a graph where the individual probe limit is now reached for consecutive values of B, with $T = B+1$. However, a formal treatment of such strategy is deemed outside the scope of this paper.

3.4. The Max-Push Strategy

As we noted in Section 3.1, the fixed rate push strategy can not match the predefined overall probe rate in case R is larger than needed to instantly find

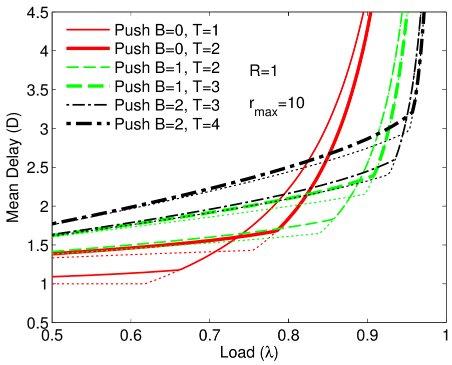

Figure 6: Mean delay of the push strategy with different settings for B and T, where $R = 1$ and $r_{max} = 10$. Dotted lines indicate the mean delay in case r is not limited.

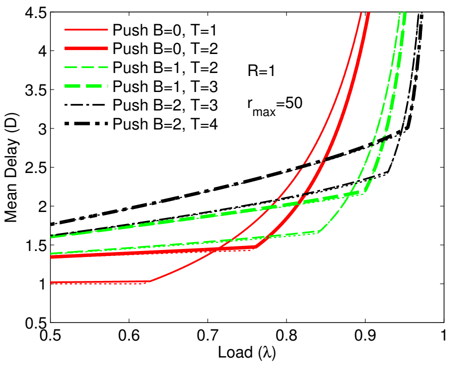

Figure 7: Mean delay of the push strategy with different settings for B and T, where $R = 1$ and $r_{max} = 50$. Dotted lines indicate the mean delay in case r is not limited.

migration targets for all tasks from queues with length $T + 1$ or more. This effectively eliminates all queues longer than T, without using the full R budget. The idea of the max-push strategy is to migrate all new arrivals at a queue with length T instantly to an eligible server, and let the queues with length exactly T probe with rate r_{mp}. We later show how to choose r_{mp}, B and T such that the resulting overall probe rate matches R.

Formally the max-push strategy is a member of $\mathcal{S}(\mathbf{r}, A)$ and defined as follows. Let $r_T = r_{mp}$ and $r_{T+1} = \infty$, with the other entries of $\mathbf{r}$ set to zero. Let $a_{i,j}$ be one in case i is either T or $T + 1$, and $j \leq B$.

In [19] the max-push strategy was introduced for $B = 0$, which we now generalize for $B > 0$. We discern two cases: $T > B + 1$ and $T = B + 1$.

In case $T > B + 1$, the evolution of the max-push strategy is given by a set of ODEs denoted as $dx(t)/dt = H(x(t))$, where $x(t) = (x_1(t), x_2(t), \dots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. This is an adaptation of Equation (1) and this set of ODEs can be written as

$$\frac{dx_i(t)}{dt} = \left(\lambda + \frac{\lambda x_T(t)}{1 - x_{B+1}(t)} + r_{mp} x_T(t)\right)(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{10}$$

for $1 \leq i \leq B + 1$, and

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{11}$$

for $B + 2 \leq i < T$, and

$$\frac{dx_T(t)}{dt} = \lambda(x_{T-1}(t) - x_T(t)) - x_T(t)(1 + r_{mp}(1 - x_{B+1}(t))). \tag{12}$$

Note that all new arrivals at queues of length T are migrated to servers with a maximum length of B, as indicated by $\lambda x_T(t)$ in (10). Probes are sent to random

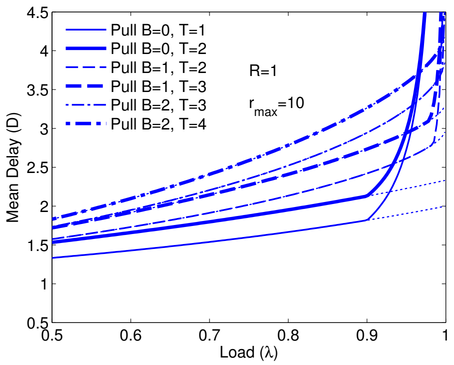

Figure 8: Mean delay of the pull strategy with different settings for B and T, where $R = 1$ and $r_{max} = 10$. Dotted lines indicate the mean delay in case r is not limited.

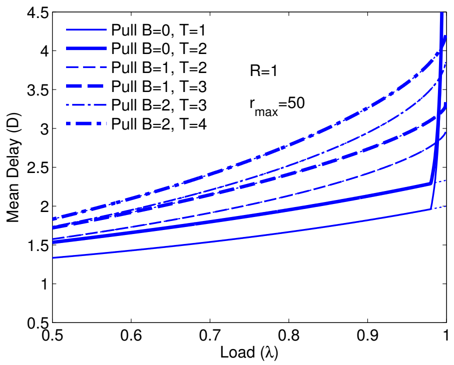

Figure 9: Mean delay of the pull strategy with different settings for B and T, where $R = 1$ and $r_{max} = 50$. Dotted lines indicate the mean delay in case r is not limited.

servers with equal probability for each server. Consequently, the migrations from new arrivals at queues of length T are uniformly distributed across servers with length B or less. Therefore, these migrations arrive at a queue with length $i-1$ with probability $(x_{i-1}(t) - x_i(t))/(1 - x_{B+1}(t))$, increasing the fraction of servers with queue length $i \leq B+1$.

For the case $T > B+1$ all migrations have the same target, specifically queues with length at most B. This is no longer true if we allow $T = B+1$. The new arrivals at a queue with length T can be migrated to any queue with length at most B. However, a probe from a queue with length T should find a target with length at most $B-1$ in order for the migration to result in a delay reduction. Therefore, the evolution of the system is described by a different set of ODEs given below.

In case $T = B+1$, the evolution of the max-push strategy is given by a set of ODEs denoted as $dx(t)/dt = I(x(t))$, where $x(t) = (x_1(t), x_2(t), \dots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\frac{dx_i(t)}{dt} = \left(\lambda + \frac{\lambda x_T(t)}{1 - x_T(t)} + r_{mp} x_T(t)\right)(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \quad (13)$$

for $1 \leq i \leq B$, and

$$\begin{aligned}\frac{dx_T(t)}{dt} =& \lambda(x_{T-1}(t) - x_T(t)) - x_T(t)(1 + r_{mp}(1 - x_{T-1}(t))) \\ &+ \frac{\lambda x_T(t)}{1 - x_T(t)}(x_{T-1}(t) - x_T(t)). \end{aligned} \quad (14)$$

The same remarks as for $H(x(t))$ apply, with a modification in $dx_T(t)/dt$. Queues with length T can now also be created by migrating an arrival in a queue with length T, to a queue with length $T-1$. This corresponds with the term $\lambda x_T(t)(x_{T-1}(t) - x_T(t))/(1 - x_T(t))$ in (14). Queues with length T ($x_T(t)$) again send probes with rate r, and are now successful with probability $1 - x_{T-1}(t)$.

The sets of ODEs $H(x(t))$ and $I(x(t))$ have a unique fixed point $\dot{\pi}$ and $\hat{\pi}$, respectively. We derive the formulas for these fixed points further on, expressing the overall probe rate and migration rate first.

For both cases ($T > B + 1$ and $T = B + 1$) the overall probe rate can be formulated as

$$R_{mp} = \frac{\lambda \breve{\pi}_T}{1 - \breve{\pi}_{B+1}} + r_{mp}\breve{\pi}_T, \tag{15}$$

with $\breve{\pi}_i$ equal to $\dot{\pi}$ or $\hat{\pi}$ depending on the value of T. This relation states the following: new arrivals at a queue of length T ($\lambda\breve{\pi}_T$) must find a server to migrate to, and find one on average by spending $1/(1 - \breve{\pi}_{B+1})$ probes. Queues with length T ($\breve{\pi}_T$) also send probes at the finite rate r_{mp}.

Similarly we can define the migration rate, i.e., the rate at which probes are successful:

$$\begin{aligned} M_{mp|T>B+1} &= \lambda\dot{\pi}_T + r_{mp}\dot{\pi}_T(1 - \dot{\pi}_{B+1}), \\ M_{mp|T=B+1} &= \lambda\hat{\pi}_T + r_{mp}\hat{\pi}_T(1 - \hat{\pi}_B). \end{aligned}$$

For both cases ($T > B + 1$ and $T = B + 1$) new arrivals at a queue with length T ($\lambda\breve{\pi}_T$) are migrated. The rest of the migrations are due to probes sent at rate r_{mp} by queues with length T ($\breve{\pi}_T$). These are successful with probability $(1 - \dot{\pi}_{B+1})$ in case $T > B + 1$ and with probability $(1 - \hat{\pi}_B)$ in case $T = B + 1$.

Having expressed the overall probe rate and migration rate, the fixed points are given in the next two theorems.

Theorem 3. *The set of ODEs given by (10-12) has a unique fix point* $\dot{\pi} = (\dot{\pi}_0, \ldots, \dot{\pi}_T) \in F := \{x \in \mathbb{R}^{T+1} \mid 1 = x_0 \geq \cdots \geq x_T \geq 0\}$. *Let* $\dot{\eta}_i := \dot{\pi}_i - \dot{\pi}_{i+1}$, *then one finds*

$$\begin{aligned} \dot{\eta}_i &= (1 - \lambda)(\lambda + R_{mp})^i, & 0 \leq i \leq B + 1 \\ \dot{\eta}_i &= \dot{\eta}_{B+1}\lambda^{i-(B+1)}, & B + 2 \leq i < T. \end{aligned}$$

Moreover $\dot{\pi}_{B+1}$ *is the unique root on* $(0, \lambda^{B+1})$ *of the ascending function:*

$$\dot{g}(x) = (x - 1) + (1 - \lambda)\sum_{i=0}^{B} \dot{u}(x, \dot{w}(x))^i,$$

with $\dot{u}(x, y) = \lambda + \frac{\lambda y}{1-x} + ry$ *and* $\dot{w}(x) = \frac{\lambda^{T-(B+1)}(1-\lambda)x}{(1-\lambda^{T-B}) + r(1-x)(1-\lambda^{T-(B+1)})}$. *The value of* $\dot{\pi}_T$ *is given by* $\dot{w}(\dot{\pi}_{B+1})$.

