Performance of Rate-Based Pull and Push Strategies in Heterogeneous Networks

I. Van Spilbeeck and Benny Van Houdt
Department of Mathematics and Computer Science,
University of Antwerp - iMinds, Belgium
{ignace.vanspilbeeck,benny.vanhoudt}@uantwerpen.be

Abstract

Response times in large distributed systems can be reduced by exchanging jobs between idle servers and servers with pending jobs. When a pull strategy is deployed the initiative to exchange jobs is taken by the idle servers, while servers with pending jobs initiate the exchange when a push strategy is implemented. In this paper the performance of a class of rate-based pull and push strategies for large heterogeneous networks is studied using a mean field model. These strategies have the advantage that the rate at which servers *probe* other servers to initiate a job exchange can be controlled, allowing a fair comparison between pull and push strategies.

Based on two natural conjectures we derive a simple condition for the required probe rate to establish system stability when the system size becomes large and consists of two types of servers. In some specific cases we show that this condition coincides with the existence of a unique positive fixed point for which we also present an explicit expression. This fixed point is used to express the queue length distribution and mean response time in the system in explicit form. The accuracy of both the stability condition and mean queue lengths as predicted by the mean field model is validated using time-consuming simulation experiments. We end the paper with some numerical results that compare the performance of the rate-based pull and push strategies in a heterogeneous setting.

1 Introduction

Systems consisting of a large number of servers benefit from the ability to exchange jobs between idle servers and servers with pending jobs. Roughly speaking the strategies used to exchange jobs between servers can be classified in two categories: pull and push strategies depending on which servers take the initiative to exchange jobs. When a traditional pull strategy is used, servers that become idle trigger the transmission of a probe message to a number of randomly selected servers, while with a traditional push strategy the transmission of probe messages is triggered by job arrivals that find the queue (sufficiently) occupied. These probe messages are used to locate a server that is willing to exchange a job. The more probe messages one transmits the more likely such an exchange can occur and more pronounced the reduction in the response times become.

The performance of both the traditional push and pull strategies (as well as generalizations thereof) has been studied by a number of authors [4, 13, 11, 6]. A homogeneous system with Poisson arrivals and exponential job lengths was analyzed in [4, 3] using a decoupling assumption that relied on the numerical solution of some nonlinear equation. This approach was subsequently extended to heterogeneous systems in [11] again by relying on a decoupling assumption. While the insights provided by these models are very valuable, the comparison is somewhat biased as the rate R at which the pull and push strategies transmit probe messages may differ significantly for a given job arrival and job service rate (see Section VI in [10]).

To allow for a more fair comparison the rate-based pull and push strategies were introduced in [8, 10]. The main difference between the rate-based and traditional pull strategy is that probes are no longer transmitted whenever a server becomes idle, but instead are transmitted at some rate r *as long as* the server remains idle. Similarly, job arrivals do not trigger the transmission of probe messages under the rate-based push strategy, but instead probe messages are transmitted at some rate $\hat{r}$ whenever there are pending jobs. For a given set of system parameters (e.g. arrival and service rates), we can set the parameters r and $\hat{r}$ such that the mean number of probe messages R that is transmitted per time unit coincides for both strategies. Apart from allowing a more fair comparison, these rate-based strategies have the additional benefit that closed-form expressions can be derived for the mean response time in case of a large homogeneous system, avoiding the need to solve any nonlinear equations.

In this paper we extend the analysis of [10] to a heterogeneous network. We restrict ourselves to a system that consists of two types of servers (each type with its own arrival rate), though the model can be easily extended to more types. The main contributions of the paper are as follows: (i) We extend the mean field model of [10] to a heterogeneous network, this generalization is not hard (Section 3). (ii) Based on this mean field model and two natural conjectures we provide a simple necessary and sufficient stability condition when the system becomes large (Section 4). (iii) In some specific cases we show that this stability condition coincides with the existence of a unique positive fixed point for the set of ODEs that captures the evolution of the mean field model (Section 5). (iv) We derive an explicit expression for the unique fixed point in these special cases

(Sections 5 and 6). (v) Finally, we validate the mean field model using simulation experiments (Section 7) and provide insights on the comparison of the rate-based pull and push strategies using our model (Section 8).

2 Pull and push strategies

We consider a system consisting of N servers, each server can process one job at a time and can store infinitely many waiting jobs. To transfer jobs between the servers we consider the following two rate-based strategies introduced in [10]:

1. **Pull:** Whenever a server has $i = 0$ jobs in its queue, meaning the server is idle, the server will generate probe messages at rate r. Thus, as long as the server remains idle, probes are sent according to a Poisson process with rate r. This process is interrupted whenever the server becomes busy. A probe is successful and results in the exchange of a job whenever there are jobs waiting to be served in the probed server.

2. **Push:** Whenever a server has $i \geq 2$ jobs in its queue, meaning $i - 1$ jobs are waiting to be served, the server will generate probe messages at rate $\hat{r}$. Thus, as long as the number of jobs in the server remains above 1, probes are sent according to a Poisson process with rate $\hat{r}$. Whenever the queue length i drops to 1, this process is interrupted and remains interrupted as long as the queue length remains below 2. A probe is successful and results in the exchange of a job whenever the probed server is idle.

We do not consider hybrid strategies in which servers both pull and push jobs as these were argued to be inferior in a homogeneous setting in [10]. It should also be possible to extend the results in this paper to a more general class of rate-based pull and push strategies as was done in [9] for the homogeneous network.

The main objectives of this paper are to study the performance of these strategies in a heterogeneous network, that is, in a network where the number of jobs arriving per time unit and the processing speed is *not* the same in each server. To this end we assume that the servers can be partitioned in K types of servers, where servers of the same type are assumed to have the same processing speed and arrival rate. To determine the server that is being probed the servers make use of a stochastic $K \times K$ matrix P. Entry $P_{k,k'}$ of P determines the probability that a type k server probes a random type k' server, for $k, k' \in \{1, \ldots, K\}$. In other words, if we have $N_{k'}$ type k' servers, a specific type k' server is probed by a type k server with probability $P_{k,k'}/N_{k'}$.

Throughout the paper we limit ourselves to $K = 2$ types of servers, but note that the system dynamics of the mean field model in Section 3 is trivial to extend to $K > 2$ server types.

3 The mean field model

In this section we introduce a mean field model that is used to study the performance of the pull and push strategy in a heterogeneous setting. We assume that we have $K = 2$ types of servers and denote γ_1 as the fraction of type 1 servers and $\gamma_2 = 1 - \gamma_1$. Each server can store infinitely many jobs. Let λ_k be the Poisson arrival rate of jobs in a type k server, for $k = 1, 2$, and μ_k the rate of a type k server (we assume exponential job durations). Define $\rho_k = \lambda_k/\mu_k$, $\lambda = \gamma_1\lambda_1 + \gamma_2\lambda_2$, $\mu = \gamma_1\mu_1 + \gamma_2\mu_2$ and $\rho = \lambda/\mu$. As in [4, 13, 11, 6, 10] we assume that the time required to transfer probe messages and jobs between different servers can be neglected in comparison with the processing time (i.e., the transfer times are assumed to be zero).

Given the above assumptions, it is clear that we obtain an N-dimensional Markov chain by simply keeping track of the number of jobs present in each of the N servers. Further, it is not hard to see that this Markov chain belongs to the family of *density dependent* Markov chains in the sense of Kurtz [7, 5] (as the rate of change of the number of jobs in a specific server is affected only by the content of the other servers through the fraction of idle servers and servers with pending jobs). As this Markov chain does not appear to have a product form we approximate it using a mean field model. One can show, using [12, Theorem 3.13]), that this mean field model corresponds to the limit process of the family of N-dimensional Markov chains as N tends to infinity over any finite time scale. Proving that the convergence extends to the stationary regime, as was done in [10] for the homogeneous case, is in general far more challenging. This is especially true in this case as each server can store infinitely many jobs, which implies that it is not sufficient to prove that the set of ODEs has a global attractor [1].

3.1 System dynamics

The models introduced in this section are a generalization of the model introduced in [10] which was restricted to a homogeneous network (that is, $K = 1$). In contrast to [10] we can no longer make use of a single set of ODEs to capture the behavior of both the pull and push strategy. The model makes use of the variables $x_{k,i}(t)$, with $i \geq 0$ and $k = 1, 2$, that represent the fraction of servers of type k that contain i or more jobs at time t and its evolution is described by means of a set of ODEs. To distinguish between the variables of the pull and push strategy we add a hat on all the variables that relate to the push strategy.

