

# A Routing Policy for the X Call-Center Model Designed to Respond to Unexpected Overloads

by

Ohad Perry and Ward Whitt

IEOR Department  
Columbia University  
{op2105, ww2040}@columbia.edu

## *Abstract*

We consider a large-scale  $X$  call-center model – a stochastic model with two customer classes and two associated service pools containing large numbers of agents, with each pool primarily dedicated to the designated customer class, but with all agents cross-trained and allowed to serve the other class, even though they may do so inefficiently or ineffectively. Under normal loads, we want class- $i$  customers to be served by type- $i$  agents, but we activate sharing (serving the other class) when there is an unexpected overload, allowing sharing in only one way at any time. We propose a *fixed-queue-ratio assignment rule with thresholds* (FQR+T) for available agents. Assignments depend on a weighted-difference stochastic process:  $D(t) \equiv Q_1(t) - rQ_2(t)$ , where  $Q_i(t)$  is the queue length of class- $i$  customers and  $r$  is a weighting factor, which management can set. Provided that there is no current sharing in the other direction, an available agent in service pool 2 (1) serves the first class-1 (2) waiting customer when  $D(t) \geq \kappa_{1,2}$  ( $\leq -\kappa_{2,1}$ ), where  $\kappa_{1,2}$  and  $\kappa_{2,1}$  are threshold parameters that management can set. We develop approximations to describe system performance when overloads occur, and perform simulations to verify that the approximations are effective.

March 13, 2008



## 1. Introduction

**Unexpected Overloads.** In this paper we propose a family of routing policies for assigning customers to agents (servers) in call centers and other large-scale service systems, designed to automatically respond to unexpected overloads whenever they occur by activating sharing.

In a typical call center, under normal circumstances, the arrival rates vary by time of day in a predictable way, and the staffing responds to that anticipated pattern, typically with fixed staffing levels over short time periods, such as half hours; see Gans et al. (2003) and Aksin et al. (2007) for background. In addition, there are fluctuations about the arrival rates, so that the overall arrival process is well modelled by a nonhomogeneous Poisson process. Ways to apply stationary stochastic models (in a time-dependent way) to staff with the time-varying demand have been developed; see Green et al. (2007).

In this paper we are concerned with deviations from that familiar pattern. We are thinking that the arrival rates will usually be near their forecasted levels, but occasionally, for various reasons, there will be unforeseen surges in demand (or unavailable service), going significantly beyond the usual fluctuations. A demand surge might occur because of a catastrophic event in emergency response, an intense television advertising campaign in retail, or a system failure experienced by an alternative service provider. Such unexpected demand surges typically cause congestion that cannot be eliminated entirely. Our goal is to help reduce that congestion by activating sharing (help from less loaded agents).

**The X Model.** In this paper we restrict attention to the  $X$  model, depicted in Figure 1. The  $X$  model has two customer classes and two agent pools. We assume that each customer class has a service pool primarily dedicated to it, but all agents are cross-trained, so that they can handle calls from the other class, even though they may do so inefficiently or ineffectively. Under normal loading, we want each class to be served by its designated agents, but we want to allow sharing when there are unexpected unbalanced overloads, either when only one class is overloaded or when both classes are overloaded but one is much more overloaded than the other. In some service systems, one customer class is considered more important, so that we may want stronger sharing in one direction than the other. Thus we want a

control that can separately control the extent of sharing in each direction. We may even want to allow sharing in only one direction.

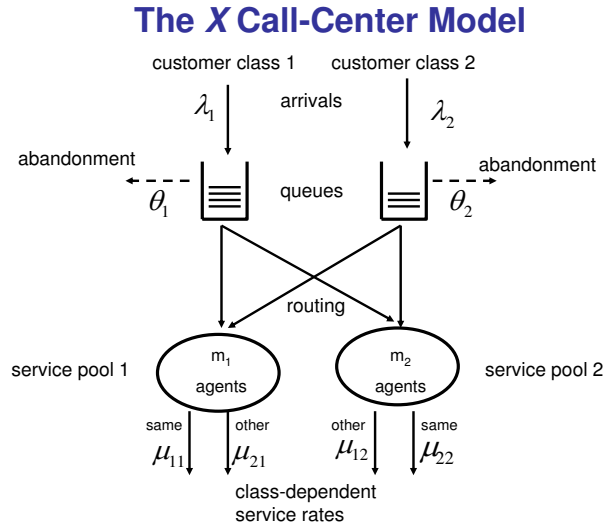


Figure 1: The  $X$  model

Even though we only consider the  $X$  model, we are also interested in much more complex scenarios. For example, there might be two separate multi-class multi-pool call centers, possibly located on different continents, each with a traditional form of skill-based routing. We might then want one call center to help the other when it is overloaded (in the unbalanced way), but not otherwise. In addition to analyzing the  $X$  model for its own sake, we regard it as an idealization of this more general problem. Thus, we provide insight into this more general problem as well.

**FQR Routing.** We suggest a modification of the *fixed-queue-ratio* (FQR) *routing rule* for multi-class multi-pool systems proposed and analyzed by Gurvich and Whitt (2007). With two queues, FQR can be implemented by considering a (weighted) *queue-difference stochastic process*

$$D(t) \equiv Q_1(t) - rQ_2(t), \quad t \geq 0, \quad (1.1)$$

where  $Q_i(t)$  is the class- $i$  queue length at time  $t$  and  $r$  is a target-ratio parameter that management can set. With FQR for the  $X$  model, a newly available agent in either service pool serves the customer at the head of the class-1 queue if  $D(t) > 0$ , and serves the customer at the head of the class-2 queue otherwise. The goal of FQR is to maintain a nearly constant queue ratio:  $Q_1(t)/Q_2(t) \approx r$  throughout time.

The FQR control has two very desirable features for large-scale service systems, which makes it possible to reduce the multi-class multi-pool staffing (and performance analysis) problem to the well-understood single-class single-pool staffing (and performance analysis) problem. First, FQR tends to produce *state-space collapse* (SSC); for the  $X$  model, the two-dimensional vector queue-length process  $(Q_1(t), Q_2(t))$  tends to evolve as a one-dimensional process, determined by the total queue length  $Q_\Sigma(t) \equiv Q_1(t) + Q_2(t)$ . In particular,

$$(Q_1(t), Q_2(t)) \approx (p_1 Q_\Sigma(t), p_2 Q_\Sigma(t)), \quad \text{where} \quad p_1 = \frac{r}{1+r} = 1 - p_2. \quad (1.2)$$

Indeed, Gurvich and Whitt (2007) showed that, under regularity conditions, FQR achieves SSC asymptotically in the quality-and-efficiency-driven (QED) many-server heavy-traffic limiting regime. (See §2 for more on the QED regime.)

Second, with FQR, it is possible to choose the ratio parameter  $r$  (or, equivalently, the queue proportions  $p_i$ ) in order to provide desired service-level differentiation. For example, we might want 80% of class-1 customers to wait less than 20 seconds, while 80% of class-2 customers wait less than 60 seconds. To see how this can be done, let  $T_i$  be the class- $i$  delay target (e.g., 20 seconds, which corresponds to 0.067 if we measure time in mean service times and they are 5 minutes); let  $W_i$  be the class- $i$  waiting time; let  $p_i$  be the queue proportion determined by  $r$ . The following string of approximations show how the individual class- $i$  performance targets  $P(W_i > T_i) \leq \alpha$ , for both  $i$ , can be reduced into a single-class single-pool performance target  $P(W > T) \leq \alpha$  for an appropriate choice of the queue proportions  $p_i$  and the aggregate target  $T$ :

$$\begin{aligned} P(W_i > T_i) &\approx P(Q_i > \lambda_i T_i) \approx P(p_i Q_\Sigma > \lambda_i T_i) \approx P\left(Q_\Sigma > \sum_{k=1}^2 \lambda_k T_k\right) \\ &\approx P\left(\lambda W > \sum_{k=1}^2 \lambda_k T_k\right) \approx P(W > T) \leq \alpha, \end{aligned} \quad (1.3)$$

where we define  $p_i \equiv \lambda_i T_i / (\lambda_1 T_1 + \lambda_2 T_2)$ ,  $\lambda \equiv \lambda_1 + \lambda_2$  and  $T \equiv (\lambda_1 T_1 + \lambda_2 T_2) / (\lambda_1 + \lambda_2)$ . The first approximation in (1.3) follows by a heavy-traffic generalization of Little's law, establishing that the steady-state queue-length and waiting-time random variables are related approximately by  $Q_i \approx \lambda_i W_i$ . The second approximation in (1.3) is due to SSC:  $Q_i \approx p_i Q_\Sigma$ . The third approximation is obtained by choosing  $p_i$  as specified above. The fourth approximation in (1.3) follows from the heavy-traffic

generalization of Little’s law once again, for the entire system:  $Q_\Sigma \approx \lambda W$  for  $\lambda$  as defined above, where  $W$  is the waiting time for an arbitrary customer. The fifth and final approximation follows by the appropriate definition of the aggregate target  $T$ , as defined above. With this reduction, we can determine the overall staffing by using elementary established methods for the single-class single-pool  $I$  model.

However, in our setting, if the service provided by non-designated agents is inefficient, then FQR is not appropriate, because it produces too much sharing. (In that case, the conditions in the key theorems of Gurvich and Whitt (2007) are violated.) Indeed, in the Appendix we show that, even if there is normal loading without sharing, FQR can cause the queue length to explode without customer abandonment if the service rates for serving the other class are less than the service rates for serving the designated class, because it can make the model unstable by inducing undesired sharing.

**The Proposed Control: FQR-T.** In order to permit sharing only in the presence of unbalanced overloads, we suggest *fixed-queue-ratio routing with thresholds* (FQR-T). We introduce positive thresholds  $\kappa_{1,2}$  and  $\kappa_{2,1}$ . Upon service completion, a newly available type-2 agent serves the customer at the head of the class-1 queue if  $D(t) \geq \kappa_{1,2}$ ; a newly available type-1 agent serves the customer at the head of the class-2 queue if  $D(t) \leq -\kappa_{2,1}$ ; otherwise the agents serve only their own class. Upon arrival, a class- $i$  customer is routed to pool  $i$  if there are idle servers; otherwise the arrival goes to the end of the class- $i$  queue. An arrival might increase the queue to a point that sharing is activated. Then the first customer in queue is served by the other class (presumably the agent that has been idle the longest).

In order to further prevent unwanted sharing, we also restrict the routing to *one-way sharing* at any time. We do not allow a newly available type-2 agent to serve a waiting class-1 customer if there are any type-1 agents busy serving class-2 customers. And similarly in the other direction.

**Analyzing the Performance.** The rest of this paper is devoted to developing tractable approximations to show how the FQR-T control performs. Having relatively simple methods to predict the performance makes FQR-T a more attractive control. These approximate performance descriptions provide a means to select appropriate control parameters.