Proof. Using $\sum_{i=1}^{T} d\dot{\pi}_i/dt = 0$ we find that $\dot{\pi}_1 = \lambda$ and thus $\dot{\eta}_0 = 1 - \lambda$. The expressions for $\dot{\eta}_i$, for $1 \leq i < T$, easily follow from equations (10) and (11). We now show that $\dot{\pi}_{B+1}$ and $\dot{\pi}_T$ are uniquely determined. For ease of notation, we write $n = B + 1$ and $m = T - (B + 1)$. From equation (12) we find that:

$$\lambda^m(1 - \lambda)(\lambda + R_{mp})^n - \dot{\pi}_T(1 + r(1 - \dot{\pi}_{B+1})) = 0, \tag{16}$$

from $\sum_{i=0}^{B} \dot{\eta}_i = 1 - \dot{\pi}_{B+1}$ we find:

$$(\dot{\pi}_{B+1} - 1) + (1-\lambda)\sum_{i=0}^{n-1} (\lambda + R_{mp})^i = 0, \tag{17}$$

and taking the sum $\sum_{i=B+1}^{T-1} \dot{\eta}_i = \dot{\pi}_{B+1} - \dot{\pi}_T$ we find:

$$(\dot{\pi}_T - \dot{\pi}_{B+1}) + (1-\lambda^m)(\lambda + R_{mp})^n = 0. \tag{18}$$

Equations (15-18) for $(\dot{\pi}_{B+1}, \dot{\pi}_T)$ are equivalent to finding an element $(x, y) \in \mathbb{R}^2$ for which $0 \leq y \leq x \leq 1$ and $\dot{f}(x,y) = \dot{g}(x,y) = \dot{h}(x,y) = 0$, with:

$$\begin{aligned}
\dot{f}(x,y) &= (1-\lambda)\lambda^m \dot{u}(x,y)^n - y(1 + r(1-x)) \\
\dot{g}(x,y) &= (x-1) + (1-\lambda)\sum_{i=0}^{n-1} \dot{u}(x,y)^i \\
\dot{h}(x,y) &= (y-x) + (1-\lambda^m)\dot{u}(x,y)^n.
\end{aligned}$$

The proof now proceeds by first showing that $\dot{h}(x,y) = \dot{f}(x,y) = 0$ implies that $y = \dot{w}(x)$. Next we argue that $0 \leq \dot{w}(x) \leq x$ for $x \in (0,1)$ and the proof completes by showing that $\dot{g}(x, \dot{w}(x))$ has a unique root in $(0,1)$.

From $\dot{h}(x,y) = 0$ we find $\dot{u}(x,y)^n = \frac{x-y}{1-\lambda^m}$. Plugging this into $\dot{f}(x,y) = 0$ shows that we must have $y = \dot{w}(x)$. Taking the derivative of $\dot{w}(x)$, we find:

$$\frac{\partial \dot{w}(x)}{\partial x} = \frac{\lambda^m(1-\lambda)(1-\lambda^{m+1} + r(1-\lambda^m))}{(\lambda^m(rx - \lambda - r) - rx + r + 1)^2} > 0.$$

Note that this means that for $x \in (0,1)$ we have $0 \leq \dot{w}(x)$ as $\dot{w}(0) = 0$. We now show that $\dot{w}(x) \leq x$ for $x \in (0,1)$. The inequality $\dot{w}(x) \leq x$ can be restated as:

$$\lambda^m(1 + r - rx) \leq 1 + r - rx$$

which clearly holds for $0 < \lambda < 1$.

The fact that $\partial \dot{w}(x)/\partial x > 0$ on $(0,1)$ yields

$$\frac{\partial \dot{u}(x, \dot{w}(x))}{\partial x} = \underbrace{\frac{\partial \dot{u}}{\partial x}(x, \dot{w}(x))}_{= \frac{\lambda \dot{w}(x)}{(1-x)^2}} + \underbrace{\frac{\partial \dot{u}}{\partial y}(x, \dot{w}(x))}_{= \frac{\lambda}{1-x} + r} \cdot \frac{\partial \dot{w}(x)}{\partial x} > 0.$$

This implies that $\partial \dot{g}(x, \dot{w}(x))/\partial x > 0$. One can easily verify that $\dot{g}(0, \dot{w}(0)) = -\lambda^n < 0$ and $\dot{g}(\lambda^n, \dot{w}(\lambda^n)) \geq 0$ (as $\dot{u}(\lambda^n, \dot{w}(\lambda^n)) \geq \dot{u}(0, \dot{w}(0)) = \dot{u}(0,0) = \lambda$). Hence there exists a unique x in $(0, \lambda^{B+1})$ for which $\dot{g}(x) = 0$. Thus $\dot{\pi}_{B+1}$ must be equal to this unique root and $\dot{\pi}_T = \dot{w}(\dot{\pi}_{B+1}) \leq \dot{\pi}_{B+1}$. □

Theorem 4. *The set of ODEs given by (13-14) has a unique fixed point $\hat{\pi} = (\hat{\pi}_0, \ldots, \hat{\pi}_T) \in F$. Let $\hat{\eta}_i := \hat{\pi}_i - \hat{\pi}_{i+1}$, then:*

$$\hat{\eta}_i = (1-\lambda)(\lambda + R_{mp})^i \qquad\qquad 0 \leq i \leq B.$$

Moreover $\hat{\pi}_T$ is the unique root on $(0, \lambda^T)$ of the ascending function:

$$\hat{f}(x) := (x-1) + (1-\lambda)\sum_{i=0}^{B} \hat{u}(x)^i,$$

with $\hat{u}(x) := \lambda + \frac{\lambda x}{1-x} + rx = \frac{\lambda}{1-x} + rx$.

Proof. Let $\hat{\pi}$ be a fix point of (13-14), using $\sum_{i=1}^{T} d\hat{\pi}_i/dt = 0$ we find that $\hat{\pi}_1 = \lambda$ and thus $\hat{\eta}_0 = 1-\lambda$. The expressions for $\hat{\eta}_i$ easily follow from (13).

We now verify that the given set of equations has a unique solution that satisfies (14). By definition $1 - \hat{\pi}_T = \sum_{i=0}^{B} \hat{\eta}_i$, which yields the relation:

$$(\hat{\pi}_T - 1) + (1-\lambda)\sum_{i=0}^{B}(\lambda + R_{mp})^i = 0.$$

Due to (15) the above equation corresponds to having $\hat{f}(x) = 0$. We now show that $\hat{f}$ is ascending and has exactly one root on $(0, \lambda^T)$. For ease of notation we let $n = B + 1 = T$. It is easy to check that $\hat{f}(0) = -\lambda^n < 0$ and $\hat{f}(\lambda^n) \geq 0$ (as $\hat{u}(\lambda^n) \geq \lambda$). Further $d\hat{u}(x)/dx = \frac{\lambda}{(x-1)^2} + r > 0$ on $(0,1)$, which shows that $d\hat{f}(x)/dx > 0$.

We end by checking that the the unique root of $\hat{f}(x)$ satisfies equation (14). This equation can be rewritten as $\hat{g}(\hat{\pi}_T) = 0$ with $\hat{g}(x) = (1-\lambda)\hat{u}(x)^n + rx^2 - (1+r)x$, as

$$\begin{aligned} 0 &= \lambda\hat{\eta}_{T-1} - \hat{\pi}_T(1 + r(1 - \hat{\pi}_{T-1} + \hat{\pi}_T - \hat{\pi}_T)) + \frac{\lambda\hat{\pi}_T}{1-\hat{\pi}_T}\hat{\eta}_{T-1} \\ &= \left(\lambda + r\hat{\pi}_T + \frac{\lambda\hat{\pi}_T}{1-\hat{\pi}_T}\right)\hat{\eta}_{T-1} + r\hat{\pi}_T^2 - (1+r)\hat{\pi}_T. \end{aligned}$$

The fact that $\hat{g}(\hat{\pi}_T) = 0$ now follows from:

$$\begin{aligned} (1-\hat{u}(x))\hat{f}(x) &= (1-\hat{u}(x))(x-1) + (1-\lambda)(1-\hat{u}(x)^n) \\ &= -(1-\lambda)\hat{u}(x)^n + \underbrace{(1-\hat{u}(x))(x-1) + 1 - \lambda}_{=-rx^2+(1+r)x} \\ &= -\hat{g}(x), \end{aligned}$$

which completes the proof. □

From the formulation of the max-push strategy it is clear that there is a requirement on R, for the strategy to be well-defined. If R is too low, not all new arrivals at a queue of length T can be migrated. If R is too high, queues with length T will be exhausted and we face the same problem as before.

A valid parameter set can be determined as follows: Let $\Gamma(B, T, R, \lambda)$ be the value of π_{T+1} as calculated by (5-7) with $r\pi_{T+1}$ replaced by R. Now we discern two cases to set T given $T - B$:

- *If* $T - B > 1$, then for a given B and λ, T must be chosen such that $\Gamma(B, T-1, R, \lambda) > 0$ and $\Gamma(B, T, R, \lambda) < 0$.
- *If* $T - B = 1$, then for a given λ, T must be chosen such that $\Gamma(B, B+1, R, \lambda) < 0$, and $\Gamma(B-1, B, R, \lambda) > 0$.

We can now express the main performance measures of the max-push strategy via Theorems 3 and 4:

Theorem 5. *The mean delay D of a job under the max-push strategy with $T \geq B+1$, equals*

$$D_{mp} = \frac{1}{1-\lambda}\left(1 - \frac{M_{mp}}{\lambda}\delta\right),$$

with

$$\delta = T - B - 1 + \frac{\lambda \breve{\pi}_T}{M_{mp}} + \beta \qquad\qquad \beta = \frac{\sum_{i=0}^{B-1}(B-i)\breve{\eta}_i}{1 - \breve{\pi}_{B+1}}$$

Proof. Here, $\breve{\pi}$ and $\breve{\eta}$ is used to denote $\dot{\pi}$ and $\dot{\eta}$ or $\hat{\pi}$ and $\hat{\eta}$, in case $T > B+1$ or $T = B+1$ respectively. Also M_{mp} is to be substituted with $M_{mp|T>B+1}$ or $M_{mp|T=B+1}$, depending on the values for T and B.

The reasoning is the same as in Theorem 2. Migrating tasks skip on average δ places in the queue.

All tasks skip $T - B - 1$ places by construction of the strategy. The fraction of migrating arrivals at a queue of length T skips one extra place ($\lambda\breve{\pi}_T / M_{mp}$). Tasks can skip more places depending on the length of the queue receiving the task, accounting for β places on average. □

Figures 10 and 11 show the mean delay of the max-push strategy, for $T > B+1$ and $T = B+1$, respectively. The max-push connects the points where the push can no longer match R. Connected points all use the same value for parameter B. The values for r_{mp} are shown in Figure 12 for $T = B+1$.

Conjecture 2. *The optimal choice for a rate-based push strategy in class $\mathcal{S}(\mathbf{r}, A)$ is a max-push strategy with $T = B+1$, with T chosen depending on the load λ as outlined in the text preceding Theorem 5.*

Intuitively, it appears desirable to let the longer queues spend as much of the probe budget as possible. The choice of $T = B+1$ indicates that a task is transferred if the transfer results in a lower mean delay without further constraints on how much this gain should be.