Pull: We start with the drift equations of the variables $x_{k,i}(t)$ when using the pull strategy:

$$\begin{aligned}\frac{dx_{k,1}(t)}{dt} &= \lambda_k(x_{k,0}(t) - x_{k,1}(t)) - \mu_k(x_{k,1}(t) - x_{k,2}(t)) \\ &\quad + r\left(\sum_{k'=1}^{2} \frac{x_{k',2}(t)}{x_{k',0}(t)} P_{k,k'}\right)(x_{k,0}(t) - x_{k,1}(t)), \qquad (1)\\ \frac{dx_{k,i}(t)}{dt} &= \lambda_k(x_{k,i-1}(t) - x_{k,i}(t)) - \mu_k(x_{k,i}(t) - x_{k,i+1}(t)) \\ &\quad - r\left(\sum_{k'=1}^{2} (x_{k',0}(t) - x_{k',1}(t))P_{k',k}\right)\frac{x_{k,i}(t) - x_{k,i+1}(t)}{x_{k,0}(t)}, \qquad (2)\end{aligned}$$

for $i \geq 2$ and $x_{k,0}(t) = \gamma_k$. The drift of these variables is composed of three terms: one due to job arrivals, one due to job completions and one due to job transfers. The number of type k servers with i or more jobs, for $i \geq 1$, increases by one whenever a job arrives in a server holding exactly $i-1$ jobs, that is, at rate $\lambda_k(x_{k,i-1}(t) - x_{k,i}(t))$. Similarly, it decreases by one if a service completion occurs in a server with exactly i jobs, i.e., at rate $\mu_k(x_{k,i}(t) - x_{k,i+1}(t))$. The changes due to job transfers are somewhat more involved.

Jobs are always transferred between a server with at least 2 jobs and an empty server. Hence, they cause an increase in the number of servers with 1 or more jobs and a decrease in the number of servers with i or more jobs, for some $i \geq 2$. The rate of increase of $x_{k,1}(t)$ is equal to the fraction of type k servers that is empty $(x_{k,0}(t) - x_{k,1}(t))$ times the rate r at which these servers probe times the probability that such a probe is successful. This latter probability can be expressed as $\sum_{k'=1}^{2} P_{k,k'}x_{k',2}(t)/x_{k',0}(t)$ as a type k server probes a type k' server with probability $P_{k,k'}$ and a type k' server contains at least 2 jobs with probability $x_{k',2}(t)/\gamma_k$. The rate at which $x_{k,i}(t)$ decreases, for $i \geq 2$, is given by the probability that a type k server contains exactly i jobs $(x_{k,i}(t) - x_{k,i+1}(t))/\gamma_k$ times the rate at which type k servers are probed. This latter rate equals $r\sum_{k'=1}^{2}(x_{k',0}(t) - x_{k',1}(t))P_{k',k}$ as $(x_{k',0}(t) - x_{k',1}(t))$ is the fraction of empty type k' servers and these probe a type k server at rate $rP_{k',k}$.

Push: For the push strategy we have the following drift equations. They only differ from the drift equations of the pull strategy in the terms corresponding to the job transfers.

$$\begin{aligned}\frac{d\hat{x}_{k,1}(t)}{dt} &= \lambda_k(\hat{x}_{k,0}(t) - \hat{x}_{k,1}(t)) - \mu_k(\hat{x}_{k,1}(t) - \hat{x}_{k,2}(t)) \\ &\quad + \hat{r}\left(\sum_{k'=1}^{2} \hat{x}_{k',2}(t)P_{k',k}\right)\frac{\hat{x}_{k,0}(t) - \hat{x}_{k,1}(t)}{\hat{x}_{k,0}(t)}, \qquad (3)\\ \frac{d\hat{x}_{k,i}(t)}{dt} &= \lambda_k(\hat{x}_{k,i-1}(t) - \hat{x}_{k,i}(t)) - \mu_k(\hat{x}_{k,i}(t) - \hat{x}_{k,i+1}(t)) \\ &\quad - \hat{r}\left(\sum_{k'=1}^{2} \frac{\hat{x}_{k',0}(t) - \hat{x}_{k',1}(t)}{\hat{x}_{k',0}(t)}P_{k,k'}\right)(\hat{x}_{k,i}(t) - \hat{x}_{k,i+1}(t)), \qquad (4)\end{aligned}$$

With the push strategy the servers with pending jobs initiate the job transfers. As such the rate at which these cause an increase in $\hat{x}_{k,1}(t)$ is equal to the rate $\hat{r}\left(\sum_{k'=1}^{2}\hat{x}_{k',2}(t)P_{k',k}\right)$ at which type k servers receive probes times the probability $(\hat{x}_{k,0}(t)-\hat{x}_{k,1}(t))/\gamma_k$ that a type k server is empty. The decrease in $\hat{x}_{k,i}(t)$, for $i \geq 2$, on the other hand is given by the rate $\hat{r}(\hat{x}_{k,i}(t)-\hat{x}_{k,i+1}(t))$ at which the type k servers with exactly i jobs probe times the probability $P_{k,k'}$ that they probe a type k' server (for any k') times the probability $(\hat{x}_{k',0}(t)-\hat{x}_{k',1}(t))/\gamma_{k'}$ that this type k' server is empty.

3.2 Fixed points

It is useful to note that the drift equation of the pull strategy can be written as

$$\frac{dx_{k,1}(t)}{dt} = (\lambda_k+\eta_k(t))(x_{k,0}(t)-x_{k,1}(t)) - \mu_k(x_{k,1}(t)-x_{k,2}(t)), \tag{5}$$

$$\frac{dx_{k,i}(t)}{dt} = \lambda_k(x_{k,i-1}(t)-x_{k,i}(t)) - (\mu_k+\sigma_k(t))(x_{k,i}(t)-x_{k,i+1}(t)), \tag{6}$$

for $i \geq 2$, by defining $\sigma_k(t)$ and $\eta_k(t)$ as

$$\eta_k(t) = r\sum_{k'=1}^{2}\frac{x_{k',2}(t)}{\gamma_{k'}}P_{k,k'}, \tag{7}$$

$$\sigma_k(t) = \frac{r}{\gamma_k}\sum_{k'=1}^{2}(x_{k',0}(t)-x_{k',1}(t))P_{k',k}. \tag{8}$$

The same holds for the push strategy if we replace x by $\hat{x}$, η by $\hat{\eta}$ and σ by $\hat{\sigma}$ with

$$\hat{\eta}_k(t) = \frac{\hat{r}}{\gamma_k}\sum_{k'=1}^{2}\hat{x}_{k',2}(t)P_{k',k}, \tag{9}$$

$$\hat{\sigma}_k(t) = \hat{r}\sum_{k'=1}^{2}\frac{\hat{x}_{k',0}(t)-\hat{x}_{k',1}(t)}{\gamma_{k'}}P_{k,k'}. \tag{10}$$

Property 1. *Assume $x = \{x_{k,i} | i \geq 0, k = 1,2\}$ with $x > 0$ and $\sum_{i\geq 0}(x_{k,i}-x_{k,i+1}) = \gamma_k$ is a positive fixed point of the set of ODEs given by* (1) *and* (2)*. Let η_k and σ_k be given by* (7) *and* (8)*, respectively, when replacing $x_{k',i}(t)$ by $x_{k',i}$. Let $\pi_{k,i} = (x_{k,i}-x_{k,i+1})$, then*

$$\pi_{k,i} = \pi_{k,0}\frac{\lambda_k+\eta_k}{\mu_k}\left(\frac{\lambda_k}{\sigma_k+\mu_k}\right)^{i-1}, \tag{11}$$

for $i \geq 1$ and $\pi_{k,0}$ such that $\sum_{i\geq 0}\pi_{k,i} = \gamma_k$. Further,

$$\mu - \lambda = \mu_1\pi_{1,0} + \mu_2\pi_{2,0}. \tag{12}$$

Proof. From (5) and (6) we observe that $\pi_{k,i}$ is an invariant vector of the rate matrix Q_k

$$Q_k = \begin{bmatrix} -\lambda_k-\eta_k & \lambda_k+\eta_k & 0 & \cdots & \\ \mu_k & -\lambda_k-\mu_k & \lambda_k & 0 & \cdots \\ 0 & \sigma_k+\mu_k & -\lambda_k-\sigma_k-\mu_k & \lambda_k & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Equation (11) therefore follows from the birth death structure of Q_k. The equality $\mu - \lambda = \mu_1 \pi_{1,0} + \mu_2 \pi_{2,0}$ is obtained by demanding that $\sum_{k,i} \frac{dx_{k,i}(t)}{dt} = 0$ and verifying that $\sum_k \eta_k \pi_{k,0} = \sum_k \sigma_k x_{k,2}$. □

The next property is proven in exactly the same manner for the push strategy.