The ability to analyze FQR-T depends on the SSC property that proved so useful to analyze FQR in Gurvich and Whitt (2007). With appropriate parameters, under normal loading there is little sharing, making the  $X$  model behave as two independent single-class single-pool  $I$  models, so that there clearly is no SSC. Thus, without overloads, the system can be analyzed by standard methods. On the other hand, in the presence of unbalanced overloads, there is substantial sharing. That typically causes both classes to be overloaded after sharing, making the large-scale  $X$  model exhibit SSC. If class 1 is more overloaded, then the queue lengths tend to be related approximately by  $Q_1(t) \approx rQ_2(t) + \kappa_{1,2}$ . If instead class 2 is more overloaded, then we have  $Q_1(t) \approx rQ_2(t) - \kappa_{2,1}$ . Moreover, that overloading tends to put large-scale systems in the efficiency-driven (ED) many-server heavy-traffic limiting regime, so that we can accurately approximate the performance by deterministic fluid models and diffusion-process refinements appropriate for the ED regime, as in Whitt (2004, 2006). As a consequence, we obtain relatively simple normal approximations for the steady-state distribution of  $(Q_1(t), Q_2(t))$ , whose accuracy is confirmed by simulation experiments.

As we explain in §2, and intend to show in subsequent papers, the approximations we develop in this paper can be based on heavy-traffic stochastic-process limits involving the concepts of state-space collapse and a heavy-traffic averaging principle, as in Coffman et al. (1995). However, we prove no limit theorems here. Here we contribute by showing how state-space collapse and the heavy-traffic averaging principle can be applied as engineering principles to better understand how large systems perform, and heuristically develop useful quantitative performance approximations. We empirically justify the approximations we develop by making extensive comparisons with simulations.

**Organization of the Rest of the Paper.** In §2 we introduce the model and discuss the efficiency-driven (ED) many-server heavy-traffic limiting regime. In §3 we develop the deterministic fluid approximation for the steady-state performance. There are two important cases of unbalanced overloads: (1) when both classes are overloaded after sharing, and (2) when one class is overloaded and the other class remains underloaded even after the sharing. In §4 we develop the deterministic fluid approximation for the transient behavior in the fully-overloaded case; that is a system of ordinary differential equations, based on the heavy-traffic averaging principle. In §5 we discuss two stochastic refinements to

the deterministic approximations for the steady-state quantities. In §6 we compare our approximations for mean values with simulations. In §7 we develop a diffusion approximation to generate approximations for the full steady-state distributions. In §8 we compare our distribution approximations with simulations. Finally, in §9 we draw conclusions.

Additional material appears in an appendix. First, in §A we show that FQR alone, without thresholds, can produce very bad performance even in normal loading when cross-trained service is inefficient. We develop approximations for this case as well and show that they are accurate. Second, in §B we discuss the advantage of the lower thresholds for very large systems to relax the one-way sharing restriction in order to make the FQR-T control more adaptive. Finally, In §C we present additional simulations results in order to give a broader picture of the performance of FQR-T.

## 2. The Model

In order to analyze the approximate performance of the  $X$  model with FQR-T routing, we consider a Markovian model. Customers from the two classes arrive according to independent Poisson processes with arrival rates  $\lambda_1$  and  $\lambda_2$ . There is a queue for each customer class. We assume that waiting customers have limited patience. A class- $i$  customer will abandon if he does not start service before a random time that is exponentially distributed with mean  $1/\theta_i$ .

There are two service pools, with pool  $j$  having  $m_j$  homogeneous servers working in parallel. The service times are mutually independent exponential random variables, but the mean may depend on both the customer class and the service pool. The mean service time for a class- $i$  customer served by a type- $j$  agent is  $1/\mu_{i,j}$ . Let the service times, abandonment times and arrival processes be mutually independent. Let  $Q_i(t)$  be the number of class- $i$  customers in queue and let  $Z_{i,j}(t)$  be the number of type- $j$  agents busy serving class- $i$  customers, at time  $t$ . With the assumptions above, the stochastic process  $(Q_i(t), Z_{i,j}(t); i = 1, 2; j = 1, 2)$  is a six-dimensional continuous-time Markov chain.

Our primary goal is to develop approximations for the steady-state random quantities  $Q_i \equiv Q_i(\infty)$  and  $Z_{i,j} \equiv Z_{i,j}(\infty)$  in the presence of overloads. In particular, we show that  $(Q_1, Q_2)$  can be approximated by a bivariate normal distribution, having correlation 0 without unbalanced overloads and having

correlation 1 under unbalanced overloads. We develop explicit formulas for the means and variances.

**The ED Many-Server Heavy-Traffic Limiting Regime.** In future work, we intend to show that the approximations developed here are asymptotically correct in the efficiency-driven (ED) many-server heavy-traffic limiting regime, but we do not establish any limits here. Nevertheless, this ED limiting regime helps to understand how we get the approximations and when they should perform well.

The many-server heavy-traffic regimes can be specified by considering a sequence of models indexed by  $n$ ; we let a superscript denote the quantity associated with model  $n$ . The main idea is that the system scale should grow. Accordingly, we assume that the arrival rates and numbers of servers grow proportionally to  $n$ :

$$\frac{\lambda_i^{(n)}}{n} \rightarrow \bar{\lambda}_i \quad \text{and} \quad \frac{m_j^{(n)}}{n} \rightarrow \bar{m}_j \quad \text{as} \quad n \rightarrow \infty, \quad (2.1)$$

where  $\bar{\lambda}_i$  and  $\bar{m}_j$  are positive constants for  $i = 1, 2$  and  $j = 1, 2$ . However, the behavior of individual customers and agents should not change, so the individual abandonment rates  $\theta_i$  and service rates  $\mu_{i,j}$  remain constant for all  $n$ .

For a Markovian model with one service pool, one customer class and customer abandonment, i.e., the  $M/M/m + M$  model, three different many-server heavy-traffic limiting regimes were identified in Garnett et al. (2002): If the system is asymptotically overloaded, then it is called the *efficiency-driven* (ED) limiting regime; if the system is asymptotically critically loaded, then it is called the *quality-and-efficiency-driven* (QED) limiting regime; if the system is asymptotically underloaded, then it is called the *quality-driven* (QD) limiting regime. Similar cases without abandonment had been specified by Halfin and Whitt (1981). For one class and one pool, it is natural to let  $n$  be the total number of servers ( $m_n = n$  for all  $n$ , so that  $\bar{m} = 1$  in (2.1)). Then the regimes are determined by the limit

$$\lim_{n \rightarrow \infty} \left(1 - \rho^{(n)}\right) \sqrt{n} \rightarrow \beta \quad \text{as} \quad n \rightarrow \infty, \quad (2.2)$$

where  $\rho^{(n)} \equiv \lambda^{(n)}/n\mu$  is the traffic intensity in model  $n$ . The regimes (i) ED, (ii) QED, and (iii) QD then occur, respectively, if the limit in (2.2) holds with (i)  $\beta = -\infty$ , (ii)  $-\infty < \beta < \infty$ , and (iii)  $\beta = +\infty$ .

We will be concentrating on overloaded systems, i.e., the ED regime, which for the  $I$  model is discussed in Whitt (2004). That provides important background for our work on the  $X$  model here. In that context we will consider the ED regime under the analog of the conventional more restrictive condition that  $\rho^{(n)} = \rho > 1$ . The ED regime is quite practical because even a small amount of customer abandonment keeps the queue-length processes stable; the queue lengths have proper steady-state distributions whenever the abandonment rates are positive.

**Stochastic-Process Limits for Scaled Processes.** We now indicate the kind of stochastic-process limits that should hold as  $n \rightarrow \infty$  in the ED many-server heavy-traffic limiting regime specified by (2.1) with  $\rho_i^{(n)} = \rho_i > 1$  for at least one class  $i$  (making the system overloaded for at least one class). Our descriptions will be useful when the system remains overloaded for at least one class after the sharing. Paralleling (2.1), we assume that the thresholds are asymptotically proportional to  $n$  as well:  $\kappa_{1,2}^{(n)}/n \rightarrow \bar{\kappa}_{1,2} > 0$  as  $n \rightarrow \infty$ .

First, for deterministic fluid limits, we consider the scaled processes

$$\bar{Q}_i^{(n)}(t) \equiv \frac{Q_i^{(n)}(t)}{n} \quad \text{and} \quad \bar{Z}_{i,j}^{(n)}(t) \equiv \frac{Z_{i,j}^{(n)}(t)}{n}, \quad t \geq 0. \quad (2.3)$$

We anticipate that these scaled processes converge as  $n \rightarrow \infty$ , with

$$(\bar{Q}_i^{(n)}(t), \bar{Z}_{i,j}^{(n)}(t), i = 1, 2; j = 1, 2) \Rightarrow (\bar{Q}_i(t), \bar{Z}_{i,j}(t), i = 1, 2; j = 1, 2) \quad \text{as } n \rightarrow \infty, \quad (2.4)$$

where  $\Rightarrow$  denotes convergence in distribution and the limit  $(\bar{Q}_i(t), \bar{Z}_{i,j}(t), i = 1, 2; j = 1, 2)$  evolves as a deterministic dynamical system, in particular, as a six-dimensional *ordinary differential equation* (ODE) or system of ODE's. We emphasize that the overloaded ED regime is essential for this limit to be meaningful. In the QD and QED regimes (with corresponding initial conditions) we expect these limits to hold with a trivial null (zero) limit  $\bar{Q}_i(t) = \bar{Z}_{i,j}(t) = 0$  for all  $i, j$ , and  $t \geq 0$ .

Since the limit process in (2.4) is deterministic, the mode of convergence  $\Rightarrow$  is equivalent to convergence in probability; the limit is often referred to as a weak law of large numbers (WLLN) or a functional WLLN (FWLLN). From this asymptotic perspective, we think of our deterministic fluid

approximation as being

$$Q_i^{(n)}(t) \approx n\bar{Q}_i(t) \quad \text{and} \quad Z_{i,j}^{(n)}(t) \approx n\bar{Z}_{i,j}(t). \quad (2.5)$$

Moreover, we anticipate that all processes have well-defined steady-state limits as  $t \rightarrow \infty$  and that the double limit in (2.4) as  $n \rightarrow \infty$  and  $t \rightarrow \infty$  (in any order) is valid and equals the limit as  $t \rightarrow \infty$  of the ODE, which is the unique stationary point for the ODE (but none of that will be proved here). From this asymptotic perspective, we think of our deterministic fluid approximation for the steady-state random variables  $Q_i$  and  $Z_{i,j}$  as being

$$Q_i \equiv Q_i^{(n)}(\infty) \approx n\bar{Q}_i(\infty) \quad \text{and} \quad Z_{i,j} \equiv Z_{i,j}^{(n)}(\infty) \approx n\bar{Z}_{i,j}(\infty) \quad (2.6)$$

for some suitably large  $n$ , but we will not include the  $n$  in our heuristic development of the approximations.