3.5. Numerical Validation of Max-Push

We compare the predictions of the infinite system model with respect to a finite system using the max-push strategy with $B \geq 1$ in this section. The setting $B = 0$ was already discussed in [19]. The experimental setup is the same as in Section 3.2, we choose $R_{mp} = 1$ for all experiments and determined r_{mp} using (15).

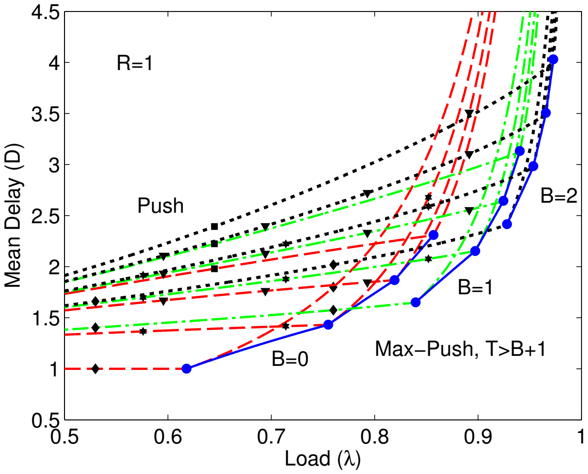

Figure 10: Mean delay of the max-push strategy, with $T > B + 1$ and for $B = 0, 1, 2$, using $R = 1$. For comparison the mean delay of the fixed rate push strategy is also shown for $B = 0$ (dashed), $B = 1$ (dot-dashed) and $B = 2$ (dotted). The markers indicate the value for T, with $T = B + 1$ represented by diamonds, $T = B + 2$ by stars, $T = B + 3$ by triangles and $T = B + 4$ by squares.

In Table 3 we show the relative error of the mean delay observed in the finite system, compared to the infinite system model. The error decreases as the system grows larger, and is smaller for lower loads. Overall, the mean delay is accurately predicted with a relative error of at most a few percent as the system size reaches 50 nodes. The infinite system model is optimistic, predicting a lower mean delay than observed in a finite system.

The relative error of the overall probe rate is shown in Table 4. In all cases the finite system uses more probes than the requested overall probe rate R. Again the error decreases as the system grows in size. However, for high loads and a small system size we observe that the observed overall probe rate is much larger than requested, with a relative error as high as 2.69. This is due to the fact that in a small system there is a higher probability that there will be some periods that all nodes have B or more tasks. If that happens, a new arrival at a queue with length T can not find an instantaneous transfer target, but will spend many probes trying. In the infinite system model this is never a problem, but in a finite system it does occur. In our simulation we allow N probes (without replacement) for such a task, so all queues have been sampled. And if no eligible migration target is found, the queue where the task originally arrived still accepts the task. As the system becomes larger this situation occurs less frequently or not at all.

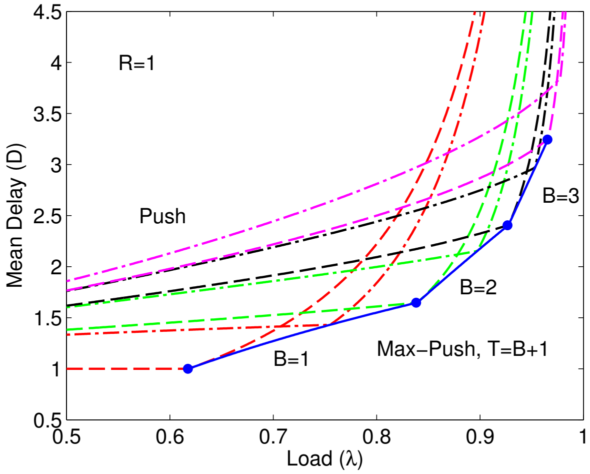

Figure 11: Mean delay of the max-push strategy (full lines), with $T = B + 1$ for $B = 1, 2$, using $R = 1$. For comparison the mean delay of the fixed rate push strategy is also shown for $T = B + 1$ (dashed) and $T = B + 2$ (dot-dashed).

4. Traditional Strategies

In this section we analyze the traditional strategies, where probes are not sent periodically but only on task arrival or completion instants. Probes are sent sequentially until an eligible target for migration is found, or the maximum of L_p probes is reached.

We also show that fixed rate strategies as discussed in Section 3 can be constructed that use the same overall probe rate and result in the same stationary queue length distribution as the traditional strategies.

4.1. Traditional Push

In the traditional push variant, up to L_p probes are sent when a new task arrives at a queue with length at least T. The task is migrated to the first node discovered that has at most B tasks. A similar setup was studied in [14] using birth-death models, with the constraint that $T = B + 1$.

The evolution of the traditional push strategy is modeled by a set of ODEs denoted as $dx(t)/dt = J(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\begin{aligned}\frac{dx_i}{dt} &= \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)) \\ &\quad + \lambda x_T(t)(1 - x_{B+1}(t)^{L_p})\frac{x_{i-1}(t) - x_i(t)}{1 - x_{B+1}(t)},\end{aligned} \tag{19}$$

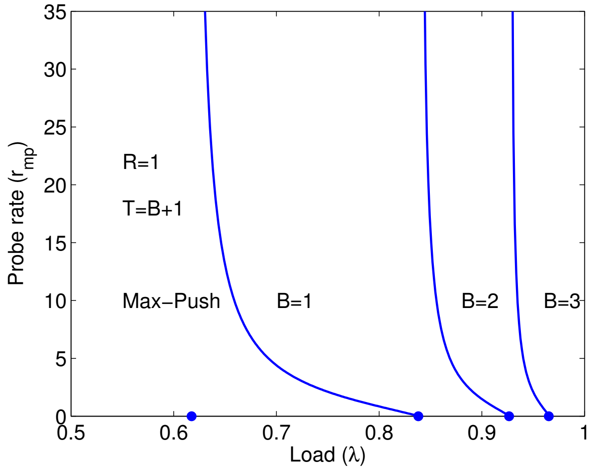

Figure 12: The individual probe rate (r_{mp}) for the max-push strategy with $T = B + 1$ and $R = 1$.

for $1 \leq i \leq B + 1$. For $B + 2 \leq i \leq T$ we have

$$\frac{dx_i}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{20}$$

and for $i > T$ we have

$$\frac{dx_i}{dt} = \lambda(x_{i-1}(t) - x_i(t))x_{B+1}(t)^{L_p} - (x_i(t) - x_{i+1}(t)). \tag{21}$$

An arrival at a queue with length at least T is not transferred if no lightly loaded node is found with L_p probes, this occurs with probability $x_{B+1}(t)^{L_p}$. So with probability $1 - x_{B+1}(t)^{L_p}$ a new arrival at a queue with length at least T (occurring at rate $\lambda x_T(t)$) is migrated to a lightly loaded node. Since each server has the same probability of being probed, the migrating tasks are distributed uniformly over the lightly loaded nodes $((x_{i-1}(t) - x_i(t))/(1 - x_{B+1}(t)))$.

Assume for now that the set of ODEs $J(x(t))$ has a unique fixed point $\tilde{\pi}$. We further assume probes are sent sequentially, and a task is migrated to the first discovered eligible node. So at least one probe is sent, and another probe follows if all previous probes failed to locate a lightly loaded node. This results in an average of $1 + \sum_{i=1}^{L_p - 1} \pi_{B+1}^i$ probes sent. Since probes are sent for each arrival (with rate λ) at a queue of length T or more (π_T), the resulting overall probe rate equals

$$R_{trad.push} = \lambda \tilde{\pi}_T \frac{1 - \tilde{\pi}_{B+1}^{L_p}}{1 - \tilde{\pi}_{B+1}}. \tag{22}$$

Having expressed the overall probe rate, the fixed point structure is given in the next theorem.

N	$B=1$				$B=2$	
	$T=2$	$T=3$	$T=4$	$T=5$	$T=3$	$T=4$
	$\lambda=0.80$	$\lambda=0.875$	$\lambda=0.915$	$\lambda=0.935$	$\lambda=0.915$	$\lambda=0.95$
25	2.07e-2	4.10e-2	6.19e-2	7.94e-2	6.25e-2	1.17e-1
50	7.93e-3	1.44e-2	2.03e-2	2.66e-2	2.22e-2	4.16e-2
100	3.70e-3	6.18e-3	7.44e-3	8.85e-3	9.04e-3	1.47e-2
200	1.81e-3	2.92e-3	3.26e-3	3.60e-3	4.14e-3	5.88e-3
400	9.20e-4	1.47e-4	1.60e-3	1.74e-3	2.06e-3	2.64e-3
800	4.25e-4	7.25e-4	7.61e-4	8.42e-4	1.04e-3	1.25e-3
1600	2.15e-4	3.74e-4	4.11e-4	4.35e-4	5.10e-4	6.26e-4

Table 3: The relative error of the mean delay D in a finite system with size N using the max-push strategy, compared to the infinite system model. We note that the infinite system model is optimistic with respect to the performance of the finite system.

N	$B=1$				$B=2$	
	$T=2$	$T=3$	$T=4$	$T=5$	$T=3$	$T=4$
	$\lambda=0.80$	$\lambda=0.875$	$\lambda=0.915$	$\lambda=0.935$	$\lambda=0.915$	$\lambda=0.95$
25	4.20e-1	8.17e-1	1.25e+0	1.53e+0	1.56e+0	2.69e+0
50	1.45e-1	3.67e-1	7.48e-1	1.06e+0	7.09e-1	2.00e+0
100	5.23e-2	1.16e-1	2.78e-1	4.84e-1	1.79e-1	8.75e-1
200	2.37e-2	4.58e-2	8.59e-2	1.46e-1	5.60e-2	2.04e-1
400	1.15e-2	2.14e-2	3.60e-2	5.22e-2	2.55e-2	5.73e-2
800	5.51e-3	1.04e-2	1.69e-2	2.38e-2	1.23e-2	2.52e-2
1600	2.78e-3	5.18e-3	8.45e-3	1.15e-2	6.01e-3	1.21e-2

Table 4: The relative error of the overall probe rate R in a finite system with size N using the max-push strategy, compared to the infinite system model. We note that when using the r_{mp} as derived from the infinite system model, the finite system produces a higher overall probe rate than requested.