Property 2. *Assume* $\hat{x} = \{\hat{x}_{k,i} | i \geq 0, k = 1, 2\}$ *with* $\hat{x} > 0$ *and* $\sum_{i \geq 0} (\hat{x}_{k,i} - \hat{x}_{k,i+1}) = \gamma_k$ *is a positive fixed point of the set of ODEs given by* (3) *and* (4)*. Let* $\hat{\eta}_k$ *and* $\hat{\sigma}_k$ *be given by* (9) *and* (10)*, respectively, when replacing* $\hat{x}_{k',i}(t)$ *by* $\hat{x}_{k',i}$*. Let* $\hat{\pi}_{k,i} = (\hat{x}_{k,i} - \hat{x}_{k,i+1})$*, then*

$$\hat{\pi}_{k,i} = \hat{\pi}_{k,0} \frac{\lambda_k + \hat{\eta}_k}{\mu_k} \left(\frac{\lambda_k}{\hat{\sigma}_k + \mu_k} \right)^{i-1}, \tag{13}$$

for $i \geq 1$ *and* $\hat{\pi}_{k,0}$ *such that* $\sum_{i \geq 0} \hat{\pi}_{k,i} = \gamma_k$*. Further,*

$$\mu - \lambda = \mu_1 \hat{\pi}_{1,0} + \mu_2 \hat{\pi}_{2,0}. \tag{14}$$

4 Stability

Let $\Omega^{(N)}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ be the set of r values for the pull strategy for which the corresponding density dependent N dimensional Markov chain (in the sense of Kurtz) is stable and define $\hat{\Omega}^{(N)}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ similarly for the push strategy. Clearly, if $\rho_1, \rho_2 < 1$ we have stability for all $r \geq 0$. Existing results on the stability of multidimensional queueing systems [14] suggest that in order to determine this set of r values, one needs to study the service rates in the *dominating* systems, that is, the systems where either all the type 1 or all the type 2 queues have an infinite queue length. For finite N these service rates are hard to determine, as such we consider the case where N tends to infinity.

Conjecture 1. *Let* $\Omega_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ *be the set of* r *values for which* (1)-(2) *has a unique positive fixed point, then* $\lim_{N \to \infty} \Omega^{(N)}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)} = \Omega_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ *and the same holds for the push strategy.*

We numerically validate this approximation in Section 7 and note that an approximation for the stability region of a multidimensional system using a mean field model was also introduced in [2].

As for the necessary and sufficient condition for the existence of a (unique) positive fixed point, we believe the following natural conjecture holds, which we prove in the next section for the pull strategy when $P_{2,2} = 1$ and for the push strategy when $P_{1,1} = 1$. Note that (12) and (14) imply that a positive fixed point cannot exist if $\rho \geq 1$.

Conjecture 2. *Let $\rho < 1$ and assume without loss of generality that $\rho_1 \leq \rho_2$. The set of ODEs for both the pull and push strategy has a positive fixed point if and only if the arrival rate λ_2 is less than the rate at which type 2 jobs are served in the* **dominating system** *where type 2 queues have an infinite number of waiting jobs at all times. Further, this fixed point is unique.*

We first note that due to (11) we have

$$\gamma_k = \pi_{k,0} \left[1 + \left(\frac{\lambda_k + \eta_k}{\mu_k} \right) \frac{1}{1 - \frac{\lambda_k}{\mu_k + \sigma_k}} \right], \tag{15}$$

$$\gamma_k = \pi_{k,0} \left(1 + \frac{\lambda_k + \eta_k}{\mu_k} \right) + x_{k,2}, \tag{16}$$

$$x_{k,2} = \frac{\lambda_k}{\mu_k + \sigma_k} (\gamma_k - \pi_{k,0}). \tag{17}$$

We now establish the following theorem for the pull strategy. It indicates that when $\rho < 1$, r has to be large enough such that sufficient type 2 jobs can be served by type 1 servers. When $\rho_2 < 1$, r_c is negative and setting $r = 0$ suffices. This is expected as we assumed that $\rho_1 \leq \rho_2$ and therefore both type 1 and 2 queues have a load below one, so no transfer of jobs is required to get a stable system.

Theorem 1. *Conjecture 2 for the* **pull** *strategy is equivalent to the following statement: the set $\Omega_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ is given by*

$$\Omega_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)} = \left\{ r \,\middle|\, r > r_c = \frac{\mu_1 \mu_2 (\rho_2 - 1)}{\mu - \lambda} \frac{\gamma_2}{1 - P_{1,1}} \right\}. \tag{18}$$

Proof. Consider the dominating system in which all the type 2 queues have infinite length. In such a system we have $\pi_{2,0} = 0$ and $x_{2,2} = \gamma_2$. Equations (7) and (8) therefore yield

$$\eta_1 = r \left(x_{1,2} P_{1,1} / \gamma_1 + P_{1,2} \right) \tag{19}$$

$$\sigma_1 = r \pi_{1,0} P_{1,1} / \gamma_1. \tag{20}$$

We first determine $\pi_{1,0}$ and $x_{1,2}$ in this dominating system. To obtain an equation for $x_{1,2}$ we use (16) with $k = 1$ to express $\pi_{1,0}$ in terms of $x_{1,2}$ (as η_1 depends on $x_{1,2}$ only). This allows us to express σ_1 in terms of $x_{1,2}$ via (20). Next, we plug this expression for $\pi_{1,0}$ and σ_1 into (15) with $k = 1$ where we also use (19) to obtain an equation where $x_{1,2}$ is the only unknown. This equation turns out to have a unique solution given by

$$x_{1,2} = \frac{\gamma_1 \lambda_1 \left(\lambda_1 + r(1 - P_{1,1}) \right)}{\mu_1^2 + r(\mu_1 - \lambda_1 P_{1,1})}. \tag{21}$$

Hence, using (16) we have

$$\pi_{1,0} = \frac{\gamma_1 (\mu_1 - \lambda_1)}{\mu_1 + r(1 - P_{1,1})}. \tag{22}$$

We have a unique positive fixed point x if and only if the arrival rate of the type 2 jobs is less than the service rate of the type 2 jobs in the dominating system provided that Conjecture 2 holds, that is,

$$\lambda_2 < \mu_2 + \frac{\gamma_1}{\gamma_2}\frac{\pi_{1,0}}{\gamma_1} r(1 - P_{1,1}), \tag{23}$$

as $\frac{\pi_{1,0}}{\gamma_1} r(1 - P_{1,1})$ is the rate at which the type 1 servers pull jobs from the type 2 servers and for each type 2 server we have γ_1/γ_2 type 1 servers.

Since $\rho < 1$, (23) can be rewritten as

$$r > \frac{\mu_1\mu_2(\rho_2 - 1)}{\mu - \lambda}\frac{\gamma_2}{1 - P_{1,1}}, \tag{24}$$

by relying on (22). □

Let us now establish a similar result for the push approach, where the proof is similar to Theorem 1 and is given in Appendix A.

Theorem 2. *Conjecture 2 for the* **push** *strategy is equivalent to the following statement: the set* $\hat{\Omega}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)}$ *is given by*

$$\hat{\Omega}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)} = \left\{ \hat{r} \,\middle|\, \hat{r} > \hat{r}_c = \frac{\mu_1\mu_2(\rho_2 - 1)}{\mu - \lambda}\frac{\gamma_1}{1 - P_{2,2}} \right\}.$$

5 Explicit results

5.1 Pull

In this section we present some explicit results for the pull strategy with $P_{2,2} = 1$, that is,

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} \\ 0 & 1 \end{bmatrix}.$$

It is worth noting at this stage that due to (11) we have an explicit expression for the fixed point x as soon as we have explicit expressions for $\pi_{1,0}, \pi_{2,0}, x_{1,2}$ and $x_{2,2}$.

When $P_{2,2} = 1$ the fixed point equations for η_k and σ_k under the pull strategy become:

$$\eta_1 = r\left(x_{1,2}P_{1,1}/\gamma_1 + x_{2,2}P_{1,2}/\gamma_2\right), \tag{25}$$

$$\eta_2 = r x_{2,2}/\gamma_2, \tag{26}$$

$$\sigma_1 = r\pi_{1,0}P_{1,1}/\gamma_1, \tag{27}$$

$$\sigma_2 = r(\pi_{1,0}P_{1,2} + \pi_{2,0})/\gamma_2. \tag{28}$$

Combining (16) with $k = 2$ with (26) gives the following expression for $\pi_{2,0}$ in terms $x_{2,2}$:

$$\pi_{2,0} = \frac{\gamma_2 \mu_2 (\gamma_2 - x_{2,2})}{\gamma_2(\lambda_2 + \mu_2) + r x_{2,2}}. \tag{29}$$

Equations (15) with $k = 2$, (26), (28) and (29) now enable us to express $\pi_{1,0}$ in terms of $x_{2,2}$

$$\pi_{1,0} = \frac{\gamma_2^2(\gamma_2 \lambda_2^2 + r x_{2,2}(\lambda_2 - \mu_2) - x_{2,2}\mu_2^2)}{r x_{2,2}(\gamma_2(\lambda_2 + \mu_2) + r x_{2,2}) P_{1,2}}. \tag{30}$$

Combining (29) and (30) yields

$$\pi_{1,0} = \frac{\gamma_2(\mu_2 \pi_{2,0} + \gamma_2(\lambda_2 - \mu_2))}{r x_{2,2} P_{1,2}}, \tag{31}$$

which can be combined with (12) to obtain the following simple expression for $\pi_{1,0}$ in terms of $x_{2,2}$:

$$\pi_{1,0} = \frac{\gamma_1(\mu_1 - \lambda_1)\gamma_2}{\gamma_2 \mu_1 + r x_{2,2} P_{1,2}}. \tag{32}$$