We anticipate that there also will be associated stochastic limits that serve as refinements of the fluid limits above. For these, we introduce the new scaled processes

$$\hat{Q}_i^{(n)}(t) \equiv \frac{Q_i^{(n)}(t) - n\bar{Q}_i(t)}{\sqrt{n}} \quad \text{and} \quad \hat{Z}_{i,j}^{(n)}(t) \equiv \frac{Z_{i,j}^{(n)}(t) - n\bar{Z}_{i,j}(t)}{\sqrt{n}}, \quad t \geq 0. \quad (2.7)$$

We anticipate that these scaled processes also converge as  $n \rightarrow \infty$ , with

$$(\hat{Q}_i^{(n)}(t), \hat{Z}_{i,j}^{(n)}(t), i = 1, 2; j = 1, 2) \Rightarrow (\hat{Q}_i(t), \hat{Z}_{i,j}(t), i = 1, 2; j = 1, 2) \quad \text{as} \quad n \rightarrow \infty, \quad (2.8)$$

where the limit  $(\hat{Q}_i(t), \hat{Z}_{i,j}(t), i = 1, 2; j = 1, 2)$  evolves as a stochastic process. From this new asymptotic perspective, we think of our stochastic refinement of the fluid approximation as being

$$Q_i^{(n)}(t) \approx n\bar{Q}_i(t) + \sqrt{n}\hat{Q}_i(t) \quad \text{and} \quad Z_{i,j}^{(n)}(t) \approx n\bar{Z}_{i,j}(t) + \sqrt{n}\hat{Z}_{i,j}(t). \quad (2.9)$$

Moreover, we again anticipate that all processes have well-defined steady-state limits as  $t \rightarrow \infty$  and that the double limit in (2.8) as  $n \rightarrow \infty$  and  $t \rightarrow \infty$  (in any order) is valid and equals the limit as  $t \rightarrow \infty$  of the stochastic limit in (2.8), which is the unique stationary distribution for the limiting stochastic process (but again none of that will be proved here). From this asymptotic perspective, we

think of our refined stochastic approximation for the steady-state quantities as being

$$Q_i \equiv Q_i^{(n)}(\infty) \approx n\bar{Q}_i(\infty) + \sqrt{n}\hat{Q}_i(\infty) \quad \text{and} \quad Z_{i,j} \equiv Z_{i,j}^{(n)}(\infty) \approx n\bar{Z}_{i,j}(\infty) + \sqrt{n}\hat{Z}_{i,j}(\infty) \quad (2.10)$$

for some suitably large  $n$ .

Even though we do not prove any of these stochastic-process limits here, we do verify them empirically with simulation by showing the performance for several values of  $n$ , in particular, for  $n = 25, 100$  and  $400$ . We see remarkable accuracy for  $n = 400$  and surprisingly good rough approximations even for  $n = 25$ .

**Scaling of the Thresholds.** Finally, we point out that the scaling itself provides very important insights. For example, here the scaling is very important when we consider the thresholds for sharing. Above, we have stipulated that  $\kappa_{1,2}^{(n)}/n \rightarrow \bar{\kappa}_{1,2} > 0$ , and we will scale that way in our simulation experiments in order to see the statistical regularity as a function of  $n$  indicated by the stochastic-process limits above. In particular, our examples have  $\bar{\kappa}_{1,2} = 0.1$ . If there are  $n$  servers in pool 1, then that threshold setting produces a delay burden (before sharing is activated) of only  $0.1\mu_{1,1}^{-1}$  (0.1 mean service times) independent of  $n$ . (There are  $0.1n$  customers being served at total rate  $\mu_{1,1}n$ .)

However, in applications we have one system (one value of  $n$ ) for which we must choose thresholds. There are two conflicting desires in the choice of the threshold values. On the one hand, we want the threshold relatively small, so that we activate sharing as soon as possible if sharing is needed. On the other hand, we want the threshold relatively large, so that we do not inadvertently cross over to sharing the wrong way due to random fluctuations in the stochastic processes. Indeed, if the thresholds are too small, then we can obtain the very bad behavior described in the Appendix.

Thus, we want threshold values that are neither too large nor too small. Fortunately, the scaling in the stochastic-process limits provides guidance. They indicate that we could let the thresholds become smaller, relatively, as  $n$  increases; i.e., the thresholds should increase with  $n$ , but we could let  $\kappa_{1,2}^{(n)}/n \rightarrow 0$ . Since the random fluctuations should be asymptotically of order  $O(\sqrt{n})$ , we can simultaneously achieve both objectives asymptotically if we let  $\kappa_{1,2}^{(n)}$  grow like  $n^\delta$  for  $1/2 < \delta < 1$ . Then, in the fluid scale, the threshold is asymptotically negligible, but at the same time the threshold will be large

compared to the  $O(\sqrt{n})$  stochastic fluctuations. This asymptotic analysis does not tell us what the thresholds should be in any instance, but it suggests that we should be able to find effective threshold levels for large systems. We verify the effectiveness of thresholds through simulations.

### 3. The Fluid Approximation for the Steady State of the X Model

In this section we develop the deterministic fluid approximation for the steady-state quantities  $Q_i$  and  $Z_{i,j}$  in the  $X$  model with unbalanced overloads. These yield helpful quick approximations that perform remarkably well, but we will also develop refinements in §5 and §7 that are more accurate. Here we do not directly consider the many-server heavy-traffic limiting regime specified in (2.1) and we do not introduce the scale factor  $n$ , but we are thinking of that regime with a suitably large  $n$ . The approximations are intended for  $X$  model with many servers and associated high arrival rates.

Without loss of generality, when we consider the behavior under unbalanced overload, **we assume that class 1 is overloaded, and more so than class 2 if class 2 is also overloaded.** We first specify the conditions for class 1 to be overloaded, and then identify two different cases for class 2: after sharing, class 2 is either overloaded or underloaded. (There is also a boundary case in which class 2 is critically loaded, but we do not consider it.)

**The Overloaded Conditions for Class 1.** When we say that class 1 is overloaded, we mean that  $\lambda_1 > m_1\mu_{1,1}$ , which is equivalent to  $\rho_1 \equiv \lambda_1/m_1\mu_{1,1} > 1$ , where  $\rho_1$  is the class-1 traffic intensity in isolation. In other words, we assume that class 1 with pool 1 alone would produce an  $M/M/m + M$  model in the ED regime, as in Whitt (2004). Since we have customer abandonment, the system is stable. From Whitt (2004), we obtain the deterministic fluid approximation for the steady-state queue length of class 1 alone, namely,

$$Q_1^{alone} \approx \frac{\lambda_1 - m_1\mu_{1,1}}{\theta_1}. \quad (3.1)$$

This ED steady-state fluid approximation can be derived heuristically by simply equating the rates in and out in equilibrium, assuming that there is a positive deterministic queue length  $Q_1$  for class 1 (the fluid approximation):

$$\text{rate in at queue 1} \equiv \lambda_1 = m_1\mu_{1,1} + Q_1\theta_1 \equiv \text{rate out at queue 1} \quad (3.2)$$

We will use similar simple heuristic reasoning for the  $X$  model. The associated approximate potential waiting time (for a customer with infinite patience) is  $W_1 \approx Q_1/m_1\mu_{1,1}$  (expressed in units of mean service times).

There are two cases for the less loaded class 2. We may either have class 2 also overloaded after the sharing, but less so than class 1, or class 2 underloaded after the sharing.

### 3.1. The Fully-Overloaded Case: Class 2 Overloaded After Sharing

We now describe the conditions for class 2 to be overloaded after sharing. **First, class 2 might be overloaded alone.** Paralleling the analysis above, that occurs if  $\lambda_2 > m_2\mu_{2,2}$  or, equivalently, if  $\rho_2 \equiv \lambda_2/m_2\mu_{2,2} > 1$ . Again, the system is still stable because of the abandonment. In that event, the steady-state queue length of class 2 alone would be  $Q_2^{alone} \approx (\lambda_2 - m_2\mu_{2,2})/\theta_2$ . In order to have sharing (class 2 helping class 1) in steady-state, we need  $Q_1^{alone} \geq rQ_2^{alone} + \kappa_{1,2}$ .

The idea then is that there should be approximately a fixed level of sharing, with  $Z_{1,2}$  type-2 agents serving class-1 customers. This level of sharing should make both classes 1 and 2 overloaded. First, given  $Z_{1,2}$ , by the same reasoning as before, the two individual queue lengths should be

$$Q_1 = \frac{\lambda_1 - (m_1\mu_{1,1} + Z_{1,2}\mu_{1,2})}{\theta_1} \quad \text{and} \quad Q_2 = \frac{\lambda_2 - (m_2 - Z_{1,2})\mu_{2,2}}{\theta_2}. \quad (3.3)$$

We also should have  $Q_1 = rQ_2 + \kappa_{1,2}$ . Then it is easy to see that the desired amount of sharing is the unique solution of the following linear equation in the single variable  $Z_{1,2}$ :

$$Q_1 = Q_1^{alone} - \frac{Z_{1,2}\mu_{1,2}}{\theta_1} = rQ_2 + \kappa_{1,2} = r \left( Q_2^{alone} + \frac{Z_{1,2}\mu_{2,2}}{\theta_2} \right) + \kappa_{1,2}. \quad (3.4)$$

Clearly, there is one and only one value of  $Z_{1,2}$  yielding equality, with  $D \equiv Q_1 - rQ_2 = \kappa_{1,2}$ , because at  $Z_{1,2} = 0$  the left side is greater than the right side, by assumption, and the left (right) side is decreasing (increasing) in  $Z_{1,2}$ , so that there must be equality for one and only one value of  $Z_{1,2}$  (assuming strictly positive parameters). If the solution yields  $Z_{1,2} > m_2$ , then the required sharing is not possible. In that event, even if all class 2 agents work on class-1 customers, the overloads can not be balanced in the desired way. However, if  $0 \leq Z_{1,2} \leq m_2$ , then we have found our desired answer. All three variables  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$  can equivalently be found by solving the following **two equations**

in two unknowns ( $Q_1$  and  $Z_{1,2}$ ):

$$Q_1 = \frac{\lambda_1 - (m_1\mu_{1,1} + Z_{1,2}\mu_{1,2})}{\theta_1} \quad \text{and} \quad Q_2 = \frac{Q_1 - \kappa_{1,2}}{r} = \frac{\lambda_2 - (m_2 - Z_{1,2})\mu_{2,2}}{\theta_2}. \quad (3.5)$$

**Now suppose that class 2 alone is underloaded.** We now seek conditions for there to be sharing, but where class 2 becomes overloaded when it helps class 1. It is easy to see that this case is also covered by the pair of equations in (3.5). The equations for  $Q_i$  can be interpreted as balancing the rate in with the rate out, assuming a fixed positive queue length for that class. The only requirement of the solution to (3.5) is that  $0 \leq Z_{1,2} \leq m_2$  and  $Q_2 \geq 0$ . We will then necessarily have  $Q_1 = rQ_2 + \kappa_{1,2}$ .