Theorem 6. *The set of ODEs given by (19-21) has a unique fixed point $\tilde{\pi} = (\tilde{\pi}_0, \tilde{\pi}_1, \dots) \in E$. Let $\tilde{\eta}_i := \tilde{\pi}_i - \tilde{\pi}_{i+1}$, then we have the relations:*

$$\tilde{\eta}_i = (1-\lambda)(\lambda + R_{trad.push})^i, \qquad 0 \le i \le B+1 \tag{23}$$

$$\tilde{\eta}_i = \tilde{\eta}_{B+1}\lambda^{i-(B+1)}, \qquad B+2 \le i \le T \tag{24}$$

$$\tilde{\eta}_i = \tilde{\eta}_T(\lambda\tilde{\pi}_{B+1}^{L_p})^{i-T}, \qquad i > T. \tag{25}$$

Moreover $\tilde{\pi}_{B+1}$ is the unique root of the ascending function on $(0, \lambda^{B+1})$:

$$\tilde{f}(x) = (x-1) + (1-\lambda) \cdot \sum_{i=0}^{B} \tilde{u}(x, \tilde{w}(x))^i,$$

with $\tilde{w}(x) = \frac{(1-\lambda)\lambda^{T-B-1}x}{(\lambda^{T-B}-\lambda)x^{L_p} + (1-\lambda^{T-B})}$ and $\tilde{u}(x,y) = \lambda + \lambda y \frac{1-x^{L_p}}{1-x}$. Further, $\tilde{\pi}_T = \tilde{w}(\tilde{\pi}_{B+1})$.

Proof. Let $\tilde{\pi}$ be a fix point of (19-21), we show that (19-21) incur relations on $\tilde{\pi}$ which make it unique. Using $\sum_{i=1}^{\infty} d\tilde{\pi}_i/dt = 0$ we find that $\tilde{\pi}_1 = \lambda$. The relations for $\tilde{\eta}_i$ easily follow from (19-21).

For ease of notation we write $m = T-B-1, n = B+1, l = L_p$. By definition we have $\sum_{i=0}^{B} \tilde{\eta}_i = 1 - \tilde{\pi}_{B+1}$, $\sum_{i=B+1}^{T-1} \tilde{\eta}_i = \tilde{\pi}_{B+1} - \tilde{\pi}_T$ and $\sum_{i=T}^{\infty} \tilde{\eta}_i = \tilde{\pi}_T$. These three equalities combined with (22) can be restated as $\tilde{f}(\tilde{\pi}_{B+1}, \tilde{\pi}_T) = \tilde{g}(\tilde{\pi}_{B+1}, \tilde{\pi}_T) = \tilde{h}(\tilde{\pi}_{B+1}, \tilde{\pi}_T) = 0$, with

$$\tilde{f}(x,y) = (x-1) + (1-\lambda)\sum_{i=0}^{n-1} \tilde{u}(x,y)^i \tag{26}$$

$$\tilde{g}(x,y) = (y-x) + (1-\lambda^m)\tilde{u}(x,y)^n \tag{27}$$

$$\tilde{h}(x,y) = -y + (1-\lambda)\frac{\lambda^m}{1-\lambda x^l}\tilde{u}(x,y)^n. \tag{28}$$

From the equation $\tilde{g}(x,y) = 0$ we can infer:

$$\tilde{u}(x,y)^n = \frac{x-y}{1-\lambda^m}.$$

Plugging this equality into $\tilde{h}(x,y) = 0$, we find that $y = \tilde{w}(x)$ must hold. We now note that

$$\frac{\partial \tilde{w}(x)}{\partial x} = \frac{(1-\lambda)\lambda^m(\lambda(l-1)(1-\lambda^m)x^l + (1-\lambda^{m+1}))}{(\lambda^{m+1}x^l - \lambda^{m+1} - \lambda x^l + 1)^2} > 0.$$

This indicates that $\tilde{w}(x) \geq 0$ for $x \in (0,1)$ as $\tilde{w}(0) = 0$. We also need to verify that $\tilde{w}(x) \leq x$, which is equivalent to

$$(1-\lambda^m)(\lambda x^l - 1) \leq 0,$$

which holds trivially. We further note that

$$\frac{\partial \tilde{u}(x, \tilde{w}(x))}{\partial x} = \frac{\partial \tilde{u}}{\partial x}(x, \tilde{w}(x)) + \frac{\partial \tilde{u}}{\partial y}(x, \tilde{w}(x))\frac{\partial \tilde{w}(x)}{\partial x} > 0,$$

which means that $\partial \tilde{f}(x, \tilde{w}(x))/\partial x > 0$. This suffices to prove the uniqueness of the fixed point. Moreover the existence follows by remarking that $\tilde{f}(0, \tilde{w}(0)) = -\lambda^n < 0$ and $f(\lambda^n, \tilde{w}(\lambda^n)) \geq 0$. □

Instead of providing an explicit formula for the mean delay, we show the following equivalence.

Theorem 7. *When using the same parameters B and T, and matching the $R_{trad.push}$ generated by the traditional push, the fixed rate push strategy has the same fixed point, resulting in an equivalent performance.*

Proof. From (5-7) and (23-25), it is clear that η_i and $\tilde{\eta}_i$ are identical for $i \leq T$ as $R_{trad.push} = r\pi_{T+1}$. What remains to be shown is that

$$\lambda\tilde{\pi}_{B+1}^{L_p} = \frac{\lambda}{1 + r_{push}(1-\pi_{B+1})},$$

which follows by noting that both the vectors η_i and $\tilde{\eta}_i$ sum to one. □

4.2. Traditional Pull

In the traditional pull variant, whenever a node with queue length at most $B+1$ has processed a task, it sends out at most L_p probes to locate a highly loaded node. The first node found with a queue length larger than T, migrates a task to the probing node. A similar setup was studied in [10] using birth-death models, with the constraint that $T = B + 1$.

The evolution of the traditional pull strategy is modeled by a set of ODEs denoted as $dx(t)/dt = K(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\frac{dx_i}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t))(1 - x_{T+1}(t))^{L_p}, \tag{29}$$

for $1 \leq i \leq B+1$. For $B+2 \leq i \leq T$ we have

$$\frac{dx_i}{dt} = \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)), \tag{30}$$

and for $i > T$ we have

$$\begin{aligned}\frac{dx_i}{dt} &= \lambda(x_{i-1}(t) - x_i(t)) - (x_i(t) - x_{i+1}(t)) \\ &- (x_1(t) - x_{B+2}(t))(1 - (1 - x_{T+1}(t))^{L_p})\frac{x_i(t) - x_{i+1}(t)}{x_{T+1}(t)}.\end{aligned} \tag{31}$$

The queue length of nodes with at most $B+1$ tasks only decreases if they fail to find a long queue to migrate a task from, this happens with probability $(1-x_{T+1})^{L_p}$. The extra negative term in (31) indicates migrations to the lightly loaded nodes. For every completion of a queue with length at most $B+1$ (rate $(x_1 - x_{B+2})$), the probes are successful with probability $(1 - (1 - x_{T+1})^{L_p})$, and the probability for discovery of a long queues with length i is uniformly distributed over all long queues $((x_i(t) - x_{i+1}(t))/(x_{T+1}(t)))$.

The set of ODEs $K(x(t))$ has a unique fixed point $\mathring{\pi}$. We first express the overall probe rate, and then describe $\mathring{\pi}$ explicitly.

We assume probes are sent sequentially, and a task is migrated from the first discovered eligible node. Thus, at least one probe is sent, and extra probes follow if all previous attempts were unsuccessful. This results in an average of $1 + \sum_{i=1}^{L_p-1}(1 - \mathring{\pi}_{T+1})^i$ probes sent. Since probes are sent for each completion at a queue with a length of at most $B+1$, the resulting overall probe rate equals:

$$R_{trad.pull} = (\mathring{\pi}_1 - \mathring{\pi}_{B+2})\frac{1 - (1 - \mathring{\pi}_{T+1})^{L_p}}{\mathring{\pi}_{T+1}} \tag{32}$$

Having expressed the overall probe rate, the fixed point is given in the next theorem.

Theorem 8. *The set of ODEs given by (29-31) has a unique fixed point $\mathring{\pi} = (\mathring{\pi}_0, \mathring{\pi}_1, \dots) \in E$. Let $\mathring{\eta}_i := \mathring{\pi}_i - \mathring{\pi}_{i+1}$, then we have the relations:*

$$\mathring{\eta}_i = (1-\lambda)\left(\frac{\lambda}{(1-\mathring{\pi}_{T+1})^{L_p}}\right)^i, \qquad 0 \le i \le B+1 \tag{33}$$

$$\mathring{\eta}_i = \mathring{\eta}_{B+1}\lambda^{i-(B+1)} \qquad B+2 \le i \le T \tag{34}$$

$$\mathring{\eta}_i = \mathring{\eta}_T\left(\frac{\lambda}{1+R_{trad.pull}}\right)^{i-T}, \qquad i > T. \tag{35}$$

Moreover, the value of $\mathring{\pi}_{T+1}$ is found as the unique root of the ascending function:

$$\mathring{f}(x) = (x-1) + (1-\lambda)\sum_{i=0}^{B+1}\mathring{u}(x)^i + \lambda(1-\lambda^{T-B-1})\mathring{u}(x)^{B+1}$$

on $(0, \lambda^{T+1})$, with $\mathring{u}(x) = \frac{\lambda}{(1-x)^{L_p}}$.

Proof. The expressions for $\mathring{\eta}_i$ readily follow from (29-31). To prove the uniqueness of $\mathring{\pi}_{T+1}$ we use:

$$\begin{aligned}
1 - \mathring{\pi}_{T+1} &= \sum_{i=0}^{B+1}\mathring{\eta}_i + \sum_{i=B+2}^{T}\mathring{\eta}_i \\
&= (1-\lambda)\sum_{i=0}^{B+1}\left(\frac{\lambda}{(1-\mathring{\pi}_{T+1})^{L_p}}\right)^i + \mathring{\eta}_{B+1}\sum_{i=B+2}^{T}\lambda^{i-(B+1)} \\
&= (1-\lambda)\sum_{i=0}^{B+1}u(\mathring{\pi}_{T+1})^i + \lambda(1-\lambda^{T-B-1})u(\mathring{\pi}_{T+1})^{B+1}.
\end{aligned}$$

Hence $\mathring{\pi}_{T+1}$ is a root of $\mathring{f}(x)$. Further, $\mathring{f}(0) = -\lambda^{T+1} < 0$, $\mathring{f}(\lambda^{T+1}) \ge 0$ and $d\mathring{f}(x)/dx > 0$ on $(0,1)$ as $d\mathring{u}(x)/dx > 0$ on $(0,1)$. □

Instead of providing an explicit formula for the mean delay, we show the following equivalence.

Theorem 9. *When using the same parameters B and T, and matching the $R_{trad.pull}$ generated by the traditional pull, the fixed rate pull strategy has the same fixed point distribution, resulting in an equivalent performance.*

Proof. From (5-7) and (33-35), it is clear that η_i and $\mathring{\eta}_i$ are identical iff

$$\frac{\lambda}{(1-\mathring{\pi}_{T+1})^{L_p}} = \lambda + r_{pull}\pi_{T+1}, \tag{36}$$

as $R_{trad.pull} = r(1-\pi_{B+1})$. This follows from noting that both the vectors η_i and $\mathring{\eta}_i$ sum to one.