An expression for $x_{1,2}$ in terms of $\pi_{1,0}$ is readily obtained from (17) with $k = 1$:

$$x_{1,2} = \frac{\lambda_1(\gamma_1 - \pi_{1,0})\gamma_1}{\gamma_1 \mu_1 + r \pi_{1,0} P_{1,1}}. \tag{33}$$

Given equations (29) and (32) for $\pi_{2,0}$ and $\pi_{1,0}$, we can use (17) with $k = 2$ to obtain a quadratic equation for $x_{2,2}$. This quadratic equation has the form $f_2 x^2 + f_1 x + f_0 = 0$ with

$$\begin{aligned}
f_2 &= r(r(\mu - \lambda) + \gamma_2 \mu_2^2) P_{1,2}, \\
f_1 &= \gamma_2^2 \mu_1 \mu_2^2 + \gamma_2 r \left(\gamma_1(\mu_2 + \lambda_2)(\mu_1 - \lambda_1) P_{1,2} + \gamma_2 \mu_1(\mu_2 - \lambda_2) - \gamma_2 \lambda_2^2 P_{1,2} \right), \\
f_0 &= -\gamma_2^3 \lambda_2^2 \mu_1.
\end{aligned}$$

As $f_2 > 0$ and $f_0 < 0$, for $r > 0$, this quadratic equation has a unique positive root denoted as

$$\xi_{pos} = \frac{\sqrt{f_1^2 - 4 f_2 f_0} - f_1}{2 f_2}. \tag{34}$$

Note that this implies that we can therefore have *at most* one positive fixed point.

Lemma 1. *Let $\rho < 1$, $f(x) = f_2 x^2 + f_1 x + f_0$ and r_c as defined in* (18). *If $r_c > 0$, we have $f(\gamma_2) = 0$ for $r = r_c$, $f(\gamma_2) > 0$ for $r > r_c$, and $f(\gamma_2) < 0$ for $r \in (0, r_c)$. If $r_c \leq 0$, then $f(\gamma_2) > 0$ for $r > 0$.*

Proof. The result follows from verifying that $f(\gamma_2)$ can be written as

$$f(\gamma_2) = \gamma_2^2(\lambda_2 + \mu_2 + r)(\gamma_2 \mu_1(\mu_2 - \lambda_2) + r(\mu - \lambda) P_{1,2}).$$

□

Lemma 2. *Let $\rho < 1$. If $\rho_2 > 1$, then $\xi_{pos} = \gamma_2$ for $r = r_c$, $\xi_{pos} \in (0, \gamma_2)$ for $r > r_c$, and $\xi_{pos} > \gamma_2$ for $r < r_c$. If $\rho_2 < 1$, then $\xi_{pos} \in (0, \gamma_2)$ for $r > 0$.*

Proof. The fact that $\xi_{pos} = \gamma_2$ for $r = r_c$ is immediate from the previous lemma. To prove the remaining two cases, for $\rho_2 > 1$, we define the Sturm chain $p_0(x) = f(x)$, $p_1(x) = f'(x) = 2f_2x+f_1$ and $p_2(x) = p_1(x)q_0(x) - p_0(x) = f_1^2/(4f_2) - f_0$ where $q_0(x)$ is the quotient of the polynomial long division of $p_0(x)$ by $p_1(x)$. Let $\sigma(x)$ be the number of sign changes in the sequence $p_0(x)$, $p_1(x)$, $p_2(x)$. Then, by Sturm's theorem the number of distinct zeros in $(a, b]$ for $a < b$ real is given by $\sigma(a) - \sigma(b)$.

Clearly, $\sigma(0) = 1$ as $p_0(0) = f_0 < 0$ and $p_2(0) = f_1^2/(2f_2) - f_0 > 0$. Further, $\sigma(+\infty) = 0$ as $\lim_{x\to+\infty} p_0(x)$, $p_1(x)$ and $p_2(x)$ is positive. Note, this confirms that we have exactly one positive real root. By the previous lemma we find that when $r \in (0, r_c)$, $p_0(\gamma_2) < 0$ and $p_2(0) = f_1^2/(2f_2) - f_0 > 0$, meaning $\sigma(\gamma_2) = 1$. Therefore the unique positive real root ξ_{pos} is larger than γ_2 by Sturm's theorem. When $r > r_c$ the previous lemma shows that $p_0(\gamma_2) > 0$ and $p_2(0) = f_1^2/(2f_2) - f_0 > 0$, meaning $\sigma(\gamma_2) = 0$ (as it cannot be equal to 2) and the unique root lies in $(0, \gamma_2)$. The argument for $\rho_2 < 1$ is similar. □

Theorem 3. *When $P_{2,2} = 1$ and $\rho_1 \leq \rho_2$ the set of r values for which (1)-(2) has a unique positive fixed point is given by*

$$\Omega_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)} = \left\{ r \,\middle|\, r > r_c = \frac{\mu_1\mu_2(\rho_2 - 1)}{\mu - \lambda} \frac{\gamma_2}{1 - P_{1,1}} \right\}.$$

Proof. Assume $r \leq r_c > 0$, then by Lemma 2 we have $x_{2,2} > \gamma_2$ and by (29) we find that $\pi_{2,0} \leq 0$. Hence, a positive fixed point does not exist when $r \leq r_c$. For $r > r_c > 0$, Lemma 2 and (29) imply that both $x_{2,2}$ and $\pi_{2,0}$ are positive. Further, (32) shows that $0 < \pi_{1,0} < \gamma_1(1 - \rho_1)$ when $\rho_1 < 1$. The fact that $x_{1,2}$ is positive therefore follows from (33). The argument for $r_c \leq 0$ is similar. □

5.2 Push

In this section we present explicit results for the push strategy with $P_{1,1} = 1$. In this particular case the fixed point equations for $\hat{\eta}_k$ and $\hat{\sigma}_k$ become:

$$\hat{\eta}_1 = \hat{r}(\hat{x}_{1,2} + \hat{x}_{2,2}P_{2,1})/\gamma_1,$$

$$\hat{\eta}_2 = \hat{r}\hat{x}_{2,2}P_{2,2}/\gamma_2,$$

$$\hat{\sigma}_1 = \hat{r}\hat{\pi}_{1,0}/\gamma_1,$$

$$\hat{\sigma}_2 = \hat{r}(\hat{\pi}_{1,0}P_{2,1}/\gamma_1 + \hat{\pi}_{2,0}P_{2,2}/\gamma_2).$$

As these equations resemble (25-28) we can use a similar approach as for the pull strategy to establish:

$$\begin{aligned}
\hat{\pi}_{2,0} &= \frac{\gamma_2\mu_2(\gamma_2 - \hat{x}_{2,2})}{\gamma_2(\lambda_2+\mu_2) + P_{2,2}\hat{r}\hat{x}_{2,2}},\\
\hat{\pi}_{1,0} &= \frac{\gamma_1^2(\mu_1-\lambda_1)}{\gamma_1\mu_1 + \hat{r}\hat{x}_{2,2}P_{2,1}},\\
\hat{x}_{1,2} &= \frac{\lambda_1(\gamma_1-\hat{\pi}_{1,0})\gamma_1}{\gamma_1\mu_1+\hat{r}\hat{\pi}_{1,0}}.
\end{aligned}$$

While $\hat{x}_{2,2}$ is now the solution of the quadratic equation $\hat{f}(x) = \hat{f}_2x^2 + \hat{f}_1x + \hat{f}_0 = 0$ with

$$\begin{aligned}
\hat{f}_2 &= \hat{r}(\hat{r}(\mu-\lambda)P_{2,2} + \gamma_2\mu_2^2)P_{2,1},\\
\hat{f}_1 &= \gamma_1\gamma_2\mu_1\mu_2^2 + \gamma_2\hat{r}(\gamma_1(\mu_2+\lambda_2)(\mu_1-\lambda_1)P_{2,1} + \gamma_1\mu_1(\mu_2-\lambda_2)P_{2,2} - \gamma_2\lambda_2^2P_{2,1}),\\
\hat{f}_0 &= -\gamma_1\gamma_2^2\lambda_2^2\mu_1.
\end{aligned}$$

that has a unique positive root denoted as

$$\hat{\xi}_{pos} = \frac{\sqrt{\hat{f}_1^2 - 4\hat{f}_2\hat{f}_0} - \hat{f}_1}{2\hat{f}_2}. \tag{35}$$

In this case one can verify that

$$\hat{f}(\gamma_2) = \gamma_2^2(\lambda_2+\mu_2+\hat{r}P_{2,2})(\gamma_1\mu_1(\mu_2-\lambda_2) + \hat{r}(\mu-\lambda)P_{2,1}),$$

which allows us to prove the following result in a manner similar to Theorem 3

Theorem 4. *When $P_{1,1} = 1$ and $\rho_1 \le \rho_2$ the set of r values for which* (3)-(4) *has a unique positive fixed point is given by*

$$\hat{\Omega}_{(\lambda_1,\lambda_2,\mu_1,\mu_2,\gamma_1,P)} = \left\{ \hat{r} \,\middle|\, \hat{r} > \hat{r}_c = \frac{\mu_1\mu_2(\rho_2-1)}{\mu-\lambda}\frac{\gamma_1}{1-P_{2,2}} \right\}.$$