**Example 3.1. (canonical example)** A canonical example has  $\lambda_i = 90$ ,  $m_i = 100$ ,  $\theta_i = 0.2$ ,  $\mu_{i,j} = 1$  for all  $i$  and  $j$ ,  $r = 1$  and  $\kappa_{1,2} = \kappa_{2,1} = 10$  without overloads, but then a shift to  $\lambda_1 = 130$  under an unexpected overload for class 1. The simple fluid approximations yield  $Q_1 \approx 150$  and  $Q_2 \approx 0$  without any sharing, and then  $Q_1 \approx Q_2 \approx 50$  with FQR-T. The associated approximate potential waiting times (expressed in mean service times) for class 1 are reduced from 1.5 to 0.5 at the expense of increasing class-2 waiting times from 0 to 0.5. Simulation shows that this is indeed what happens, approximately.

### 3.2. The Spare-Capacity Case: Class 2 Underloaded After Sharing

The remaining case is the fortunate case when the class-2 load is low when the class-1 load is unexpectedly high. Then pool 2 might be able to help class 1 without penalty. Clearly, the FQR-T control is very desirable in this case.

From the fluid model perspective (ignoring stochastic fluctuations), this case occurs if and only if we can simultaneously have  $Q_1 = \kappa_{1,2} - 1$  and  $Q_2 = 0$ . Assuming that there always are available agents in pool 2, whenever  $Q_1(t) = \kappa_{1,2}$ , a type-2 agent immediately serves a class-1 customer. Hence, we must have  $Q_1(t) \leq \kappa_{1,2} - 1$ .

In the fluid model, we achieve that value  $\kappa_{1,2} - 1$  for  $Q_1$  if and only if  $Z_{1,2}$  serves to balance the rate in and rate out at queue 1:

$$\text{rate in at queue 1} \equiv \lambda_1 = m_1\mu_{1,1} + (\kappa_{1,2} - 1)\theta_1 + Z_{1,2}\mu_{1,2} \equiv \text{rate out at queue 1}, \quad (3.6)$$

yielding

$$Z_{1,2} = \frac{\lambda_1 - m_1\mu_{1,1} - (\kappa_{1,2} - 1)\theta_1}{\mu_{1,2}}, \quad (3.7)$$

while still allowing queue 2 to be empty; i.e., so that

$$\text{rate in at queue 2} \equiv \lambda_2 \leq \mu_{2,2}(m_2 - Z_{1,2}) \equiv \text{maximum rate out at queue 2}, \quad (3.8)$$

in which case we still have  $Q_2 = 0$  along with  $Q_1 = \kappa_{1,2} - 1$ . By (3.8), we necessarily have  $Z_{1,2} < m_2$ .

#### 4. The Fluid-Model System of ODE's in the Fully Overloaded Case

We will now introduce the deterministic fluid-model ODE's to approximate the evolution (transient behavior) of the CTMC  $(Q_i(t), Z_{i,j}(t); i = 1, 2; j = 1, 2)$  in the fully overloaded case considered in §3.1. From an asymptotic perspective, we think of this approximation being (2.5) stemming from the FWLLN (2.4), but we develop the approximation directly, without considering a sequence of models.

##### 4.1. A Heavy-Traffic Averaging Principle

The ODE's (and the later stochastic refinements of the steady-state approximation) depend on a heavy-traffic averaging principle, paralleling Coffman et al. (1995). Here we explain it and exploit it, but we do not prove it.

In §3.1 we exploited SSC to deduce that  $Q_1 \approx rQ_2 + \kappa_{1,2}$  in the fully-overloaded case. However, it is evident that SSC does not actually occur in such a simple way. Instead, the queue-difference process  $D(t)$  oscillates around the threshold  $\kappa_{1,2}$ . The key observation is that the process  $D(t)$  moves in a faster time scale than the other processes under consideration. In a very small amount of time, the fluid processes  $Q_i(t)$  and  $Z_{i,j}(t)$  do not change much relative to their values (roughly the same order as the number of servers), while  $D(t)$  moves rapidly between the two regions  $(-\infty, \kappa_{1,2})$  and  $[\kappa_{1,2}, \infty)$ , and hence reaches a time-dependent steady-state very rapidly. (We assume that  $\kappa_{1,2}$  and  $\kappa_{2,1}$  are sufficiently large that we can ignore the rare occasions when  $D(t) \leq -\kappa_{2,1}$ . Recall that we have one-way sharing. Hence,  $D(t)$  can only be in the two regions:  $(-\kappa_{2,1}, \kappa_{1,2})$  and  $[\kappa_{1,2}, \infty)$ .)

Since the process  $D(t)$  moves much faster than the other processes, we conclude that  $D(t)$  approximately reaches a time-dependent steady state instantaneously at each time  $t$ , where that steady-state

distribution depends on the time-dependent quantities  $Q_i(t)$  and  $Z_{i,j}(t)$ . It is perhaps better to write  $D_t(s)$  because, for given  $t$ , the process is evolving in a faster time scale, denoted here by  $s$ . Let  $D_t(\infty)$  denote a random variable with that time-dependent steady-state distribution of the time-dependent BD process, i.e., the distribution of  $D_t(s)$  as  $s \rightarrow \infty$ . We will then exploit the time-dependent probabilities

$$\pi_{1,2}(t) \equiv P(D_t(\infty) \geq 0), \quad t \geq 0. \quad (4.1)$$

This averaging principle allows us to regard  $D_t(s)$  approximately as a birth-and-death (BD) process, with state space in  $\mathbb{Z}$ , and birth and death rates that depend only on  $t$ . For each  $t$ , we can solve the BD balance equations to find the steady-state distribution of  $D_t(s)$ , i.e., the distribution of  $D_t(\infty)$ .

Let  $\hat{\lambda}_j(t)$  and  $\hat{\mu}_j(t)$  be these birth and death rates, respectively. These should be regarded as fixed rates operating in the fast time scale denoted by  $s$ . There are different formulas in the two regions. First, for  $j \in (-k_{2,1}, k_{1,2})$ , the birth rates are

$$\hat{\lambda}_j(t) = \lambda_1 + \mu_{1,2}Z_{1,2}(t) + \mu_{2,2}Z_{2,2}(t) + \theta_2Q_2(t), \quad (4.2)$$

corresponding to a class-1 arrival or a departure from the class-2 customer queue, caused by a type-2 agent service completion or by a class-2 customer abandonment. The death rates are

$$\hat{\mu}_j(t) = \lambda_2 + \mu_{1,1}m_1 + \theta_1Q_1(t), \quad (4.3)$$

corresponding to a class-2 arrival or a departure from the class-1 customer queue, caused by a class-1 agent service completion or by a class-1 customer abandonment.

Next, for  $j \in [k_{1,2}, \infty)$ , we have birth rates

$$\hat{\lambda}_j(t) = \lambda_1 + \theta_2Q_2(t), \quad (4.4)$$

corresponding to a class-1 arrival or a departure from the class-2 customer queue caused by a class-2 customer abandonment. The death rates are

$$\hat{\mu}_j(t) = \lambda_2 + \mu_{1,1}m_1 + \mu_{1,2}Z_{1,2}(t) + Z_{2,2}(t)\mu_{2,2} + \theta_1Q_1(t) \quad (4.5)$$

corresponding to a class-2 arrival or a departure from the class-1 customer queue, caused by a type-1 agent service completion or a type-2 agent service completion, or by a class-1 customer abandonment.

Thus, for each  $t$ , we solve the BD balance equations with (4.2)–(4.5) to solve for the distribution of  $D_t(\infty)$  and then the important quantity  $\pi_{1,2}(t)$  in (4.1).

## 4.2. The ODE's

Since we are considering the fully-overloaded case in §3.1, the arrival rates are sufficiently high that both approximate queue lengths are positive in steady state. With that in mind, here we consider the transient behavior of the fluid model under the assumption that all agents are busy, with some type-2 agents helping class-1. We are thus describing the transient behavior near equilibrium.

First, assuming that  $\pi_{1,2}(t)$  and  $Z_{1,2}(t)$  are given and fixed, we obtain ODE's for the two queue-length processes. Let  $\dot{x}$  denote the derivative; i.e.,

$$\dot{x}(t) \equiv \left. \frac{dx(u)}{du} \right|_{u=t}. \quad (4.6)$$

We let the derivative  $\dot{Q}_1(t)$  equal the rate of increase of  $Q_1(t)$  minus its rate of decrease. The rate of increase is simply the arrival rate to customer queue 1,  $\lambda_1$ . The rate of decrease is more complicated. First, there is the rate of abandonment from queue 1, which is  $Q_1(t)\theta_1$ . Second, there is the rate of decrease from queue 1 due to service completions by servers who will next take customers from queue 1. The rate of service completions by servers who will next take customers from customer queue 1 depends on the state of the weighted-difference stochastic process  $D(t)$ . Exploiting the averaging principle, we will not focus on the actual state, but instead focus on the average state, assuming that the weighted-difference process oscillates relatively rapidly compared to the other processes. We thus assume that a proportion  $\pi_{1,2}(t)$  of the time the threshold  $k_{1,2}$  is exceeded, in which case all agents will next select a waiting customer from customer queue 1. Thus, that portion of the decrease rate is  $\pi_{1,2}(t) (m_1\mu_{1,1} + Z_{1,2}(t)\mu_{1,2})$ . There will be a corresponding, but different, rate of decrease for the proportion of time  $1 - \pi_{1,2}(t)$  that the weighted-difference process  $D(t)$  is below the threshold  $\kappa_{1,2}$ .