□

5. d-Choices Strategies

In this section we study variants of the d-choices strategy. The original strategy was introduced in [5], where an infinite system model was used to describe its behavior. Let $x(t) = (x_1(t), x_2(t), \ldots)$, where $x_i(t)$ represents the fraction of nodes with at least i jobs at time t. Then the evolution of queue lengths under the d-choices strategy is formulated as the following set of ODEs denoted as $dx(t)/dt = L(x(t))$:

$$\frac{dx_i(t)}{dt} = \lambda(x_{i-1}(t)^d - x_i(t)^d) - (x_i(t) - x_{i+1}(t)). \tag{37}$$

Results in [5] show that all trajectories converge to a unique fixed point

$$\bar{\pi}_i = \lambda^{\frac{d^i-1}{d-1}}. \tag{38}$$

As explained further on an equivalent distributed variant requires fewer than d probes per task. Additionally, we construct equivalent rate-based variants that send either single probes or batches of probes periodically instead of on task arrival instants.

5.1. Distributed d-Choices

The original d-choices as introduced in [5] assumes that a central dispatcher sends d probes for every task arrival. When assuming a central dispatcher, other approaches are known to perform better with less probes [4]. We assume that tasks originate at the nodes themselves.

In a sense this setup provides the information of exactly one probe message, that is the queue length of the queue where the task arrives. Therefore, an equivalent strategy to a central dispatcher sending d probes is to let the nodes send $d-1$ probes on a task arrival instant. The task is then forwarded to the least loaded probed node, or stays at the originating node if no shorter queue is found.

The evolution of the distributed d-choices strategy is given by a set of ODEs denoted as $dx(t)/dt = M(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\begin{aligned}\frac{dx_i(t)}{dt} =& \lambda(x_{i-1}(t) - x_i(t))x_{i-1}(t)^{d-1} - (x_i(t) - x_{i+1}(t)) \\ &+ \lambda x_i(t)(x_{i-1}(t)^{d-1} - x_i(t)^{d-1}),\end{aligned} \tag{39}$$

for $i > 0$, with $x_0(t) = 1$. Queues of length i are created by arrivals in a queue with length $i-1$ ($\lambda(x_{i-1}(t) - x_i(t))$), only if $d-1$ probes could not find a shorter queue (probability $x_{i-1}^{d-1}(t)$). Additionally, queues of length i are created if an arrival at a queue with length at least i ($\lambda x_i(t)$), sends $d-1$ probes and finds a queue with length $i-1$ the shortest (probability $x_{i-1}^{d-1}(t) - x_i^{d-1}(t)$)

Algebraic manipulation on (39) immediately shows the equivalence with the original formulation of the d-choices strategy in (37).

Using fixed point from (38) we can formulate the mean delay in terms of migrations in the next theorem.

Theorem 10. *The mean delay of both the distributed and centralized d-choices strategy can be formulated as*

$$\frac{1}{1-\lambda}(1-\frac{\alpha}{\lambda}),$$

with

$$\alpha = \lambda \sum_{i=1}^{\infty} (\bar{\pi}_i - \bar{\pi}_{i+1}) \sum_{j=0}^{i-1} (\bar{\pi}_j^{d-1} - \bar{\pi}_{j+1}^{d-1})(i-j).$$

Proof. The improvement over the mean delay of an M/M/1 queue can be formulated as the average number of places a task will skip in the queue due to a migration. Here, for every arrival (λ) at a queue of length i ($\bar{\pi}_i - \bar{\pi}_{i+1}$), the $d-1$ probes could find a shorter queue. The shortest queue found is of length j with probability ($\bar{\pi}_j^{d-1} - \bar{\pi}_{j+1}^{d-1}$), in which case the task skips $(i-j)$ places. □

Although there is an infinite sum in α of the above theorem, the terms quickly become small as $\bar{\pi}$ decreases doubly exponentially.

We note that the required overall request rate of the distributed d-choices can be lower than $\lambda(d-1)$. First, if a task originates at an empty server, no probes need to be send as no shorter queue can be found. Similarly, the $d-1$ probes could be sent sequentially and stop once an empty server is found. Thus only servers with at least one job need to send probes at task arrival instants until either an empty server is found or the maximum of $d-1$ probes is reached. Analytically, this results in an overall probe rate of

$$R_{dChoices} = \bar{\pi}_1 \lambda \left(1 + \sum_{i=1}^{d-2} \bar{\pi}_1^i \right),$$

where $\bar{\pi}_1 \lambda$ is the rate of probe events (arrivals at busy servers), and $(1+\sum_{i=1}^{d-1} \bar{\pi}_1)$ is the number of probes per event. At least one probe is sent, and a next probe follows if all previous probes found busy servers, up to a maximum of $d-1$ probes in total. Since $\bar{\pi}_1$ equals λ we can simplify the expression to

$$R_{dChoices} = \frac{\lambda^2(1-\lambda^{d-1})}{1-\lambda}. \tag{40}$$

We will match this probe rate in the following sections to create equivalent strategies.

5.2. Rate-based Variant Sending Probes in Batch

Instead of sending out $d-1$ probes at task arrival instants, we can adapt the strategy to send batches of probes according to a Poisson process with rate r. We will call the sending of a batch of probes a probe event. It is our aim to find a strategy equivalent to the d-choices strategy, i.e. one that achieves the same stationary distribution when using the same overall probe rate.

The first attempt at finding such a strategy lets queues with two or more jobs send out batches of probes periodically with a rate r that is independent of the queue's length. The evolution of such a strategy is modeled by the set of ODEs denoted as $dx(t)/dt = N(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\frac{dx_1}{dt} = \lambda(1 - x_1(t)) + rx_2(t)(1 - x_1(t)^{d-1}) - (x_1(t) - x_2(t)), \tag{41}$$

and for $i \geq 2$ we have

$$\begin{aligned}\frac{dx_i}{dt} =& \lambda(x_{i-1}(t) - x_i(t)) + rx_{i+1}(t)(x_{i-1}(t)^{d-1} - x_i(t)^{d-1}) \\ & - (x_i(t) - x_{i+1}(t))(1 + r(1 - x_{i-1}(t)^{d-1})).\end{aligned} \tag{42}$$

Queues with length one are created by new arrivals and probes to an empty server. Tasks from all queues with tasks waiting ($x_2(t)$) are eligible for transfer to an empty server, and those queues generate probe events with rate r. An empty server is located by $d-1$ probes with probability $(1 - x_1(t)^{d-1})$. In general, queues of length i are created when a probe event of a queue with length at least $i+1$ identifies a queue with length $i-1$ as shortest among the $d-1$ probed servers. Likewise, the fraction of queues with length i decreases if the probe events (which occur at rate r) locate a queue with length lower than $i-1$ (with probability $(1 - x_{i-1}(t)^{d-1})$).

From (40) we note that the rate of probe events must be λ^2, as you send $(1-\lambda^{d-1})/(1-\lambda)$ probes on average per event. In the system above, all servers with tasks waiting generate probe events at the same rate. Therefore, in order for the system to be equivalent with d-choices, we have the condition $r\bar{\pi}_2 = \lambda^2$. In other words, r would need to be $1/\lambda^{d-1}$. Unfortunately, when using this r in conditions (42) and setting $dx_i(t)/dt = 0$, $\bar{\pi}$ is not a solution to the resulting set of equations. In other words, it is impossible to create such a strategy that has the same fixed point as the d-choices strategy.

However, when we let each queue send at a rate r_i depending on its length i, we can find a strategy equivalent with d-choices by choosing r_i appropriately. We call this strategy push-d-batch. The evolution of such a strategy is modeled by the set of ODEs denoted as $dx(t)/dt = P(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. As explained below, this set of ODEs can be written as

$$\begin{aligned}\frac{dx_1}{dt} =&\lambda(1-x_1(t)) - (x_1(t)-x_2(t)) \\ &+ (1-x_1(t)^{d-1})\sum_{j=2}^{\infty} r_j(x_j(t)-x_{j+1}(t)),\end{aligned} \tag{43}$$

and for $i \geq 2$ we have

$$\begin{aligned}\frac{dx_i}{dt} =&\lambda(x_{i-1}(t)-x_i(t)) \\ &- (x_i(t)-x_{i+1}(t))(1+r_i(1-x_{i-1}(t)^{d-1})) \\ &+ (x_{i-1}(t)^{d-1}-x_i(t)^{d-1})\sum_{j=i+1}^{\infty} r_j(x_j(t)-x_{j+1}(t)).\end{aligned} \tag{44}$$

The same remarks as for $N(x(t))$ apply. The difference here is that queues with length i generates probe events with rate r_i, so we now have to sum the r_i over the queue lengths: $\sum_{j=i+1}^{\infty} r_j(x_j(t)-x_{j+1}(t))$.

We aim to achieve the same stationary distribution as d-choice, so we will use $\bar{\pi}$ from (38) to denote the fixed point. When substituting x_i with $\bar{\pi}_i$ in (43), the expression reduces to zero as required. We also aim to use the same rate of probe events, therefore

$$\sum_{j=2}^{\infty} r_j(\bar{\pi}_j(t)-\bar{\pi}_{j+1}(t)) = \lambda^2.$$

Achieving both objectives is accomplished by the choice of r_i. As we know the fixed point of the d-choices strategy ($\bar{\pi}$), we can find r_i from dx_i/dt in (44) by rewriting the sum term as the known total sum (λ^2) minus the missing terms. For example we find r_2 from

$$\begin{aligned}\frac{dx_2}{dt} = 0 =&\lambda(\bar{\pi}_1-\bar{\pi}_2) - (\bar{\pi}_2-\bar{\pi}_3)(1+r_2(1-\bar{\pi}_1^{d-1})) \\ &+ (\bar{\pi}_1^{d-1}-\bar{\pi}_2^{d-1})(\lambda^2 - r_2(\bar{\pi}_2-\bar{\pi}_3)),\end{aligned}$$

where all terms are known except r_2. Repeating this procedure for $i \geq 2$ we find the general expression

$$r_{i|batch} = \frac{-\lambda(1-\lambda^{d^i-1})}{(1-\lambda^{d^i})(1-\bar{\pi}_i^{d-1})} - \frac{1-\lambda^{1-d^{i-1}}}{(1-\lambda^{d^i})(1-\bar{\pi}_{i-1}^{d-1})}.$$

By allowing queues to generate probe events at a rate dependent on the queue length, we have shown that a rate-based variant equivalent to d-choices can be constructed for which probe events need not be at task arrival instants. In this formulation probes are still sent in batch, and therefore this strategy is not a member of the class $\mathcal{S}(\mathbf{r}, A)$. In the next section we construct an equivalent rate-based variant where probe events consist of a single probe, thus belonging to the class $\mathcal{S}(\mathbf{r}, A)$.

5.3. Rate-based Variant Sending Single Probes

In the previous section we showed that generating probe events as a Poisson process can be just as effective as sending probes at arrival instants. In this section we demonstrate that sending probes in batch is also not required to achieve the same stationary distribution as d-choice.