6 Performance measures

In order to compute the main performance measures using the mean field model we first compute a fixed point of the set of ODEs. For some specific cases (see Section 5) we have an explicit expression for the unique fixed point. However, in general we rely on an iterative procedure, which is presented in Algorithm 1 for the pull strategy. Numerical experiments on thousands of randomly generated input parameters suggest that this iterative method converges to a positive fixed point if the condition on r in Theorem 1 (or 2) is met and the convergence is monotone.

```
Input: $r, \gamma_1, \gamma_2, \mu_1, \mu_2, \lambda_1$ and $\lambda_2$
Output: $\pi_{k,0}$ and $x_{k,2}$, for $k = 1, 2$
for $k = 1$ to $2$ do
    $\pi_{k,0} = \gamma_k$; $\pi_{k,0}^{(old)} = 1$; $x_{k,2} = 0$;
end
while $\sum_k |\pi_{k,0} - \pi_{k,0}^{(old)}| > 10^{-14}$ do
    for $k = 1$ to $2$ do
        $\pi_{k,0}^{(old)} = \pi_{k,0}$;
        $\eta_k = r \sum_{k'=1}^{2} \frac{x_{k',2}}{\gamma_{k'}} P_{k,k'}$;
        $\sigma_k = \frac{r}{\gamma_k} \sum_{k'=1}^{2} \pi_{k,0} P_{k',k}$;
    end
    for $k = 1$ to $2$ do
        $\pi_{k,0} = \gamma_k / (1 + \frac{\lambda_k + \nu_k}{\mu_k(1 - \lambda_k/(\mu_k + \sigma_k))})$;
        $x_{k,2} = \pi_{k,0} \lambda_k (\lambda_k + \nu_k) / (\mu_k(\mu_k + \sigma_k - \lambda_k))$;
    end
end
```

Algorithm 1: Computes $\pi_{k,0}$ and $x_{k,2}$, for $k = 1, 2$, for *pull* strategy

The iterative procedure determines $\pi_{k,0}$ and $x_{k,2}$, for $k = 1, 2$, from which all the remaining entries of x follow due to (11) and the mean type k queue length is given by

$$E[Q_k] = \frac{(\lambda_k + \eta_k)(\mu_k + \sigma_k)^2}{(\mu_k + \sigma_k - \lambda_k)((\mu_k + \eta_k)(\mu_k + \sigma_k) + \lambda_k \sigma_k)}, \tag{36}$$

while the mean response time is found via Little's formula. For the special cases discussed in Section 5 we can obtain explicit expressions for the mean type 1 and type 2 queue length. We only present the expressions for the type 1 queue length as the expressions for the type 2 queue length we obtained appear to be far less elegant.

Theorem 5. *For the pull strategy with $P_{2,2} = 1$ and $\rho_1 < 1$ we have*

$$E[Q_1] = \frac{1}{(1-\rho_1)} \frac{(\gamma_2 \lambda_1 + s)}{(\gamma_2 \mu_1 + s)} \frac{(\gamma_2 \mu_1 + \gamma_2 (1-\rho_1) r P_{1,1} + s)}{(\gamma_2 \mu_1 + \gamma_2 r P_{1,1} + s)} \leq \frac{1}{1-\rho_1},$$

where $s = r \xi_{pos} P_{1,2}$ and ξ_{pos} is given by (34). For the push strategy with $P_{1,1} = 1$ and $\rho_1 < 1$ we have

$$E[\hat{Q}_1] = \frac{1}{(1-\rho_1)} \frac{(\gamma_1 \lambda_1 + \hat{s})}{(\gamma_1 \mu_1 + \hat{s})} \frac{(\gamma_1 \mu_1 + \gamma_1 (1-\rho_1) \hat{r} + \hat{s})}{(\gamma_1 \mu_1 + \gamma_1 \hat{r} + \hat{s})} \leq \frac{1}{1-\rho_1},$$

where $\hat{s} = \hat{r} \hat{\xi}_{pos} P_{2,1}$ and $\hat{\xi}_{pos}$ is given by (35).

Proof. The result can be obtained from (36) after plugging in the explicit expressions obtained in the previous Section. □

There are a few things we can remark with respect to the expression for $E[\hat{Q}_1]$ (or $E[Q_1]$) in the above theorem. First, if $P_{2,1} = 0$ (or $P_{1,2} = 0$ for the pull strategy), the mean queue length becomes $\rho_1/(1-\rho_1)\cdot(\mu_1+(1-\rho_1)\hat{r})/(\mu_1+\hat{r})$ which coincides with the expression of the mean queue length in a homogeneous system [10], which was identical for both strategies. Second, even if the type 2 queues are heavily overloaded and probe at a high rate, $E[\hat{Q}_1]$ is bounded by $1/(1-\rho_1)$. This can be understood by noting that (i) $1/(1-\rho_1)$ is equal to the mean queue length of an M/M/1 queue with load ρ_1 plus 1, (ii) at any point in time there is at most one type 2 job in any type 1 queue and (iii) due to the exponential service times the type 1 queue length distribution is the same as in a system where the type 1 jobs always get preemptive priority over the type 2 jobs.

At this point we should emphasize that the iterative procedure takes r or $\hat{r}$ as an input parameter, meaning we can compute the mean response time for a given r or $\hat{r}$. Sometimes we are however interested in the mean response time given R, which is the mean number of probes that a server is allowed to transmit per time unit. To determine the r or $\hat{r}$ that matches a predefined R we can make use of the following equations:

$$R_{pull} = r(\pi_{1,0} + \pi_{2,0}), \tag{37}$$

$$R_{push} = \hat{r}(\hat{x}_{1,2} + \hat{x}_{2,2}), \tag{38}$$

as under the pull strategy empty servers probe, while under the push strategy servers with at least 2 jobs probe. To determine the proper r or $\hat{r}$ value we use a bisection algorithm until the computed R matches the predefined R. We note that, as in the homogeneous case [10], arbitrarily large $\hat{r}$ values can be selected for the push strategies without exceeding R for low loads. We examine such cases in more detail in Appendix B.

Theorem 1 (or 2) provided a condition on r (or $\hat{r}$) for the existence of a unique fixed point (under Conjecture 2) and we may wonder how this condition can be expressed in terms of R using (37) and (38). For the pull strategy we have $\pi_{2,0} = 0$ in the dominating system, so (22) can be used to find

$$R_{pull} > \frac{\gamma_2(\lambda_2 - \mu_2)}{(1 - P_{1,1})}. \tag{39}$$

This expression is intuitively clear: if $\lambda_2 > \mu_2$ then $\gamma_2(\lambda_2 - \mu_2)$ represents the mean amount of work that needs to transferred per time unit, while $R_{pull}(1-P_{1,1})$ is the mean number of successful probes per time unit under the pull strategy (in the dominating system).

For the push strategy we have due to (16) (with hats added in the appropriate places)

$$R_{push} = \hat{r}\left(1 - \hat{\pi}_{1,0}\left(1 + \frac{\lambda_1 + \hat{\eta}_1}{\mu_1}\right)\right),$$

Case	N_1	N_2	$P_{1,1}$	$P_{2,2}$	λ_1	λ_2	μ_1	μ_2	r_c	$\hat{r}_c$
1	30	15	1/2	2/3	1.5	1.25	2	1	4/3	4
2	50	50	7/8	1/4	0.8	1.1	1	1	8	4/3
3	5	15	1/10	0	0.2	3.2	1	3	10/3	1

Table 1: Parameter settings of the three random cases to validate the accurary of the model for predicting stability.

in the dominating system as $\hat{x}_{2,2} = \gamma_2$. Using (42), we obtain the following condition on R_{push}:

$$R_{push} > \frac{\gamma_1\gamma_2\mu_1 P_{1,1}(\lambda_2-\mu_2)^2 + P_{2,1}\gamma_1(\lambda_2-\mu_2)\left[\gamma_1\lambda_1^2 + \gamma_2(\mu_1^2 + \lambda_1(\lambda_2-\mu_2))\right]}{P_{2,1}(P_{2,1}\mu_1 + P_{1,1}(\lambda_2-\mu_2))(\mu-\lambda)}.$$

This condition does not appear to have a simple intuitive explanation, which might be due to the fact that both the overloaded type 2 and the underloaded type 1 queues probe under the push strategy.

If we adapt the push strategy such that *only* the overloaded type 2 queues are allowed to probe (at rate $\hat{r}$), one still finds the same condition $\hat{r} > \hat{r}_c$, but now R_{push} becomes $\hat{r}\gamma_2$ in the dominating system. Hence, the condition for R_{push} becomes

$$R_{push} > \gamma_2(\lambda_2-\mu_2)\frac{\gamma_1\mu_1}{(\mu-\lambda)P_{2,1}},$$

which can be understood intuitively as $R_{push}P_{2,1}$ is the rate at which probes are sent to the type 1 queues and

$$1 - \frac{\lambda_1}{\mu_1} - \frac{\gamma_2}{\gamma_1}\frac{(\lambda_2-\mu_2)}{\mu_1} = \frac{\mu-\lambda}{\gamma_1\mu_1},$$

is the probability that such a probe finds an empty type 1 queue.