That reasoning leads to the two ODE's for the queue-length processes:

$$\begin{aligned}\dot{Q}_1(t) &= \lambda_1 - Q_1(t)\theta_1 - \pi_{1,2}(t) (m_1\mu_{1,1} + Z_{1,2}(t)\mu_{1,2} + (m_2 - Z_{1,2}(t))\mu_{2,2}) \\ &\quad - (1 - \pi_{1,2}(t)) (m_1\mu_{1,1})\end{aligned}\tag{4.7}$$

and

$$\dot{Q}_2(t) = \lambda_2 - Q_2(t)\theta_2 - (1 - \pi_{1,2}(t)) ((m_2 - Z_{1,2}(t))\mu_{2,2} + Z_{1,2}(t)\mu_{1,2}) .\tag{4.8}$$

Now we propose an approximating ODE for  $Z_{1,2}(t)$ , based on assuming that the proportion  $\pi_{1,2}(t)$  can be taken as given (depending on  $t$ ) . This additional ODE is

$$\dot{Z}_{1,2}(t) = \pi_{1,2}(t)(m_2 - Z_{1,2}(t))\mu_{2,2} - (1 - \pi_{1,2}(t))Z_{1,2}(t)\mu_{1,2} .\tag{4.9}$$

Finally, assuming that the approximate time-dependent variables  $Q_1(t)$ ,  $Q_2(t)$  and  $Z_{1,2}(t)$  are given, we solve for the steady-state distribution of the BD process with birth and death rates in (4.2)–(4.5) to calculate, first the distribution of  $D_t(\infty)$  and then  $\pi_{1,2}(t) \equiv P(D_t(\infty) \geq 0)$  as in (4.1).

To calculate all four quantities  $Q_1(t)$ ,  $Q_2(t)$ ,  $Z_{1,2}(t)$  and  $\pi_{1,2}(t)$ , we suggest an iterative procedure. If we are considering the transient behavior near steady-state, then it is natural to start by using the steady-state (as  $t \rightarrow \infty$ ) values  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$  from (3.5) as initial values of  $Q_1(t)$ ,  $Q_2(t)$  and  $Z_{1,2}(t)$ . We then can calculate an initial value of  $\pi_{1,2}(t)$  from the BD process with rates in (4.2)–(4.5). We can then iterate.

We can find the steady-state values  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$  themselves by simply setting the derivatives on the left sides of the ODE's (4.7) and (4.8) equal to 0 and imposing the steady-state SSC condition that  $Q_1 = rQ_2 + \kappa_{1,2}$ . It is easy to see that this method yields the same answers given in (3.4) and (3.5). Then we can directly apply equation (4.9) to find the limiting value of  $\pi_{1,2}(t)$  as  $t \rightarrow \infty$ , denoted by  $\pi_{1,2}$ , namely,

$$\pi_{1,2} = \frac{Z_{1,2}\mu_{1,2}}{Z_{1,2}\mu_{1,2} + (m_2 - Z_{1,2})\mu_{2,2}}.\tag{4.10}$$

For the special case in which  $\mu_{1,2} = \mu_{2,2}$ , we have  $\pi_{1,2} = Z_{1,2}/m_2$ .

## 5. Stochastic Refinements to the Deterministic Steady-State Fluid Approximation

In this section we present two stochastic refinements to the deterministic fluid-model approximations for the steady-state quantities  $Q_i$  and  $Z_{1,2}$ . The first exploits the averaging principle to determine the average weighted difference for the fully-overloaded case in §3.1. The second develops a BD approximation for the steady-state queue length in the spare-capacity case of §3.2.

### 5.1. The Average Difference $E[D]$ in the Fully-Overloaded Case

With the stochastic  $X$  model in the fully-overloaded case, SSC does not happen exactly; we do not get precisely  $Q_1 = rQ_2 + \kappa_{1,2}$ . Instead, under the overloading we are considering, the queue-difference process  $D(t)$  oscillates around the threshold  $\kappa_{1,2}$ . As discussed in §4.1 above, we can apply the heavy-traffic averaging principle to find an approximating steady-state distribution of  $D(t)$  by treating it as a BD process. Let  $D$  denote a random variable with the limit of these steady-state distributions as  $t \rightarrow \infty$ .

We propose refining our fluid approximation by replacing the target difference  $\kappa_{1,2}$  by the mean  $E[D]$ . To find  $E[D]$  we solve the balance equations of the BD process above, and then take the mean

$$E[D] = \sum_{j=-\kappa_{2,1}}^{\infty} jP(D(\infty) = j), \quad (5.1)$$

where we start summing from  $-\kappa_{2,1}$  since  $D(\infty)$  should be above  $-\kappa_{2,1}$  in steady-state. (In 4.1 we observed that the BD should visit the third region  $(-\infty, -\kappa_{2,1}]$  only rarely.)

This calculation can be easily done if  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$  are known. Since they depend on the value  $E[D]$ , we need to solve for them simultaneously. To do that, we propose a simple iterative algorithm which solves the **three equations**

$$\begin{aligned} Q_1 &= \frac{\lambda_1 - (m_1\mu_{1,1} + Z_{1,2}\mu_{1,2})}{\theta_1}, & Q_2 &= \frac{Q_1 - E[D]}{r} = \frac{\lambda_2 - (m_2 - Z_{1,2})\mu_{2,2}}{\theta_2}, \\ E[D] &= \sum_{j=-\kappa_{2,1}}^{\infty} jP(D = j). \end{aligned} \quad (5.2)$$

For the iterative procedure, it is natural to start with the values of  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$  obtained from (3.5), and then calculate the distribution of  $D$  and  $E[D]$ . We can then obtain new values of  $Q_1$ ,  $Q_2$  and  $Z_{1,2}$

by solving (3.5) again with  $E[D]$  replacing  $\kappa_{1,2}$ . We then can keep iterating. Experience indicates that this iteration consistently converges in a few iterations (typically only two).

## 5.2. A BD-Process Refinement for the Spare-Capacity Case

For the case in which queue 2 has spare capacity, considered in §3.2, we also develop another refinement, obtaining a non-degenerate approximation for the distribution of  $Q_1$ . In this case, because of the available agents in pool 2, as soon as  $Q_1$  hits the threshold  $\kappa_{1,2}$ , an idle pool-2 agent serves a customer from class 1. Thus, it is evident that we must have  $Q_1 \leq \kappa_{1,2} - 1$ .

Because of the averaging principle, it is not hard to estimate the approximate distribution of  $Q_1$ . To do so, we observe that we can regard the class-1 queue as evolving as a BD process. When the queue length is  $j$ , the birth rate is a constant  $\lambda_1$ , while the death rate is approximately  $m_1\mu_{1,1} + \theta_1 j$ . For the reason given, the birth rate is 0 when the queue is at  $\kappa_{1,2} - 1$ . The death rate should be small when the queue length is small. For the approximation to be good, we do not want  $Q_1$  to spend much time at very low levels, like 1 or 0. That can be verified approximately by looking at the approximate BD steady-state distribution. In any case, we let the death rate be 0 when the queue length is 0. Our refined approximation for the distribution of  $Q_1$  is the steady-state distribution of this finite-state BD process.

Since  $Q_1^{alone} = (\lambda_1 - m_1\mu_{1,1})/\theta_1 > \kappa_{1,2}$ , the birth rate always exceeds the death rate here. Indeed, the BD process here for  $\kappa_{1,2} - 1 - Q_1(t)$  is stochastically bounded above by the queue-length process in an  $M/M/1/\kappa_{1,2} - 1$  queue, where  $\kappa_{1,2} - 1$  serves as the size of a finite waiting room. If we take the asymptotic perspective in §2, this stochastic bound shows that the difference  $\kappa_{1,2} - 1 - Q_1$  should be of order  $O(1)$  as  $n \rightarrow \infty$ . Hence this adjustment should be asymptotically negligible in both the diffusion scale ( $\sqrt{n}$ ) and the fluid scale ( $n$ ). However, the refinement can help in actual examples, even large ones with 1000 servers in each pool.

As a refined deterministic fluid approximation, we use the mean value of the steady-state distribution of the BD process here. However, by this method, we also obtain an estimate for the variance and the entire distribution of  $Q_1$ . The observed  $M/M/1$  structure indicates that the distribution of  $\kappa_{1,2} - 1 - Q_1(t)$  should be approximately a truncated geometric distribution. That is quite different from the

approximate normal distribution we derive for the fully-overloaded case in the following subsection.

## 6. Simulation Experiments to Evaluate the Approximate Mean Values

### 6.1. The Overloaded Case

We have developed deterministic fluid approximations for the mean values in the fully overloaded case via the solutions to the two equations in (3.5) and the three equations in (5.2). We now compare these approximations to simulation estimates. In order to use the simulation to substantiate the conjectured stochastic-process limits in §2, we choose parameters corresponding to scaled systems, indexed by  $n$ , letting  $n$  take the values 25, 100 and 400. We have considered much larger  $n$ , such as  $n = 1000$ , but from the results for  $n = 400$ , we see that accurate results will be obtained for all  $n$  larger than 400.

Our simulation examples throughout the paper will have parameters related to a **base case** that we consider here. It has several parameters depending on  $n$ :  $m_i \equiv m_i^{(n)} = n$ ,  $\lambda_1 \equiv \lambda_1^{(n)} = 1.3n$ ,  $\lambda_2 \equiv \lambda_2^{(n)} = 0.9n$  and  $\kappa_{1,2} \equiv \kappa_{1,2}^{(n)} = \kappa_{2,1} \equiv \kappa_{2,1}^{(n)} = 0.1n$ . It also has several parameters independent of  $n$ :  $\theta_1 = \theta_2 = 0.2$ ,  $\mu_{1,1} = \mu_{2,2} = 1.0$  and  $\mu_{1,2} = \mu_{2,1} = 0.8$ . The arrival rates are chosen to put class 1 in a focused overload, while class 2 is initially normally loaded or slightly underloaded. The rest of the parameters are chosen to make a symmetric model, where serving the other class is less efficient. In the appendix we present corresponding results for asymmetric models.

All simulation experiments are based on five independent replications of runs, each having 300,000 arrivals. The independent replications make it possible to reliably estimate confidence intervals using the  $t$ -statistic with 4 degrees of freedom. We give the average of the five simulation runs and the half-width of the 95% confidence interval. The results for the base case above are presented in Table 1 below. Table 1 shows both the steady-state mean values and the associated scaled values (i.e., divided by  $n$ ). The unscaled values helps us evaluate the performance of the actual system, while the scaled values show the convergence of the stochastic-process limits in (2.4). Table 1 clearly shows that the level of accuracy grows as  $n$  gets larger, but even for relatively small systems, the fluid approximation gives reasonable results, and important insight about the system behavior.