Again our aim is to construct a strategy with an equivalent performance compared to d-choice, while using the same number of probes. Now a probe event consists of sending a single probe. A migration is initiated if the probe finds a queue of at least two tasks shorter, so all transfers lower the mean queue length but tasks can be migrated multiple times. Each queue with length i generates probe events at rate r_i, and the overall probe rate is equal to (40). We will call the strategy described here push-d-single.

Formally this strategy is a member of the class $\mathcal{S}(\mathbf{r}, A)$ and is defined as follows. The elements $a_{i,j}$ are one if $i > j+1$. The explicit values for r_i are introduced further on.

The evolution of push-d-single is modeled by the set of ODEs denoted as $dx(t)/dt = Q(x(t))$, where $x(t) = (x_1(t), x_2(t), \ldots)$ and $x_i(t)$ represents the fraction of the number of nodes with at least i jobs at time t. This is a simplified version of Equation (1), and the ODEs can be written as

$$\frac{dx_1}{dt} = \lambda(1 - x_1(t)) - (x_1(t) - x_2(t)) + (1 - x_1(t)) \sum_{j=2}^{\infty} r_j(x_j(t) - x_{j+1}(t)), \quad (45)$$

and for $i \geq 2$ we have

$$\begin{aligned} \frac{dx_i}{dt} =& \lambda(x_{i-1}(t) - x_i(t)) \\ & - (x_i(t) - x_{i+1}(t))(1 + r_i(1 - x_{i-1}(t))) \\ & + (x_{i-1}(t) - x_i(t)) \sum_{j=i+1}^{\infty} r_j(x_j(t) - x_{j+1}(t)). \end{aligned} \quad (46)$$

When substituting $x_i(t)$ with $\bar{\pi}_i$ of (38) in (45) and using

$$\sum_{j=2}^{\infty} r_j(x_j(t) - x_{j+1}(t)) = \frac{\lambda^2(1 - \lambda^{d-1})}{1 - \lambda},$$

the expression reduces to zero as required, indicating that this strategy could have the same fixed point as the d-choices strategy. In order to find a suitable r_i we employ the same method as in the previous section, we rewrite the sum $\sum_{j=i+1}^{\infty} r_j(x_j(t) - x_{j+1}(t))$ as the known total ($R_{dChoices}$) minus the missing terms. Then, we find r_i by substituting $\bar{\pi}$ of (38) in (46) and requiring that $dx_i/dt = 0$. For example r_2 is found from

$$\begin{aligned} \frac{dx_2}{dt} = 0 =& \lambda(\bar{\pi}_1 - \bar{\pi}_2) - (\bar{\pi}_2 - \bar{\pi}_3)(1 + r_2(1 - \bar{\pi}_1)) \\ & + (\bar{\pi}_1 - \bar{\pi}_2)\left(\frac{\lambda^2(1 - \lambda^{d-1})}{1 - \lambda} - r_2(\bar{\pi}_2 - \bar{\pi}_3)\right). \end{aligned}$$

In general, we find that r_i for $i \geq 2$ must be equal to

$$r_{i|single} = \frac{\lambda^{\frac{1}{d-1}} \left(\frac{\left(\lambda^{\frac{d^{i-1}}{d-1}} - \lambda^{\frac{d^i}{d-1}}\right)\left(\lambda^{\frac{d^i}{d-1}} - \lambda^{\frac{d}{d-1}}\right)}{\left(\lambda^{\frac{1}{d-1}} - \lambda^{\frac{d^{i-1}}{d-1}}\right)\left(\lambda^{\frac{d^{i+1}}{d-1}} - \lambda^{\frac{d^i}{d-1}}\right)} - 1 \right)}{\lambda^{\frac{1}{d-1}} - \lambda^{\frac{d^i}{d-1}}},$$

in order for the stationary distribution to match $\bar{\pi}$ while using an overall probe rate of $\sum_{i=2}^{\infty} r_i(\bar{\pi}_i - \bar{\pi}_{i+1}) = R_{dChoices}$.

6. Performance Evaluation

As we know the overall probe rate of the distributed d-choices strategy from (40), we can compare the considered strategies fairly. That is to say, we compare the mean delay given that all strategies use the same overall probe rate. We choose to compare the d-choices strategy due to its popularity, and compare it with the strategies of the class $\mathcal{S}(\mathbf{r}, A)$ that we expect to be optimal as indicated in Conjectures 1 and 2.

We let the d-choices strategy determine the overall probe rate, and make sure the max-push and fixed rate pull strategy match this rate by setting T, B and r appropriately. Figures 13 and 14 summarize the performance comparison.

The fixed rate pull strategy is clearly superior for high loads. Also notable is that its mean delay stays finite as the load λ tends to one, specifically the delay approaches $d/(d-1)$ with $R = R_{dChoices}$. This can be deduced by first observing that the limit

$$\lim_{\lambda \to 1} R_{dChoices} = \lim_{\lambda \to 1} \frac{\lambda^2(1-\lambda^{d-1})}{1-\lambda} = d-1,$$

and using $d-1$ as the value for R in $\lim_{\lambda \to 1} D_{pull}$, with D_{pull} from [18, Theorem 3]

$$D_{Pull} = \frac{1+R}{1-\lambda+R}.$$

Note that the probe rate r becomes infinite in this case, as it is given by $r = R/(1-\lambda)$. For lower loads the pull strategy is not optimal, but adopting this strategy independently of the system load might be an option as the performance is still reasonable and the simplicity of not having to switch strategies depending on the system load keeps the implementation straightforward. Furthermore, the only parameter that would have to be adjusted at runtime depending on the system load is the probe rate, as we conjecture that setting $T = 1$ and $B = 0$ is optimal.

The mean delay of the d-choices and max-push strategy are almost identical for low loads, with the max-push achieving a slightly lower mean delay. This region extends to medium loads as d increases. For higher loads the max-push strategy only slightly outperforms d-choices. This close match in mean delay is notable because we conjecture that max-push is the optimal push strategy within the class $\mathcal{S}(\mathbf{r}, A)$. This suggests that d-choices achieves a close to optimal result with a far simpler approach. The only parameter d-choices has to select is d, whereas the max-push has to adjust B, T and r depending on the system load. Furthermore, the assumption that a node can probe at an infinite rate might not hold, and will in practice be replaced by some high but finite rate. Moreover, in a setting with a finite number of servers it can occur that all queues are temporarily longer than B and thus no transfers can be made, yet new arrivals at queues with length T expect an immediate transfer. In addition, sending a batch of probes might be preferable if the latency is non-negligible in order to avoid waiting for the results of multiple sequential probes. In conclusion, d-choices is far more practical than max-push and still achieves a comparable performance.

To better understand why the performance of the d-choices and max-push strategies is so similar for low to moderate loads, we show in Table 5 several probe rates r_i used by push-d-single. Clearly r_i increases with i and d, but decreases with λ. The increase with i is fast, so that for low loads the push-d-single and the max-push behave almost the same. They both require that queues with length at least i send probes at a practically infinite rate, and that most of the remaining probes are send by the queues with length $i - 1$.

(λ, d)	r_2	r_3	r_4	r_5
$(0.5, 2)$	1.60	7.53	1.28e+2	3.28e+4
$(0.5, 4)$	1.35e+1	3.38e+4	9.22e+18	5.79e+76
$(0.75, 2)$	8.53e−1	1.80	6.81	7.41e+1
$(0.75, 4)$	4.56	9.61e+1	7.45e+7	7.23e+31
$(0.95, 2)$	5.53e−1	6.43e−1	8.61e−1	1.50
$(0.95, 4)$	1.92	3.88	3.59e+1	4.85e+5

Table 5: The first probe rates r_i of push-d-single. We note that r_i increases rapidly with i and d, and decreases with λ.

7. Conclusion

In this paper we have studied several load balancing strategies. We introduced an infinite system model of a general push and pull framework, and indicated the strategies that we expect to be optimal for this class.

We have extended the infinite system model for the fixed rate push and pull, max-push, and the traditional push and pull, to include the parameter B describing the maximal queue length of a lightly loaded server. For a push strategy increasing this B can lead to better performance, whereas for a pull strategy setting $B = 0$ appears best. In addition, we have shown that traditional and fixed rate strategies are equivalent if both use the same overall probe rate R.

Furthermore, we have revisited the popular d-choices strategy and have shown that the required overall probe rate is smaller than d probes per task,

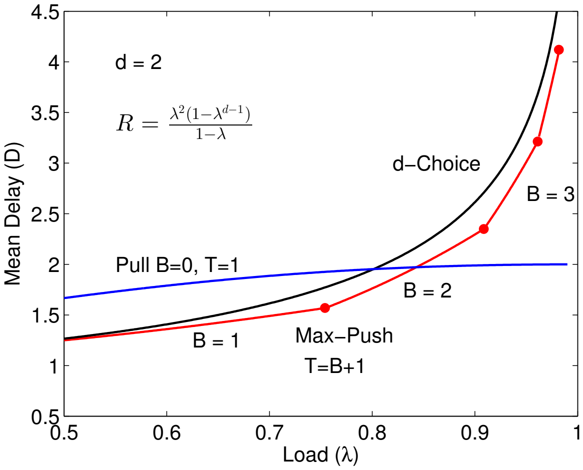

Figure 13: Mean delay of the d-choices with $d = 2$, max-push and pull strategy. All strategies produce the same overall probe rate R.

specifically $\lambda^2(1-\lambda^{d-1})/(1-\lambda)$. In the original formulation probes are sent in batch and on task arrival instants. We have shown that equivalent rate-based push strategies exist that send either single probes or a batch of probes periodically according to a Poisson process with rate r_i dependent on the queue length i.

Finally, we compared the performance of the best performing rate-based push and pull strategy with d-choices, given that the same overall probe rate is used. The pull strategy is the best choice for high loads, but its simplicity and reasonable performance for low to moderate loads makes it a viable solution in case the system must use a single strategy. For low loads the max-push and d-choices performance is nearly equivalent, with the max-push achieving a slightly lower mean delay for medium to high loads. Still, it is remarkable that the simple d-choices strategy performs so close to the more complicated max-push which we conjecture to be an optimal push strategy.

Acknowledgments

We would like to thank the anonymous referees for their many helpful suggestions that improved the quality of this work. This work was partially supported by the FWO grant G024514N.

References

[1] E. Schurman, J. Brutlag, The user and business impact of server delays, additional bytes and http chunking in web search, OReilly Velocity Web performance and operations conference (June 2009).

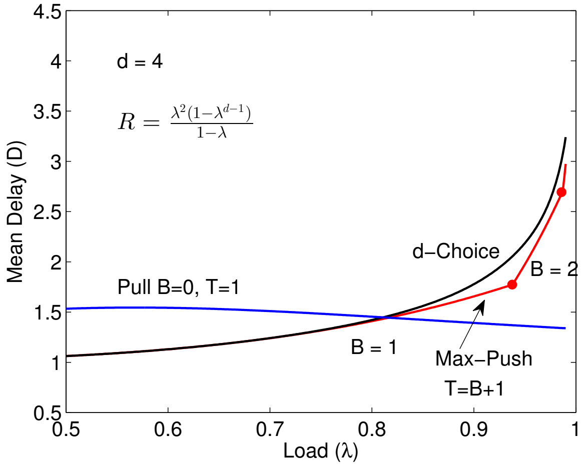

Figure 14: Mean delay of the d-choices with $d = 4$, max-push and pull strategy. All strategies produce the same overall probe rate R.