7 Mean field model validation

We validate the mean field model with two types of experiments. First, we look at the accuracy of the model to predict the stability of the system. For this purpose, we consider three randomly chosen cases such that the critical r value, that is, r_c for the push and $\hat{r}_c$ for the pull strategy, is a simple fraction. Table 1 lists the parameter settings of these three cases, where N_i is the number of type i queues used in the simulation experiment, for $i = 1, 2$. Note, in each of these cases the total number of queues $N_1 + N_2$ is at most 100, the overall load ρ is below one, but the type 2 queues are overloaded.

For each of these three cases we simulated its corresponding (uniformized) Markov chain for $4 \cdot 10^9$ events for both the pull and push strategy for two choices of $\hat{r}$ (and r): $1.01\hat{r}_c$ and $0.99\hat{r}_c$. We subsequently plotted the evolution of the average type 2 queue length as a function of time.

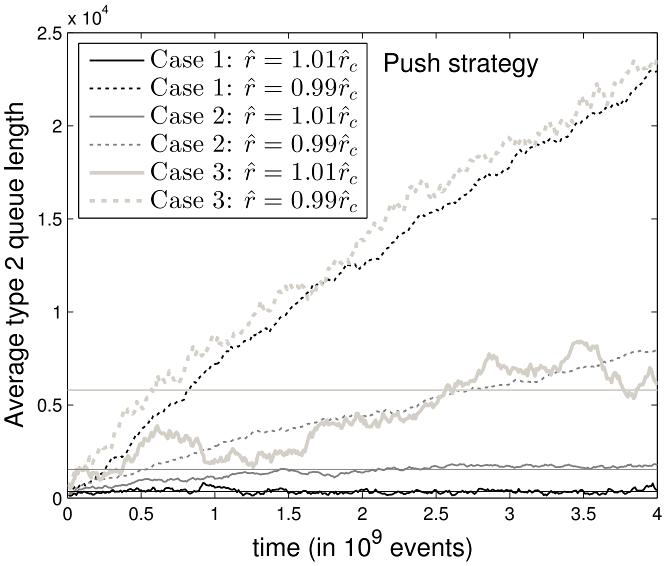

Figure 1: Evolution of the average type 2 queue length for the three randomly selected cases presented in Table 1.

Case	γ_1	$P_{1,1}$	$P_{2,2}$	λ_1	λ_2	μ_1	μ_2	r	$\hat{r}$
1	0.5	0.2	0.5	0.5	0.7	1	0.5	1	1.5
2	0.74	0.8	0.7	0.5	0.6	0.8	0.4	2.4	3
3	0.8	0.3	0.3	0.2	0.4	0.8	0.3	0.1	0.25
4	0.2	0.4	0.6	0.3	0.8	0.6	0.75	2.5	1
5	0.5	0.5	0.1	0.3	1.1	1.2	0.7	2	1.5
6	0.74	0.6	0.7	1.1	0.6	1.4	0.7	0.5	0.5
7	0.4	0.5	0.5	0.7	1.1	0.9	1.2	0.3	0.3
8	0.9	0.1	0.5	1	1	1.5	1.3	1	1
9	0.6	0.3	0.4	0.1	0.3	0.11	0.31	0.5	0.5
10	0.26	0.6	0.7	0.4	0.5	1.1	0.8	0.4	0.4

Table 2: Parameter settings of the 10 random cases to validate the accuracy of the mean field model to predict mean queue lengths.

Figure 1 depicts the evolution of the average type 2 queue length for the push strategy. If the predicted stability region of the model is accurate the type 2 queue length should stabilize when $\hat{r}$ exceeds $\hat{r}_c$ and should grow without bound otherwise. Figure 1 confirms that this is the case for the three randomly selected cases. We also included the mean type 2 queue length as predicted by the mean field model in case $\hat{r} > \hat{r}_c$ and the simulation results appear to be in agreement with the mean field model (though it is hard to make any trustworthy statements regarding its accuracy as the system is almost unstable and therefore extremely long simulation experiments would be required). Similar findings were obtained in case of the pull strategy.

In order to check the accuracy of the mean field model to predict the mean type 1 and 2 queue length, we selected 10 arbitrary cases, the parameters of which are listed in Table 2. It is worth

			Pull			Push	
Case	Type	N = 50	N = 250	ODE	N = 50	N = 250	ODE
1	1	1.3388	1.3361	1.3355	1.2963	1.2928	1.2919
	2	17.9042	17.4006	17.2621	25.8720	24.9622	24.7607
2	1	1.1711	1.1561	1.1524	1.0947	1.0805	1.0770
	2	3.1136	2.9405	2.8976	7.7400	7.2223	7.0916
3	1	0.3871	0.3870	0.3870	0.3712	0.3710	0.3710
	2	4.3457	4.3216	4.3161	14.8403	14.7941	14.7581
4	1	1.4318	1.4104	1.4057	1.5950	1.5840	1.5816
	2	77.3862	72.9694	73.1875	93.3942	91.7496	91.2499
5	1	0.7220	0.7202	0.7198	0.7658	0.7644	0.7640
	2	80.0848	77.2885	77.1264	9.2329	9.0745	9.0333
6	1	3.1671	3.1562	3.1535	3.0035	2.9923	2.9895
	2	2.5015	2.4620	2.4521	3.9858	3.9265	3.9119
7	1	3.0152	3.0012	2.9983	3.1413	3.1262	3.1238
	2	8.3764	8.3573	8.3539	8.0452	7.8179	7.8102
8	1	1.9239	1.9216	1.9212	1.7209	1.7169	1.7160
	2	0.7637	0.7560	0.7543	2.7056	2.6784	2.6727
9	1	4.4713	4.3696	4.3430	3.6753	3.5627	3.5337
	2	7.5042	7.2358	7.1715	8.4652	8.1942	8.1250
10	1	0.5375	0.5357	0.5353	0.5969	0.5949	0.5944
	2	1.3209	1.3161	1.3150	1.2306	1.2264	1.2254

Table 3: Average type 1 and 2 queue lengths: simulation vs. mean field model.

noting that the type 2 queues are overloaded in the 5 first cases, but r and $\hat{r}$ are set such that the system is stable. We compare the mean queue lengths as predicted by the mean field model with simulation results for a system consisting of $N = 50$ and $N = 250$ queues in Table 3. The simulation results were obtained based on 5 runs each with a length of $5 \cdot 10^6 \cdot (\sum_k N_k(r + \lambda_k + \mu_k))$ events and a warm-up period of 20%.

Looking at Table 3 it is fair to state that the mean field model is quite accurate, that is, the relative error is always well below 10% for $N = 50$ queues and below 2% for $N = 250$ queues. Thus, the results become more accurate as the number of queues increases from $N = 50$ to $N = 250$ and the mean field model typically provides an underestimation of the mean queue lengths for finite N. These simulation experiments required several hours to complete, while the mean field model generated results within seconds.

8 Numerical Examples

In this section we present some numerical results for a system consisting of a set of fast (type 1) and a set of slow (type 2) servers, that is, $\mu_1 \geq \mu_2$. Without loss of generality we set $\mu = \gamma_1\mu_1 + \gamma_2\mu_2 = 1$, meaning the load $\rho = \lambda$ and $\mu_2 \leq 1 \leq \mu_1$.

We consider 3 pull and push strategies that differ in the manner in which they select the server that is probed, that is, they rely on a different P matrix:

1. $P_{1,1} = P_{2,1} = \gamma_1$ (Pull Uniform): Any server can be probed by any other server, with equal probability.

2. $P_{1,1} = \gamma_1, P_{2,2} = 1$ (Pull Slower): A server is not allowed to probe a faster server: type 1 servers can probe any server (with equal probability), type 2 servers can only probe type 2 servers.

3. $P_{1,2} = P_{2,2} = 1$ (Pull Slow): A server is only allowed to probe a type 2 server (i.e., a slow server).

4. $P_{1,1} = P_{2,1} = \gamma_1$ (Push Uniform): Same as pull Uniform, but for the push strategy.

5. $P_{1,1} = 1, P_{2,2} = \gamma_2$ (Push Faster): A server is not allowed to probe a slower server.

6. $P_{1,1} = P_{2,1} = 1$ (Push Fast): A server is only allowed to probe type 1 servers (i.e., a fast server).

It is worth remarking that we can make use of the explicit expressions derived in Section 5 for 4 of the above 6 strategies, the two exceptions being the Pull and Push Uniform strategies.