Table 1 also gives the approximation for the steady-state mean of the unscaled weighted-difference

perf. meas.	n=25			n=100			n=400		
	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.
$E[Q_1]$	16.6	14.4	15.7 $\pm 0.3$	65.6	63.1	63.6 $\pm 1.9$	262.2	259.7	258.3 $\pm 5.0$
$E[Q_1/n]$	0.656	0.575	0.629 $\pm 0.013$	0.656	0.631	0.636 $\pm 0.019$	0.656	0.649	0.646 $\pm 0.013$
$E[Q_2]$	13.6	16.4	15.9 $\pm 0.4$	55.6	58.6	58.6 $\pm 1.8$	222.2	225.3	223.9 $\pm 5.0$
$E[Q_2/n]$	0.556	0.656	0.636 $\pm 0.016$	0.556	0.586	0.586 $\pm 0.018$	0.556	0.563	0.560 $\pm 0.013$
$E[D]$	–	–2.0	–0.2 $\pm 0.3$	–	4.6	5.0 $\pm 0.1$	–	34.4	34.4 $\pm 0.04$
$\kappa_{1,2} - E[D]$	–	5.0	3.2 $\pm 0.3$	–	5.4	5.0 $\pm 0.1$	–	5.6	5.6 $\pm 0.04$
$E[Z_{1,2}]$	5.3	5.8	5.6 $\pm 0.1$	21.1	21.7	21.9 $\pm 0.04$	84.4	85.1	84.2 $\pm 1.2$
$E[Z_{1,2}/n]$	0.211	0.231	0.224 $\pm 0.003$	0.211	0.217	0.219 $\pm 0.004$	0.211	0.213	0.210 $\pm 0.003$

Table 1: A comparison of the basic fluid approximations based on two equations in (3.5) and its refinement based on the three equations in (5.2) with simulation results in the base case, having  $m_1 = m_2 = 1.0n$ ,  $\lambda_1 = 1.3n$ ,  $\lambda_2 = 0.9n$ ,  $\mu_{1,1} = \mu_{2,2} = 1.0$ ,  $\mu_{1,2} = \mu_{2,1} = 0.8$ ,  $\theta_1 = \theta_2 = 0.2$  and  $\kappa_{1,2} = \kappa_{2,1} = 0.1n$  (rounding up to the nearest integer if necessary).

process  $D(t)$ , as developed in §5.1, and compares it to simulation results. The sixth row in the table is especially insightful. It shows that  $E[D]$  is about the same distance from  $\kappa_{1,2}$  for each  $n$ , thus strengthening our claim that  $D(t)$  should have fluctuations of order  $O(1)$  as  $n \rightarrow \infty$ .

In closing, we remark that rounding up to the nearest integer occurs for the thresholds  $\kappa_{1,2}$  when  $n = 25$ . The simulations in the case  $n = 25$  were conducted with  $\kappa_{i,j} = 3$ . In the table we chose to show the solution using  $\kappa_{1,2} = 2.5$  so as to make the scaled fluid solutions uniform. However, the solution using  $\kappa_{1,2} = 3$  is similar.

## 6.2. Independent Cases

One of our objectives is to avoid sharing without unbalanced overloads. That occurs in two scenarios:

(i) under normal loads, and (ii) under balanced overloads. In both of these cases our control makes the  $X$  model operate approximately as two independent  $M/M/n + M$  systems, each operating in the  $QD$

or *QED* regime in the first scenario (depending on the actual load of each queue), or the *ED* regime in the second scenario. As we show in the Appendix, if we use FQR without thresholds or one-way sharing, then the underloaded system may become overloaded due to the slower service rates for the other class, leading to serious performance degradation.

Table 2 shows results for a normally loaded case. We modify the base case used above only by changing the arrival rates. Now the arrival rates are  $\lambda_1 = \lambda_2 = 0.98n$ . With this change, we have a fully symmetric model, so we only show results for class 1. Since the arrival rates are close to the maximum possible service rates  $m_i\mu_{i,i} = 1.0n$ , the system should actually be regarded as critically loaded, but since there is significant abandonment, the system is not too heavily loaded. In Table 2 we only show the trivial null fluid approximations for the performance measures. In this case, we could obtain more accurate performance approximations by exploiting the *I*-model approximations developed by Garnett et al. (2002). Since  $E[Z_{1,2}]$  is quite small in each case, we conclude that our control is effective in preventing sharing here.

	n=25		n=100		n=400	
perf. meas.	approx.	sim.	approx.	sim.	approx.	sim.
$E[Q_1]$	0	4.8 $\pm 0.3$	0	7.3 $\pm 1.0$	0	10.5 $\pm 2.6$
$E[Q_1/n]$	0	0.19 $\pm 0.01$	0	0.07 $\pm 0.01$	0	0.03 $\pm 0.01$
$E[D]$	0	0.00 $\pm 0.27$	0	-0.15 $\pm 0.24$	0	0.10 $\pm 0.49$
$E[Z_{1,2}]$	0	1.3 $\pm 0.1$	0	1.9 $\pm 0.2$	0	1.3 $\pm 0.4$
$E[Z_{1,2}/n]$	0	0.052 $\pm 0.005$	0	0.019 $\pm 0.002$	0	0.003 $\pm 0.001$

Table 2: A comparison of the trivial *I*-model fluid approximation with simulation results for the steady-state performance measures in the case of balanced normal loading. The arrival rates are now  $\lambda_1 = \lambda_2 = 0.98n$ .

Table 3 shows results for a balanced overloaded case. Again, we modify the base case used above only by changing the arrival rates. Now the arrival rates are  $\lambda_1 = \lambda_2 = 1.2n$ . We again have a fully

symmetric model, so we only show results for class 1. The fluid approximation for class 1 is

$$Q_i^{alone} = \frac{\lambda_i - m_i \mu_{i,i}}{\theta_i} = \frac{1.2n - n}{0.2} = n.$$

Since  $E[Z_{1,2}]$  is quite small in each case, we conclude that our control is again effective in preventing sharing.

	n=25		n=100		n=400	
perf. meas.	approx.	sim.	approx.	sim.	approx.	sim.
$E[Q_1]$	25	26.7 $\pm 0.5$	100	103.9 $\pm 1.9$	400	407.7 $\pm 7.1$
$E[Q_1/n]$	1	1.07 $\pm 0.02$	1	1.04 $\pm 0.02$	1	1.02 $\pm 0.02$
$E[D]$	0	0.0 $\pm 0.4$	0	0.8 $\pm 0.7$	0	0.4 $\pm 3.2$
$E[Z_{1,2}]$	0	1.7 $\pm 0.0$	0	3.0 $\pm 0.2$	0	4.2 $\pm 1.7$
$E[Z_{1,2}/n]$	0	0.07 $\pm 0.00$	0	0.03 $\pm 0.00$	0	0.01 $\pm 0.00$

Table 3: A comparison of the  $I$ -model fluid approximation with simulation results for the steady-state performance measures in the case of balanced overloads. The arrival rates are now  $\lambda_1 = \lambda_2 = 1.2n$ .

### 6.3. The Spare-Capacity Case

For the spare capacity case, we modify the base case above to make queue-1 overloaded, while pool-2 has enough spare capacity to potentially serve all the extra class-1 customers. As before, we just change the arrival rates, in this case to  $\lambda_1 = 1.1n$  and  $\lambda_2 = 0.8n$ .

It is easy to see that pool 2 has spare capacity (in the fluid scale). We can analyze the available capacity from this deterministic-fluid-approximation perspective as follows: First, we observe that class 1 has an extra arrival rate of  $0.1n$ , whereas pool 2 has  $0.2n$  “extra” service rate, assuming that  $0.8n$  servers are enough to take care of all the class-2 arrivals. Since pool-2 agents serve class-1 customers at rate  $\mu_{1,2} = 0.8$ , we initially estimate that we need to have at least  $0.125n$  pool-2 agents working with class-1 customers. However, upon further analysis, we see that the number of pool-1 agents needed is actually less than that, because queue 1 will stabilize at  $\kappa_{1,2} = 0.1n$ , and thus  $\theta_1 Q_1 = 0.02n$  class-1

customers will abandon. Hence, only about  $0.105n$  pool-2 agents are needed to serve class 1. In any case, pool 2 has spare capacity.

We compare the approximation from §3.2 with simulation results in Table 4. The approximations are given in §3.2. Our initial approximation for  $Q_1$  from §3.2 is  $\kappa_{1,2} - 1$ , but that is not shown in Table 4. Instead, we only show the BD refinement from §5.2. (The cruder approximation would yield values of 1.5, 9.0 and 39.0 in the first row.) We see that the refined approximation is much better for large  $n$ . For the approximation of  $Z_{1,2}$ , we use (3.7).

	n=25		n=100		n=400	
perf. meas.	approx.	sim.	approx.	sim.	approx.	sim.
$E[Q_1]$	1.1	3.3 $\pm 0.1$	5.2	6.4 $\pm 0.6$	29.0	30.1 $\pm 0.5$
$E[Q_1/n]$	0.04	0.13 $\pm 0.00$	0.05	0.06 $\pm 0.01$	0.07	0.07 $\pm 0.00$
$E[Q_2]$	0	3.4 $\pm 0.05$	0	2.7 $\pm 0.5$	0	1.0 $\pm 0.2$
$E[Q_2/n]$	0	0.14 $\pm 0.00$	0	0.027 $\pm 0.005$	0	0.003 $\pm 0.000$
$E[Z_{1,2}]$	2.6	3.9 $\pm 0.1$	10.3	12.2 $\pm 0.5$	40.3	43.4 $\pm 1.2$
$E[Z_{1,2}/n]$	0.104	0.156 $\pm 0.007$	0.103	0.122 $\pm 0.007$	0.101	0.108 $\pm 0.003$

Table 4: A comparison of the approximation for the steady-state performance measures in the spare-capacity case with simulation results. The arrival rates are now  $\lambda_1 = 1.1n$  and  $\lambda_2 = 0.8n$ .

## 7. A Diffusion-Process Refinement in the Fully-Overloaded Case

In the fully-overloaded case, we now go beyond the deterministic fluid approximation in §3.1 and §5.1 to obtain a diffusion-process refinement, which yields a non-degenerate approximation for the steady-state distribution. The approximating distribution is normal, where the means are the previous fluid approximations themselves. In addition to this important characterization, we provide formulas for the approximating variances or, equivalently, the standard deviations.

**Leveraging Known Results.** We base our approximation on a special case for which we can do the asymptotic analysis exactly, and extend the approximation heuristically to other cases. The special case we can analyze exactly has  $\theta_1 = \theta_2$  and  $\mu_{1,2} = \mu_{2,2}$  (with class 1 overloaded as usual). Under those additional assumptions, the total queue length  $Q_\Sigma(t) \equiv Q_1(t) + Q_2(t)$  behaves the same as the queue length in the  $M/M/m + M$  model in the ED regime, as analyzed in Whitt (2004). Since the system is fully overloaded, we can assume that all the agents are busy all the time. Thus, the departure rate is the constant value  $m_1\mu_{1,1} + m_2\mu_{2,2}$ . The assumption that  $\mu_{1,2} = \mu_{2,2}$  implies that it does not matter which class the type-2 agents are serving. Since the total arrival process is a superposition of two independent Poisson processes, the total arrival process is directly a Poisson process with rate  $\lambda_1 + \lambda_2$ . Finally, since  $\theta_1 = \theta_2$ , there is a common abandonment rate for both classes.