[2] V. Gupta, M. Harchol-balter, K. Sigman, W. Whitt, Analysis of jointheshortest-queue routing for web server farms, in: In PERFORMANCE 2007. IFIP WG 7.3 International Symposium on Computer Modeling, Measurement and Evaluation, 2007.

[3] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, A. Greenberg, Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services, Perform. Eval. 68 (2011) 1056–1071.

[4] A. Stolyar, Pull-based load distribution in large-scale heterogeneous service systems, Queueing Systems 80 (4) (2015) 341–361. doi:10.1007/s11134-015-9448-8.
URL `http://dx.doi.org/10.1007/s11134-015-9448-8`

[5] M. Mitzenmacher, The power of two choices in randomized load balancing, IEEE Trans. Parallel Distrib. Syst. 12 (2001) 1094–1104.

[6] N. Vvedenskaya, R. Dobrushin, F. Karpelevich, Queueing system with selection of the shortest of two queues: an asymptotic approach, Problemy Peredachi Informatsii 32 (1996) 15–27.

[7] C. Graham, Chaoticity on path space for a queueing network with selection of the shortest queue among several, J. Appl. Probab. 37 (1) (2000) 198–211. doi:10.1239/jap/1014842277.
URL `http://dx.doi.org/10.1239/jap/1014842277`

[8] M. Bramson, Y. Lu, B. Prabhakar, Randomized load balancing with general service time distributions, SIGMETRICS Perform. Eval. Rev. 38 (1) (2010)

275–286. doi:10.1145/1811099.1811071.
URL `http://doi.acm.org/10.1145/1811099.1811071`

[9] L. Ying, R. Srikant, X. Kang, The power of slightly more than one sample in randomized load balancing, in: Proc. of IEEE INFOCOM, 2015.

[10] D. Eager, E. Lazowska, J. Zahorjan, A comparison of receiver-initiated and sender-initiated adaptive load sharing, Perform. Eval. 6 (1) (1986) 53–68. doi:http://dx.doi.org/10.1016/0166-5316(86)90008-8.

[11] R. Mirchandaney, D. Towsley, J. Stankovic, Analysis of the effects of delays on load sharing, IEEE Trans. Comput. 38 (11) (1989) 1513–1525. doi:http://dx.doi.org/10.1109/12.42124.

[12] N. Gast, B. Gaujal, A mean field model of work stealing in large-scale systems, SIGMETRICS Perform. Eval. Rev. 38 (1) (2010) 13–24.

[13] B. Van Houdt, Performance comparison of aggressive push and traditional pull strategies in large distributed systems, in: Proceedings of QEST 2011, Aachen (Germany), IEEE Computer Society, 2011, pp. 265–274.

[14] D. Eager, E. Lazowska, J. Zahorjan, Adaptive load sharing in homogeneous distributed systems, IEEE Transactions on Software Engineering SE-12 (5) (1986) 662 –675.

[15] R. Mirchandaney, D. Towsley, J. A. Stankovic, Adaptive load sharing in heterogeneous distributed systems, J. Parallel Distrib. Comput. 9 (4) (1990) 331–346. doi:http://dx.doi.org/10.1016/0743-7315(90)90118-9.

[16] I. Van Spilbeeck, B. Van Houdt, Performance of rate-based pull and push strategies in heterogeneous networks, Performance Evaluation 91 (2015) 2 – 15, special Issue: Performance 2015. doi:http://dx.doi.org/10.1016/j.peva.2015.06.002.
URL `http://www.sciencedirect.com/science/article/pii/S0166531615000504`

[17] M. S. Squillante, R. D. Nelson, Analysis of task migration in shared-memory multiprocessor scheduling, SIGMETRICS Perform. Eval. Rev. 19 (1) (1991) 143–155. doi:10.1145/107972.107987.
URL `http://doi.acm.org/10.1145/107972.107987`

[18] W. Minnebo, B. Van Houdt, A fair comparison of pull and push strategies in large distributed networks, IEEE/ACM Transactions on Networking 22 (2014) 996–1006.

[19] W. Minnebo, B. Van Houdt, Improved rate-based pull and push strategies in large distributed networks, in: IEEE MASCOTS'13, 2013, pp. 141–150.

[20] M. Mitzenmacher, Analyses of load stealing models based on families of differential equations, Theory of Computing Systems 34 (2001) 77–98.

[21] S. Dhakal, Load balancing in delay-limited distributed systems, Ph.D. thesis, The University of New Mexico (2003).

[22] R. Mirchandaney, Adaptive load sharing in the presence of delays, Ph.D. thesis, Yale University (1988).

[23] W. Minnebo, B. Van Houdt, Analysis of rate-based pull and push strategies with limited migration rates in large distributed networks, ACM, 2016. doi:10.4108/eai.14-12-2015.2262564.

[24] M. Benaïm, J. Le Boudec, On mean field convergence and stationary regime, CoRR abs/1111.5710.

[25] F. de Blasi, G. Pianigiani, Uniqueness for differential equations implies continuous dependence only in finite dimension, Bulletin of the London Mathematical Society 18 (4) (1986) 379–382.

Appendix A. Global attraction

We start by proving that there exists a global attractor for the set of ODEs given by (2-4). The proof proceeds in the same manner as the proof of Theorem 1 in [6] and relies on monotonicity. Let $K > T$ and $0 \leq c < 1$, consider the set of ODEs given by (2-4) for $i = 1$ to K with the boundary conditions $x_0(t) = 1$, $x_{K+1}(t) = c \geq 0$ and $x_i(0) = g_i$ for $t \geq 0$ and $i = 1, \ldots, K$. We refer to this set of ODEs as the *truncated* system.

Lemma 1. *Assume $1 = g_0 \geq g_1 \geq g_2 \geq \ldots \geq g_K \geq g_{K+1} = c \geq 0$, then the solution of the truncated system satisfies $1 = x_0(t) \geq x_1(t) \geq x_2(t) \geq \ldots \geq x_K(t) \geq x_{K+1}(t) = c \geq 0$ for all t.*

Proof. As the solution of the truncated ODE is continuous in the initial values, it suffices to prove the lemma in case of strict inequalities. Assume there exists a $t_0 > 0$ where the inequalities no longer hold, then there either exists an i such that $x_{i-1}(t_0) > x_i(t_0) = x_{i+1}(t_0)$ or a j such that $x_{j-1}(t_0) = x_j(t_0) > x_{j+1}(t_0)$ as $1 = x_0(t_0) > x_{K+1}(t_0) = c$. In the first case we have $dx_i(t_0)/dt \geq \lambda(x_{i-1}(t_0) - x_i(t_0)) > 0$ and $dx_{i+1}(t_0)/dt \leq 0$, which contradicts the fact that $x_i(t) > x_{i+1}(t)$ for $t < t_0$. In the second case $dx_j(t_0)/dt \leq x_{j+1}(t_0) - x_j(t_0) < 0$ and $dx_{j-1}(t_0)/dt \geq 0$, contradicting $x_{j-1}(t) > x_j(t)$ for $t < t_0$. □

Lemma 2. *Let $x_i^{(1)}(t)$ and $x_i^{(2)}(t)$, for $i = 1, \ldots, K$, be two solutions of the first K ODEs of (2-4), with $x_i^{(1)}(0) \geq x_i^{(2)}(0)$ for $i = 1, \ldots, K$ and $x_{K+1}^{(1)}(t) \geq x_{K+1}^{(2)}(t)$ for all t, then $x_i^{(1)}(t) \geq x_i^{(2)}(t)$ for all t.*

Proof. As in Lemma 1 a proof in case of strict inequalities suffices. Assume $x_i^{(1)}(t_0) = x_i^{(2)}(t_0)$ at time t_0 for some i, while $x_i^{(1)}(t) > x_i^{(2)}(t)$ for $t < t_0$. Assume i is the largest index for which this equality holds (note $i \leq K$ as

$x_{K+1}^{(1)}(t) > x_{K+1}^{(2)}(t)$ for all t). We now argue that $dx_i^{(1)}(t_0)/dt > dx_i^{(2)}(t_0)/dt$, which contradicts $x_i^{(1)}(t) > x_i^{(2)}(t)$ for $t < t_0$. For $i > T$, we have

$$\frac{dx_i^{(1)}(t_0)}{dt} - \frac{dx_i^{(2)}(t_0)}{dt} = \underbrace{\lambda(x_{i-1}^{(1)}(t_0) - x_{i-1}^{(2)}(t_0))}_{\geq 0} + \underbrace{(x_{i+1}^{(1)}(t_0) - x_{i+1}^{(2)}(t_0))}_{>0}$$
$$+ r\left[(1 - x_{B+1}^{(2)}(t_0))(x_i^{(2)}(t_0) - x_{i+1}^{(2)}(t_0)) - (1 - x_{B+1}^{(1)}(t_0))(x_i^{(1)}(t_0) - x_{i+1}^{(1)}(t_0))\right].$$

As $x_{B+1}^{(1)}(t_0) \geq x_{B+1}^{(2)}(t_0)$ and $x_i^{(1)}(t_0) = x_i^{(2)}(t_0)$, the last term is at least $r(1 - x_{B+1}^{(2)}(t_0))(x_{i+1}^{(1)}(t_0) - x_{i+1}^{(2)}(t_0)) \geq 0$.

For $B + 2 \leq i \leq T$, $dx_i^{(1)}(t_0)/dt - dx_i^{(2)}(t_0)/dt$ is identical to the first two terms of the case with $i > T$ and is therefore strictly positive. Finally, for $1 \leq i \leq B + 1$, we have

$$\frac{dx_i^{(1)}(t_0)}{dt} - \frac{dx_i^{(2)}(t_0)}{dt} = \underbrace{\lambda(x_{i-1}^{(1)}(t_0) - x_{i-1}^{(2)}(t_0))}_{\geq 0} + \underbrace{(x_{i+1}^{(1)}(t_0) - x_{i+1}^{(2)}(t_0))}_{>0}$$
$$+ r\left[x_{T+1}^{(1)}(t_0)(x_{i-1}^{(1)}(t_0) - x_i^{(1)}(t_0) - x_{T+1}^{(2)}(t_0))(x_{i-1}^{(2)}(t_0) - x_i^{(2)}(t_0))\right].$$

Since $x_{T+1}^{(1)}(t_0) \geq x_{T+1}^{(2)}(t_0)$ and $x_i^{(1)}(t_0) = x_i^{(2)}(t_0)$, the last term is at least $rx_{T+1}^{(1)}(t_0))(x_{i-1}^{(1)}(t_0) - x_{i-1}^{(2)}(t_0)) \geq 0$. □

Note that in the above lemma we do not demand that $x_{K+1}^{(1)}(t)$ is constant as a function of t. The result also implies the monotonicity of the truncated systems.