In a first set of experiments we set $\lambda_1 = \lambda_2 = \lambda$. When comparing these six strategies we set r and $\hat{r}$ such that the average probe rate matches the predefined R_{pull} and R_{push}. We let $\rho \in (0, 1)$ (as $\rho > 1$ implies instability for all strategies) and let μ_1/μ_2 vary between 1 and 3 (where $\mu_1 = \mu_2$ corresponds to the homogeneous case). Note the ratio μ_1/μ_2 is a measure for the heterogeneity of the servers in the network. We further note that according to the mean field model all the pull strategies are stable in this entire range of ρ and μ_1/μ_2 values when $R_{pull} = 1$. For instance, for the pull slow strategy demands that R_{pull} exceeds $\gamma_2(\lambda - \mu_2)$ which is less than one (see (39)). The push strategies on the other hand are not stable in this entire range of ρ values.

Figure 2 depicts the $(\rho, \mu_1/\mu_2)$ combinations for which each of the 6 strategies outperforms the other 5. As expected, the pull strategies are superior for large loads. The uniform pull and push strategies are best when the server speeds are close to each other, while the pull slow and push fast are best when the server speeds differ a lot (the region where the pull slow is best when $\gamma_1 = 0.5$ starts when μ_1/μ_2 is approximately 3). This can be understood by noting that both these strategies attempt to move jobs to the faster, less heavily loaded servers only. This figure also indicates that the load required for the pull strategies to outperform the push strategies is not very sensitive to the server heterogeneity when $\lambda_1 = \lambda_2$ (especially for $\gamma_1 = 0.5$). When

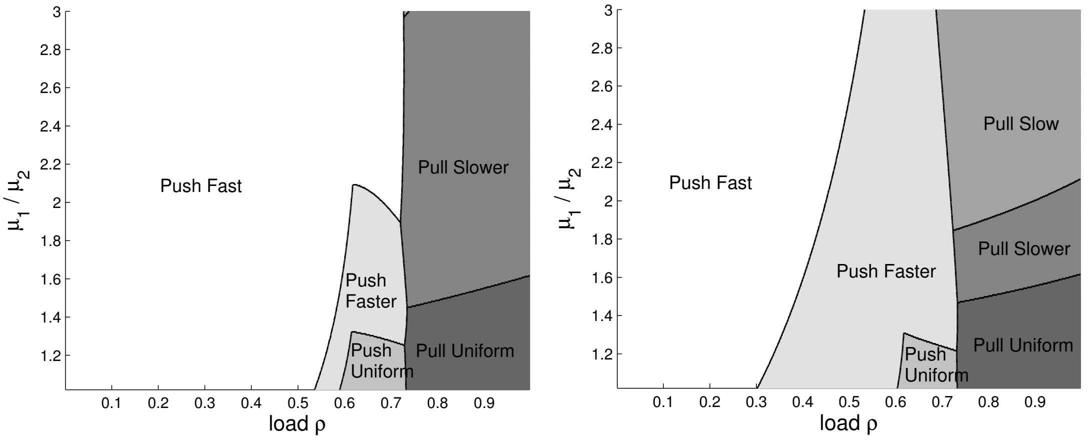

Figure 2: The different areas identify the $(\rho, \mu_1/\mu_2)$ combinations for which each of the 6 strategies outperforms the remaining 5 strategies when $R_{pull} = R_{push} = 1$. The fraction of fast servers γ_1 is equal to 0.5 (left) and 0.1 (right). In this case $\lambda_1 = \lambda_2$ and $\mu = \gamma_1\mu_1 + \gamma_2\mu_2 = 1$.

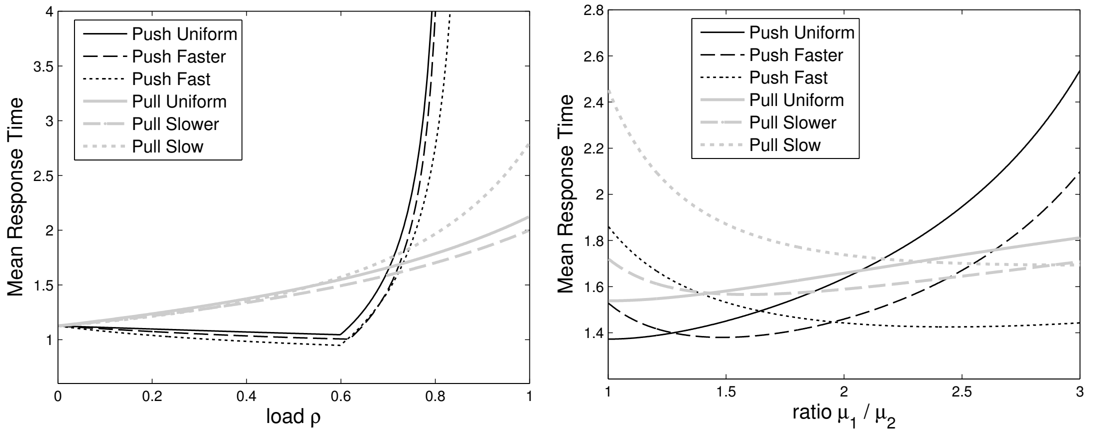

Figure 3: Mean response time as a function of the load ρ for $\mu_1/\mu_2 = 2$ (left) and as a function of μ_1/μ_2 for $\rho = 0.7$ (right) with $R = 1$ and $\gamma_1 = 0.5$. In this case $\lambda_1 = \lambda_2$ and $\mu = \gamma_1\mu_1 + \gamma_2\mu_2 = 1$.

comparing the results for $\gamma_1 = 0.5$ and $\gamma_1 = 0.1$, we see that in the latter case both the Pull Slow and Push Faster strategies outperform the others for a much larger range of $(\rho, \mu_1/\mu_2)$. Indeed when $\gamma_1 = 0.1$ only 10% of the servers is fast and 90% is slow, as such pushing all the jobs to the fast servers only is less effective (unless the load is low), while pulling jobs from the slow servers only does become more attractive if more servers are slow.

Figure 3 gives an impression of the behavior of the mean response times for the 6 strategies considered when we fix either the ratio μ_1/μ_2 or the load ρ. The left figure indicates that the mean response times of the push strategies are initially close to 1 and decrease as a function of the load. This is the region where the jobs under the push strategies do not require any queueing (see Appendix B). The decrease in the mean response time is caused by the fact that more jobs

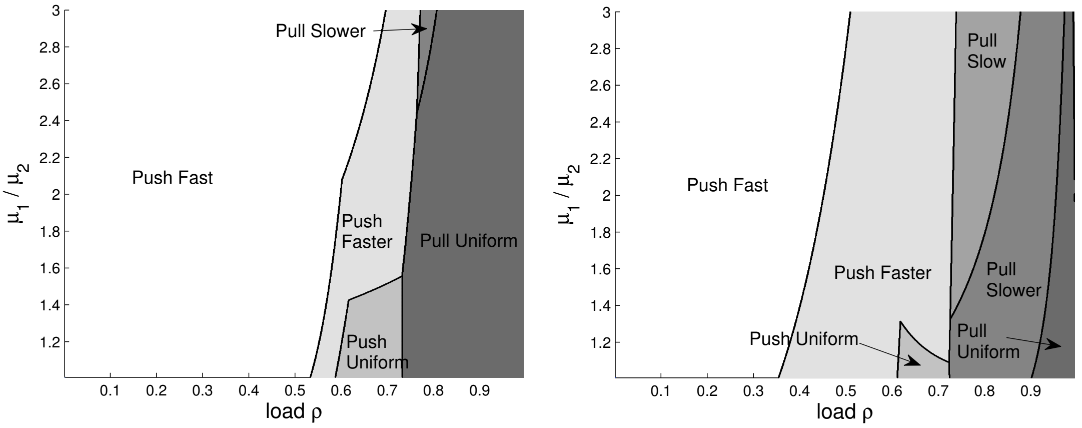

Figure 4: The different areas identify the $(\rho, \mu_1/\mu_2)$ combinations for which each of the 6 strategies outperforms the remaining 5 strategies when $R_{pull} = R_{push} = 1$. The fraction of fast servers γ_1 is equal to 0.5 (left) and 0.1 (right). In this case $\rho_1 = \rho_2$ and $\mu = \gamma_1\mu_1 + \gamma_2\mu_2 = 1$.

are executed on the fast servers and is therefore also the most pronounced for the Push Fast strategy. As for the pull strategies we remark that even though the system is unstable for $\rho = 1$, the mean response time remains bounded as the load ρ approaches one (as in the homogeneous case, see [10]). The figure on the right indicates that whether increasing the server heterogeneity μ_1/μ_2 increases the mean response times very much depends on μ_1/μ_2 and the strategy under consideration.

In a second set of experiments we used the same setup as in the first, but this time we set the arrival rates such that all the servers have the same load, that is, $\lambda_1/\mu_1 = \rho = \lambda_2/\mu_2$. This implies that all of the 6 strategies are stable for $\rho < 1$. Figure 4 depicts the areas in which each of the 6 strategies outperforms the other 5 for $R_{pull} = R_{push} = 1$ for $\gamma_1 = 0.5$ and 0.1. The regions for the push strategies are similar in shape as in the first set of experiments (see Figure 2) and the load at which the pull strategies outperform the push strategies is still not very sensitive with respect to μ_1/μ_2.