So in this special case we can directly obtain a FCLT like (2.8) for the total queue-length stochastic process, using scaling as in (2.7). From Whitt (2004), we see that the appropriately scaled version of the difference between the total queue length and its fluid approximation can be approximated by an Ornstein-Uhlenbeck (OU) diffusion process with infinitesimal mean  $m(x) = -\theta_1 x$  and infinitesimal variance  $\sigma^2 \equiv \sigma^2(x) = 2(\lambda_1 + \lambda_2)$ . It is well known that this OU process has a normal steady-state distribution with mean zero and variance

$$\text{Var}(Q_\Sigma) \approx \frac{(\lambda_1 + \lambda_2)}{\theta_1}. \quad (7.1)$$

At this point, we invoke the SSC to get associated limits and approximations for the individual queue lengths. (We are applying SSC without proof here.) We start from the approximation for  $(Q_1, Q_2)$  in (5.2). Thus we invoke the SSC to get

$$Q_1 \approx \frac{r(Q_\Sigma + E[D])}{1+r} \quad \text{and} \quad Q_2 \approx \frac{Q_\Sigma - E[D]}{1+r} \quad (7.2)$$

for the steady-state variables. That gives a joint normal distribution for  $(Q_1, Q_2)$  with correlation 1 and individual variances

$$\text{Var}(Q_1) \approx \frac{r^2(\lambda_1 + \lambda_2)}{(1+r)^2\theta_1} \quad \text{and} \quad \text{Var}(Q_2) \approx \frac{(\lambda_1 + \lambda_2)}{(1+r)^2\theta_1}. \quad (7.3)$$

**A Heuristic Extension.** Now we heuristically extend this same tractable OU approximation with a normal steady-state distribution to more general cases. First, when  $\mu_{1,2} \neq \mu_{2,2}$ , we again act as if all agents are busy all the time. The total service rate at time  $t$  is then  $m_1\mu_{1,1} + Z_{1,2}(t)\mu_{1,2} + (m_2 - Z_{1,2}(t))\mu_{2,2}$ . To obtain the desired constant rate, we act as if  $Z_{1,2}(t)$  is constant, assuming its determined deterministic steady-state fluid approximation. This is a heuristic approximation, because we are ignoring the stochastic fluctuations in  $Z_{1,2}$ . Experiments show that this simple approximation works pretty well, but in the Appendix we show that, as  $n \rightarrow \infty$  in the ED regime, the infinitesimal mean of the scaled queue-length process does in fact depend on the stochastic behavior of the scaled version of the stochastic process  $Z_{1,2}$  (as we would expect); i.e., this heuristic extension is *not* asymptotically correct as  $n \rightarrow \infty$ .

We also treat the abandonments in a similar way when  $\theta_1 \neq \theta_2$ . We will approximate by a constant abandonment rate applying to all customers. For this step we also will invoke SSC (ignoring the difference), and assume that  $Q_1(t) \approx rQ_\Sigma(t)/(1+r)$  (and similarly for  $Q_2$ ), just as in (1.2). Thus our approximating constant abandonment rate to apply to the total queue length is

$$\theta \approx \frac{r\theta_1}{1+r} + \frac{\theta_2}{1+r}. \quad (7.4)$$

With the new approximating total service rate and average abandonment rate, we again are in the domain of an OU approximation, with normal steady-state distribution. Paralleling (7.1), we obtain a new approximate variance for the total queue length,

$$Var(Q_\Sigma) \approx \frac{(1+r)(\lambda_1 + \lambda_2)}{(r\theta_1 + \theta_2)}. \quad (7.5)$$

Then SSC again gives a joint normal distribution for  $(Q_1, Q_2)$  with correlation 1. The individual variances are approximated by

$$Var(Q_1) \approx \frac{r^2(\lambda_1 + \lambda_2)}{(1+r)(r\theta_1 + \theta_2)} \quad \text{and} \quad Var(Q_2) \approx \frac{(\lambda_1 + \lambda_2)}{(1+r)(r\theta_1 + \theta_2)}. \quad (7.6)$$

## 8. Simulation Experiments to Evaluate the Approximate Distributions

### 8.1. The Unbalanced-Overload Case

We now compare the approximating steady-state distributions to simulation results. We again consider the base case in Table 1 with  $\lambda_1 = 1.3n$  and  $\lambda_2 = 0.9n$ . The results are given in Table 5.

We give the standard-deviations of the total queue length  $Q_\Sigma = Q_1 + Q_2$  as well as the two queues. As before, we treat both the actual values and the scaled values, but now we are scaling in diffusion scale (dividing by  $\sqrt{n}$  after subtracting the order- $O(n)$  mean), as in (2.7), so that we will be substantiating the stochastic-process limit in (2.8). To further substantiate both the stochastic-process limit and the normal approximations, we also give the quantiles of the scaled queue lengths  $\hat{Q}_1$  and  $\hat{Q}_2$ . To save space, we omit the confidence intervals for the scaled standard deviations; these can be computed from the confidence intervals of the actual queues by dividing the half widths by  $\sqrt{n}$ .

We also give the quantiles for the centered steady-state queue difference  $\tilde{D} \equiv D - E[D]$ . (Table 1 already showed that the approximation for the mean  $E[D]$  is accurate for  $n \geq 100$ .) The approximate distribution of  $D$  is obtained from the BD process in §4.1. The quantiles of the distribution of  $\tilde{D}$  pose a problem, since  $D$  is integer-valued. We thus calculate a linear interpolation of two values. For example, for the 0.05 quantile, we took the largest value  $d_0$  such that  $P(\tilde{D} \leq d_0) < 0.05$  and linearly interpolate this value with the smallest value  $d_1$  such that  $P(\tilde{D} \leq d_1) > 0.05$ . The linear interpolation becomes just the weighted average of the two values  $d_0$  and  $d_1$ . As in Table 1,  $\tilde{D}$  is not scaled by any division.

To further illustrate how the approximations perform, we show two figures based on a simulation run with  $n = 100$  (the second case in Table 5). To show that SSC actually occurs with FQR-T, we show a plot of a segment of the queue-length sample paths in Figure 2. We have centered about the means, so that the average difference should be zero. To justify the normal approximation, we show a histogram of the class-1 queue-length distribution in Figure 3.

### 8.2. The Balanced-Overload Case

In the balanced-overload case, we compare the simulation results to approximations based on the assumption that the two queues operate independently, as we did in §6.2. The approximations are cal-

		n=25		n=100		n=400	
perf. meas.		Approx.	Sim.	Approx.	Sim.	Approx.	Sim.
$std(Q_\Sigma)$		16.6	16.0 $\pm 0.3$	33.2	33.7 $\pm 1.4$	66.3	67.6 $\pm 2.9$
$std(\hat{Q}_\Sigma)$		3.32	3.21	3.32	3.37	3.32	3.38
$std(Q_1)$		8.3	8.8 $\pm 0.1$	16.6	17.2 $\pm 0.7$	33.2	33.9 $\pm 1.4$
$std(\hat{Q}_1)$		1.66	1.75	1.66	1.72	1.66	1.7
$std(Q_2)$		8.3	8.6 $\pm 0.1$	16.6	17.1 $\pm 0.7$	33.2	33.9 $\pm 1.5$
$std(\hat{Q}_2)$		1.66	1.73	1.66	1.71	1.66	1.69
$\hat{Q}_1$ quantiles	0.05	-2.72	-2.75 $\pm 0.06$	-2.72	-2.84 $\pm 0.11$	-2.72	-2.72 $\pm 0.19$
	0.25	-1.12	-1.27 $\pm 0.08$	-1.12	-1.14 $\pm 0.03$	-1.12	-1.18 $\pm 0.08$
	0.75	1.12	1.13 $\pm 0.08$	1.12	1.14 $\pm 0.08$	1.12	1.11 $\pm 0.08$
	0.95	2.72	2.97 $\pm 0.11$	2.72	2.82 $\pm 0.20$	2.72	2.92 $\pm 0.16$
$\hat{Q}_2$ quantiles	0.05	-2.72	-2.94 $\pm 0.14$	-2.72	-2.82 $\pm 0.15$	-2.72	-2.68 $\pm 0.21$
	0.25	-1.12	-1.18 $\pm 0.08$	-1.12	-1.14 $\pm 0.04$	-1.12	-1.17 $\pm 0.06$
	0.75	1.12	1.18 $\pm 0.07$	1.12	1.14 $\pm 0.09$	1.12	1.11 $\pm 0.08$
	0.95	2.72	2.90 $\pm 0.10$	2.72	2.80 $\pm 0.20$	2.72	2.91 $\pm 0.15$
centered $D$ quantiles	0.05	-17.4	-13.4 $\pm 0.7$	-18.4	-16.6 $\pm 0.6$	-19.5	-18.2 $\pm 0.6$
	0.25	-7.4	-6.0 $\pm 0.0$	-8.4	-7.6 $\pm 0.6$	-8.5	-8.0 $\pm 0.0$
	0.75	-1.4	-0.8 $\pm 0.6$	-1.4	-1.0 $\pm 0.1$	-1.4	-1.0 $\pm 0.0$
	0.95	0.5	5.0 $\pm 1.8$	0.5	1.0 $\pm 0.1$	0.5	1.0 $\pm 0.0$

Table 5: A comparison of the approximating distributions of steady-state performance measures in the unbalanced-overload case with simulation results for the base case with  $\lambda_1 = 1.3n$  and  $\lambda_2 = 0.9n$ .

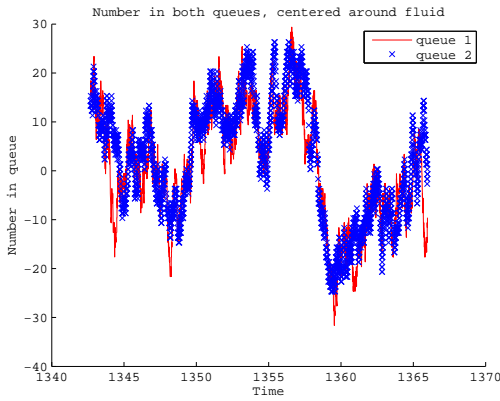


Figure 2: State-space collapse for  $n = 100$ .

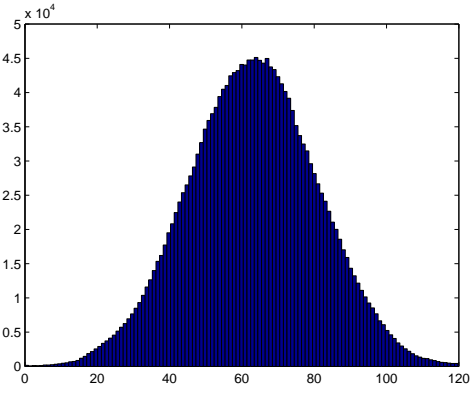


Figure 3: Histogram of  $Q_1$  for  $n = 100$ .

culated using Whitt(2004). We simulated the same systems as before, again changing only the arrival rates to  $\lambda_1 = \lambda_2 = 1.2n$ . The results are shown in Table 6.