Define $\bar{E} = \{(x_i)_{i\geq 0} | 1 = x_0 \geq x_1 \geq ... \geq 0\}$ and E the subset of $\bar{E}$ such that additionally $\sum_{i=0}^{\infty} x_i < \infty$ holds.

Lemma 3. *Let $g \in \bar{E}$, then the unique solution $x_i(t)$ of (2-4) with $x_i(0) = g_i$, for $i \geq 0$, is obtained as the limit of the unique solutions $x_i^{<K>}(t)$ of the truncated systems with $x_i^{<K>}(0) = g_i$ for $i = 1, \ldots, K$ and $x_{K+1}^{<K>}(t) = 0$.*

Proof. The proof is identical to the one of Lemma 3 in [6]. The existence of the limit is based on the fact that $x_{K+1}^{<K+1>}(t) \geq 0 = x_{K+1}^{<K>}(t)$ due to Lemma 1; hence, Lemma 2 implies that $x_i^{<K>}(t)$ does not decrease as a function of K for fixed i and t. □

Combining Lemma 2 and 3, we immediately have:

Lemma 4 (Monotonicity). *Let $x_i^{(1)}(t)$ and $x_i^{(2)}(t)$ be the unique solution of (2-4) with $x_i^{(k)}(0) = g_i^{(k)}$, for $k = 1, 2$ and $i \geq 0$, and $g_i^{(1)} \geq g_i^{(2)}$ for all i, then $x_i^{(1)}(t) \geq x_i^{(2)}(t)$ for all i and t.*

Define $v_k(x) = \sum_{i=k}^{\infty} x_i$ for $x \in \bar{E}$. Note that $v_1(x) < \infty$ whenever $x \in E$.

Lemma 5. *Let $x_i(t)$ be the unique solution of (2-4) with $x_i(0) = g_i$ for $i \geq 0$ and $g \in E$. Assume $\pi \in E$ is a fixed point with $\pi_1 = \lambda < 1$ and assume $g_i \leq \pi_i$ for all i or $g_i \geq \pi_i$ for all i. Then $v_k(x(t))$ is bounded uniformly in t and we have*

$$\lim_{t\to\infty}(x_i(t) - \pi_i) = 0,$$

for all $i \geq 1$.

Proof. If $g_i \leq \pi_i$ for all i then $x_i(t) \leq \pi_i$ for all i by Lemma 4 and $v_1(x(t)) \leq \sum_i \pi_i < \infty$. In case $g_i \geq \pi_i$ for all i, we have $x_1(t) \geq \pi_1 = \lambda$. Hence, it suffices to note that $dv_1(x(t))/dt = \lambda - x_1(t) \leq 0$ to conclude that $v_1(x(t)) \leq \sum_i g_i < \infty$. As $0 \leq v_k(x(t)) \leq v_1(x(t))$ the uniform boundedness follows for all k.

To prove the remaining part we rely on the equality

$$\frac{dv_k(x(t))}{dt} = \lambda x_{k-1}(t) - x_k(t) - r x_{\max(T+1,k)}(t)(1 - x_{\min(B+1,k-1)}(t)),$$

which can be obtained by summing the ODEs in (2-4) for $i \geq k$. As π is a fixed point, we have $\lambda\pi_{k-1} - \pi_k - r\pi_{\max(T+1,k)}(1 - \pi_{\min(B+1,k-1)}) = 0$. Subtracting this from $dv_k(x(t))/dt$ and adding $rx_{\max(T+1,k)}(1 - \pi_{\min(B+1,k-1)}) - rx_{\max(T+1,k)}(1 - \pi_{\min(B+1,k-1)}) = 0$ yields

$$\begin{aligned}\frac{dv_k(x(t))}{dt} = {}& \lambda(x_{k-1}(t) - \pi_{k-1}) - (x_k(t) - \pi_k) \\ & + r x_{\max(T+1,k)}(t)(x_{\min(B+1,k-1)}(t) - \pi_{\min(B+1,k-1)}) \\ & - r(1 - \pi_{\min(B+1,k-1)})(x_{\max(T+1,k)}(t) - \pi_{\max(T+1,k)}). \qquad \text{(A.1)}\end{aligned}$$

To show that $\lim_{t\to\infty}(x_1(t) - \pi_1) = 0$, (A.1) implies

$$v_1(x(t)) - v_1(g) = \int_{s=0}^{t} (\pi_1 - x_1(s))ds.$$

As $v_1(x(t))$ is uniformly bounded in t we have $\int_{s=0}^{\infty}(\pi_1 - x_1(s))ds < \infty$. Further, the sign of $\pi_1 - x_1(t)$ is the same for all t and the derivative $dx_i(t)/dt \leq 1$, this yields that $x_1(t) - \pi_1$ must tend to zero as t goes to infinity.

Next we argue that $\lim_{t\to\infty}(x_i(t) - \pi_i) = 0$ for $i = 2$ and $i = T + 1$. By (A.1) we have

$$\begin{aligned}v_2(x(t)) - v_2(g) = \int_{s=0}^{t} & [\lambda(x_1(s) - \pi_1) - (x_2(s) - \pi_2) \\ & + r x_{T+1}(s)(x_1(s) - \pi_1) - r(1 - \pi_1)(x_{T+1}(s) - \pi_{T+1})]ds.\end{aligned}$$

Therefore by the uniform boundedness of $v_2(x(t))$ and $\int_{s=0}^{\infty}(x_1(s) - \pi_1) < \infty$, we find

$$\int_{s=0}^{\infty}[(\pi_2 - x_2(s)) + r(1 - \pi_1)(\pi_{T+1} - x_{T+1}(s))]ds < \infty.$$

Hence, $\int_{s=0}^{\infty}(x_i(s)-\pi_i)ds<\infty$ for $i=2$ and $i=T+1$ as $(\pi_2-x_2(t))$ and $(\pi_{T+1}-x_{T+1}(t))$ have the same sign for all t. Thus, $\lim_{t\to\infty}(x_i(t)-\pi_i)=0$ for $i=2$ and $T+1$ because $dx_i(t)/dt\leq 1$ for all i and t. The proof for $i\neq 1,2$ and $T+1$ proceeds similarly by induction on i using (A.1) and the uniform boundedness of $v_i(x(t))$. □

Theorem 11. *Let $x_i(t)$ be the unique solution of (2-4) with $x_i(0)=g_i$ for $i\geq 0$ and $g\in E$. Assume $\pi\in E$ is a fixed point and $\pi_1=\lambda<1$, then*

$$\lim_{t\to\infty}(x_i(t)-\pi_i)=0,$$

for all $i\geq 1$.

Proof. Define $(g^+)_i=max(g_i,\pi_i)$ and $(g^-)_i=min(g_i,\pi_i)$ for $i\geq 1$, then $g^-\leq g\leq g^+$. Let $x^+(t)$ be the unique solution of (2-4) with $x^+(0)=g^+$ and define $x^-(t)$ similarly. By Lemma 5 we know that both $x^-(t)$ and $x^+(t)$ converge to π entry-wise, while Lemma 4 indicates that $x^-(t)\leq x(t)\leq x^+(t)$ for all $t\geq 0$. □

Appendix B. Weak convergence of invariant measures

Consider a finite system of N queues operating under a fixed rate pull or push strategy with Poisson arrivals and exponential service times. Let $X_i^N(t)$ be the fraction of queues with at least i jobs at time t. Clearly, $(X_0^N(t),X_1^N(t),\ldots)$ forms a Markov process on the state space $\{(x_0,x_1,\ldots)|1=x_0\geq x_1\geq\ldots\geq 0,x_iN\in\{0,\ldots,N\}\}\subseteq E$. Let Π^N be the stationary measure of the Markov process corresponding to the system with N queues. The next theorem indicates that the sequence of measures Π^N converges weakly to the dirac measure δ_π, where π is the unique fixed point of (2-4).

Theorem 12. *Consider the metric space (E,ρ) where $\rho(x,y)=\sum_{i\geq 0}|x_i-y_i|/2^i$, then the sequence of probability measures Π^N converges weakly to the dirac measure δ_π, that is,*

$$\lim_{N\to\infty}\int f(y)\Pi^N(dy)=f(\pi),$$

for any bounded continuous function f from (E,ρ) to $\mathbb{R}$.

Proof. (Sketch) The main idea is to prove that any subsequence $\{\Pi^{N_k}\}$ of $\{\Pi^N\}$ has a further sub-subsequence that converges weakly to the same limit, being δ_π, as this implies that the sequence $\{\Pi^N\}$ converges weakly to δ_π (this property is sometimes called the Urysohn property). The first step exists in showing that every subsequence $\{\Pi^{N_k}\}$ has a weakly convergent sub-subsequence (without the need that all of them have the same limit). This can be done by proving that the sequence of probability measures $\{\Pi^N\}$ is tight and by applying the Prokhorov's theorem (which states that tightness implies relative compactness). To prove the tightness we can proceed as in [18] by first using a coupling argument to show that the queue length distribution of the system that relies on a pull/push strategy is bounded by a constant plus the queue length distribution of a set of N independent M/M/1 queues. Tightness of $\{\Pi^N\}$ therefore follows from the tightness of the set $\{\hat{\Pi}^N\}$ proven in [18] (for $T = 0$ and $B = 1$), where $\hat{\Pi}^N$ is the invariant measure of the system of N independent M/M/1 queues.

The next step exists in showing that all the convergent sub-subsequences must have the same limit δ_π. For this purpose we make use of [24, Corollary 1]. This corollary shows that any limit point of a subsequence $\{\Pi^{N_k}\}$ is the dirac measure δ_π if the following additional three conditions are met (given the tightness). First, the stochastic system should converge weakly to the limiting ODE, that is, for any fixed T, the conditional expectation $E[h(X^N(T))|X^N(0) = g]$ converges to $h(x(T, g))$, where $x(t, g)$ is the unique solution of the ODE with boundary condition $x(0, g) = g \in E$, for any bounded continuous function $h : E \to \mathbb{R}$. In fact, a stronger convergence result can be established by relying on [20, Theorem 3.13] completely analogue to [18] (for $T = 0$ and $B = 1$). Second, the deterministic limit process should be a continuous semi-flow, which mainly involves checking the requirement that $x(t, g)$ is continuous in both t and g. This continuity follows (see [25]) from the observation that the ODE characterized by (2-4) is Lipschitz continuous on the Banach space (E, ρ). Finally, the ODE should have a unique global attractor. In Theorem 11 we showed that all trajectories starting in E converge to the unique fixed point π entry-wise, which is sufficient to have convergence under the metric ρ. □