However, the regions for the pull strategies are now quite different in shape. When $\gamma_1 = 0.5$ the pull uniform is best for all loads above 0.8 in the range of μ_1/μ_2 values considered. This is probably due to the fact that having equal loads makes it less attractive to pull jobs from a subset of the servers only. Nevertheless, Figure 4 shows that if only a limited fraction of the servers is fast, i.e., 10%, it is better to pull jobs from the slow servers only, unless the load is close to one.

9 Conclusion

In this paper we studied a class of rate-based pull and push strategies in a large heterogeneous setting and proposed a simple formula for the required probe rate to achieve system stability in case of two types of servers. For some specific cases we also derived explicit expressions for the unique positive fixed point which can be used to express the mean queue lengths. For the general case a simple iterative algorithm was introduced to compute a fixed point. We compared the performance of 6 specific pull and push strategies in the presence of a set of fast and slow servers and identified the regions where each of these strategies outperforms the others in terms of the mean response time. Possible subjects for future work include considering more general rate-based strategies, proving the conjectures used to derive the proposed stability condition, proposing stability conditions and explicit results in the presence of more than two classes of servers, etc.

References

[1] M. Benaïm and J. Le Boudec. On mean field convergence and stationary regime. *CoRR*, abs/1111.5710, Nov 24 2011.

[2] C. Bordenave, D. McDonald, and A. Proutiére. Performance of random medium access control, an asymptotic approach. In *ACM Sigmetrics*, pages 1–12, New York, NY, USA, 2008. ACM.

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

[4] D. Eager, E. Lazowska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *Perform. Eval.*, 6(1):53–68, 1986.

[5] S. Ethier and T. Kurtz. *Markov processes: characterization and convergence.* Wiley, 1986.

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

[7] T. Kurtz. *Approximation of population processes.* Society for Industrial and Applied Mathematics, 1981.

[8] W. Minnebo and B. Van Houdt. Pull versus push mechanism in large distributed networks: Closed form results. In *Proc. of the 24-th International Teletraffic Congress*, Krakau (Poland), 2012.

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

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

[11] R. Mirchandaney, D. Towsley, and J. A. Stankovic. Adaptive load sharing in heterogeneous distributed systems. *J. Parallel Distrib. Comput.*, 9(4):331–346, 1990.

[12] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California, Berkeley, 1996.

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

[14] W. Szpankowski. Stability conditions for some multiqueue distributed systems: Buffered random access systems. *Advances in Applied Probability*, 26:498–515, 1994.

A Proof of Theorem 2

For the push strategy the equation for $\hat{\eta}_1$ in the dominating system becomes

$$\hat{\eta}_1 = \hat{r}\left(\hat{x}_{1,2}P_{1,1}/\gamma_1 + \gamma_2 P_{2,1}/\gamma_1\right). \tag{40}$$

The main thing to note is that $\hat{\eta}_1$ is still a function of $\hat{x}_{1,2}$ only, so the same argument used to prove Theorem 1 can be used to find

$$\hat{x}_{1,2} = \frac{\gamma_1\lambda_1\left(\gamma_1\lambda_1 + \hat{r}\gamma_2(1-P_{2,2})\right)}{\gamma_1\mu_1^2 + \hat{r}(\gamma_2\mu_1(1-P_{2,2}) + \gamma_1(\mu_1-\lambda_1)P_{1,1})}, \tag{41}$$

and

$$\hat{\pi}_{1,0} = \frac{\gamma_1^2(\mu_1-\lambda_1)}{\gamma_1\mu_1 + \hat{r}\gamma_2(1-P_{2,2})}. \tag{42}$$

For the push strategy Conjecture 2 implies that we have a unique positive fixed point if and only if

$$\lambda_2 < \mu_2 + \hat{r}P_{2,1}\frac{\hat{\pi}_{1,0}}{\gamma_1}, \tag{43}$$

as a type 2 queue probes a type 1 queue at rate $\hat{r}P_{2,1}$ and this queue is empty with probability $\hat{\pi}_{1,0}/\gamma_1$. Plugging in the expression for $\hat{\pi}_{1,0}$ completes the proof.

B Push strategies: no queueing

In this section we look at how large R_{push} should be such that the push strategy can make use of any r value without exceeding R_{push}. Note if $\hat{r}$ can be selected arbitrarily large, jobs no longer experience any queueing delay and their response time is equal to the processing time (which depends on the server that executes the job).

Let $\hat{x}_{1,k}$ be the probability that a server of type k is busy at any given moment and assume $\hat{r}$ is infinitely large. The probability that a server of type k starts probing when a new arrival occurs is thus $\hat{x}_{k,1}$. A probe sent from a type k server is successful with probability $P_{k,1}(1-\hat{x}_{1,1}) + P_{k,2}(1-\hat{x}_{2,1})$, meaning on average $\frac{1}{1-P_{k,1}\hat{x}_{1,1}-P_{k,2}\hat{x}_{2,1}}$ probes are needed until one is successful. This implies that

$$R_{push} \geq \frac{\gamma_1\lambda_1\hat{x}_{1,1}}{1-P_{1,1}\hat{x}_{1,1}-P_{1,2}\hat{x}_{2,1}} + \frac{\gamma_2\lambda_2\hat{x}_{2,1}}{1-P_{2,1}\hat{x}_{1,1}-P_{2,2}\hat{x}_{2,1}}, \tag{44}$$

is required to avoid queueing. Next we set up a system of equations to determine the unknowns $\hat{x}_{1,1}$ and $\hat{x}_{2,1}$.

The rate of type k jobs that are served locally is clearly given by $\lambda_k(1-\hat{x}_{k,1})$, while the rate of type k' jobs that are pushed to a type k server, with $k, k' \in \{1,2\}$, can be written as

$$\lambda_{k'}\hat{x}_{k',1}\frac{P_{k',k}(1-\hat{x}_{k,1})}{1-P_{k',1}\hat{x}_{1,1}-P_{k',2}\hat{x}_{2,1}}.$$

The equations used to determine the unknowns $\hat{x}_{1,1}$ and $\hat{x}_{1,2}$ can now be obtained by noting that the rate at which the fraction of type k servers are working should match the sum of the rate of type k jobs that are served locally plus the rate of type k' jobs served by a type k server that originated in another server (possibly of the same type):

$$\gamma_k\mu_k\hat{x}_{k,1} = (1-\hat{x}_{k,1})\left(\gamma_k\lambda_k + \sum_{k'=1}^{2}\frac{\gamma_{k'}\lambda_{k'}\hat{x}_{k',1}P_{k',k}}{1-P_{k',1}\hat{x}_{1,1}-P_{k',2}\hat{x}_{2,1}}\right), \tag{45}$$

for $k=1,2$. Further, as any incoming job needs to be processed somewhere we have

$$\gamma_1\mu_1\hat{x}_{1,1}+\gamma_2\mu_2\hat{x}_{2,1}=\lambda.$$

We now present some explicit results for $\hat{x}_{1,1}$ and $\hat{x}_{2,1}$ in some special case.

Case 1: When $P_{1,1}=P_{2,1}=1$ the system of equations given by (45) can be solved to find

$$\begin{aligned}\hat{x}_{1,1} &= \rho_1 + \frac{\gamma_2\mu_2}{\gamma_1\mu_1}(\rho_2-\hat{x}_{2,1}),\\ \hat{x}_{2,1} &= \frac{\rho_2}{1+\rho_2},\end{aligned}$$

where $\rho_2=\lambda_2/\mu_2$. Hence, due to (44) we find

$$R_{push} \geq \gamma_2\lambda_2\rho_2 + \frac{\gamma_1\lambda_1}{1-\rho_1}\left(\rho_1 + \frac{\gamma_2\lambda_2\rho_2}{\gamma_1(\mu_1-\lambda_1)(1+\rho_2)-\gamma_2\lambda_2\rho_2}\right).$$

This result implies that as long as $\gamma_2\lambda_2\rho_2 < \gamma_1(1+\rho_2)(\mu_1-\lambda_1)$ one can set R_{push} such that jobs are never queued. This condition is equivalent to demanding that $\hat{x}_{1,1}$ is less than 1.

Case 2: When $P_{1,1}=1$ and $P_{2,2}=\gamma_2$ the system of equations given by (45) has two solutions and the solution with $\hat{x}_{2,1}\in(0,1)$ is given by

$$\begin{aligned}\hat{x}_{1,1} &= \rho_1 + \frac{\gamma_2\mu_2}{\gamma_1\mu_1}(\rho_2-\hat{x}_{2,1}),\\ \hat{x}_{2,1} &= \frac{1+\rho_2-\sqrt{(1+\rho_2)^2-4\gamma_2\rho_2}}{2\gamma_2},\end{aligned}$$

where the expression for $\hat{x}_{1,1}$ in terms of $\hat{x}_{2,1}$ is identical to case 1.