Table 3 already showed that the two queues are not completely independent, because some agents are serving customers from the other class. Thus there is some dependency between the queues. This causes the queues to be somewhat larger than if the two service pools were operating independently, because serving the other class is done somewhat inefficiently. However, the sharing is not altogether bad: although there is some increase in the queues sizes (as shown in table 3), we also gain by decreasing the variance of the queues. From the efficiency point of view, we see a tradeoff between the economies of scale provided by the sharing and the inefficiency caused by the slower service rates when sharing.

For this reason, the simulations do not match approximations precisely, but the differences are not large. Indeed, to show the differences more clearly, for the case  $n = 100$  we added another column of simulation results for an  $M/M/100 + M$  system having a Poisson arrival process with rate  $\lambda = 120$ . (The simulations for the  $M/M/100 + M$  model were performed with the  $X$  model simulator, taking  $\kappa_{i,j} = 400$ , thus assuring the difference between the two queues will not go above the thresholds.) Table 6 shows that the simulation results for this case are much closer to the approximations. To further illustrate the difference (and the resemblance), we show histograms of the distribution of  $Q_1$  in the two cases in Figures 4 and 5. It is easy to see that the queues in both systems have a distribution

that is very close to a normal distribution, and that the variance of the queue in the  $X$  model is smaller than the variance of the queue in the  $M/M/100 + M$  system.

		n=25		n=100			n=400	
perf. meas.		Approx.	Sim.	Approx.	Sim. $X$ model	Sim. Ind.	Approx.	Sim.
$std(Q_\Sigma)$		17.3	17.3 $\pm 0.5$	34.6	33.2 $\pm 2.6$	34.7 $\pm 0.7$	69.3	63.9 $\pm 3.9$
$std(\hat{Q}_\Sigma)$		3.46	3.46	3.46	3.32	3.47	3.46	3.20
$std(Q_1)$		12.2	10.37 $\pm 0.2$	24.5	20.4 $\pm 1.3$	24.7 $\pm 0.64$	49.0	38.1 $\pm 1.7$
$std(\hat{Q}_1)$		2.45	2.07	2.45	2.04	2.47	2.45	1.91
$\hat{Q}_1$ quantiles	0.05	-4.03	-3.38 $\pm 0.07$	-4.03	-3.35 $\pm 0.26$	-4.07 $\pm 0.13$	-4.03	-3.17 $\pm 0.26$
	0.25	-1.65	-1.38 $\pm 0.07$	-1.65	-1.37 $\pm 0.11$	-1.73 $\pm 0.09$	-1.65	-1.24 $\pm 0.09$
	0.75	1.65	1.30 $\pm 0.07$	1.65	1.35 $\pm 0.07$	1.69 $\pm 0.10$	1.65	1.30 $\pm 0.12$
	0.95	4.03	3.50 $\pm 0.07$	4.03	3.45 $\pm 0.20$	4.14 0.09	4.03	3.07 $\pm 0.05$

Table 6: A comparison of approximations for the standard deviations and quantiles of the steady-state queue lengths with simulation estimates in the balanced-overload case with  $\lambda_1 = \lambda_2 = 1.2n$ .

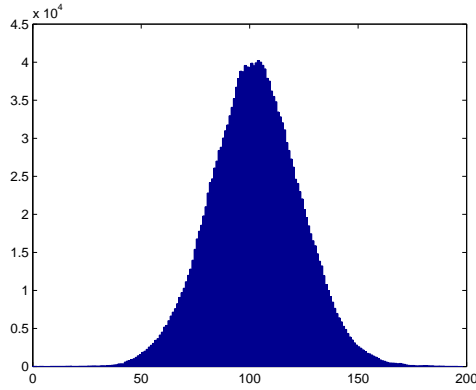


Figure 4: A histogram for  $Q_1$  in the balanced-overload case with  $n = 100$ ,  $\lambda_i = 1.2n$  and  $\kappa_{1,2} = 10$ .

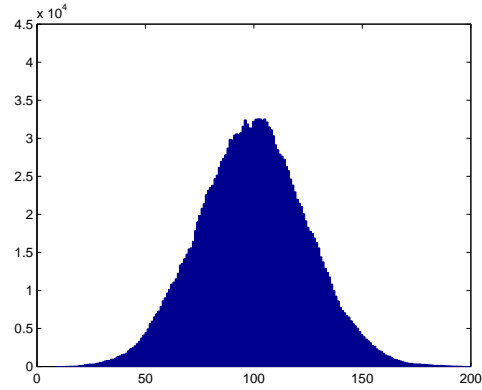


Figure 5: A histogram for the queue length in an  $M/M/100 + M$  model.

## 9. Conclusions

In this paper we proposed the FQR-T routing policy for the  $X$  model to activate sharing in response to unbalanced overloads. The FQR-T control is appealing for several reasons: (1) it is simple and easy to understand, (2) it is robust, not assuming specified arrival rates, (3) it requires only minimal state information and processing, so that it is inexpensive to implement, and (4) its performance can be analyzed.

We also demonstrated that the performance of the FQR-T control can be analyzed. For that purpose, we developed tractable approximations, exploiting the fact that the overloading puts the system in the ED many-server heavy-traffic limiting regime. Even though the approximations have a complicated basis, supported by stochastic-process limits not established here, the approximations are relatively simple and easy to apply.

From the theoretical point of view, the main contribution of this paper is the reduction of a complicated mathematical model to more elementary and elegant approximate models, using the heavy-traffic *averaging principle* in the development of the deterministic fluid approximation (the system of ODE's in §4.1) and *state space collapse* (SSC) in the diffusion approximations (and resulting approximate normal distribution in §7). The relatively simple initial fluid approximation in §3 was refined in useful ways in §5 and §7. The approximations reveal how the FQR-T performs and provides a means for selecting the parameters in order to achieve performance objectives. We also conducted simulations to show that both the FQR-T routing policy and the performance approximations perform as intended.

The whole discussion was limited to the two-class-two-pool  $X$ -model setting, but we believe that the control and the results can be generalized to larger networks. Work is in progress to establish additional results: First, we intend to show that the deterministic fluid approximation is asymptotically correct in the many-server heavy-traffic ED regime, which involves justifying the heavy-traffic averaging principle in this context. Work is also in progress to extend the performance approximations to the  $X$  model with non-exponential distributions, paralleling the previous results in Whitt (2006) for the single-class single-pool  $I$  model.

**Acknowledgments.** This research was supported by NSF Grant DMI-0457095.

## References

- Aksin, Z., M. Armony, V. Mehrotra, 2007. The modern call center: a multi-disciplinary perspective on operations management research. working paper.  
Available at: <http://www.stern.nyu.edu/om/faculty/armony/>
- Coffman, E. G., M. I. Reiman, A. A. Puhalskii. 1995. Polling systems with zero switchover times: a heavy-traffic averaging principle. *Annals of Applied Probability* 5, 681–719.
- Gans, N., G. Koole, A. Mandelbaum. 2003. Telephone call centers: tutorial, review and research prospects. *Manufacturing and Service Opns. Mgmt.*, 5, 79–141.
- Garnett, O., A. Mandelbaum, M. I. Reiman. 2002. Designing a call center with impatient customers. *Manufacturing and Service Opns. Mgmt.*, 4, 208-227.
- Green, L., P. J. Kolesar and W. Whitt. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management* 16 (2007) 13–39.
- Gurvich, I., W. Whitt. 2007. Service-level differentiation in many-server service systems: a solution based on fixed-queue-ratio routing. working paper.  
Available at: <http://www.columbia.edu/~ig2126/>
- Halfin, S. and W. Whitt. 1981. Heavy-traffic limits for queues with many exponential servers. *Operations Research* 29, 567–588.
- Whitt, W. 2004. Efficiency-driven heavy-traffic approximations for many-server queues with abandonments. *Management Science* 50 (10), 1449–1461.
- Whitt, W. 2006. Fluid models for multiserver queues with abandonments. *Operations Research*, 54 (1) 37–54.

## Appendix

This appendix contains additional material complementing the main paper. In §A we show that FQR without thresholds or one-way sharing can produce poor performance for systems that are normally loaded without sharing. In §B we show the advantage of having additional lower thresholds in order to activate sharing in the other direction more quickly, so as to be more adaptive. In §C we present additional simulation results, considering asymmetric models and more challenging boundary cases in order to understand the limitations of the approximations.

### A. Bad Behavior Without Thresholds

We have two objectives in this section. First, we want to demonstrate the need for the thresholds and the one-way sharing, which are the key ingredients of the FQR-T routing policy. To do so, we show that basic FQR, as in Gurvich and Whitt (2007), can perform poorly under normal loads, because it can activate sharing that makes the system overloaded. When customers do not abandon, the system becomes unstable and explodes because of the inefficient sharing. When customers do abandon, the queue lengths stabilize at undesirably high levels. These bad properties are consistent with the positive results in Gurvich and Whitt (2007), because the  $X$  model with service rates  $\mu_{i,j}$  depending upon both  $i$  and  $j$  is explicitly prohibited by the conditions in the theorems of Gurvich and Whitt (2007).

In addition to demonstrating that standard FQR can perform very poorly, we also develop new approximations, like those in the main paper, to describe the performance in this case. We will show that these new approximations accurately predict the extreme performance degradation.

We will focus on the case of a symmetric model with  $r = 1$ . Then FQR reduces to the policy of *servicing the longer queue* (SLQ). It also reduces to our FQR-T control with thresholds set at  $\kappa_{1,2} = \kappa_{2,1} = 1$  for all  $n$ , without imposing the constraint of one-way sharing. Throughout this section we will refer to this policy as the SLQ policy, which is a special case of the serve-the-longest-queue (SLQ) policy when there are more than two queues. For the SLQ policy, it is important to specify how ties are broken, because the control tends to make ties occur quite frequently. Here we are assuming that a server will always serve a customer from his own class if the queue lengths are equal.

The reason that we have to develop new approximations is that now the queue-difference stochastic process  $D(t)$  in (1.1) can visit all three possible regions, which here are  $(-\infty, 1]$ ,  $\{0\}$  and  $[1, \infty)$ . Also, there can now be two-way sharing, so the potential inefficiency can be reached quite easily. Hence we need to develop new fluid equations. We will exploit the symmetry in order to simplify the analysis. We have also developed a more complicated system of equations describing the performance of the asymmetric model when two-way sharing is allowed. The way to do that will be evident from the analysis below.

There are two cases: with and without customer abandonment. We first consider the case of no customer abandonment, and then afterwards the case of customer abandonment. For both, we will give results for a symmetric  $X$  model with parameters:

$$m_i = n, \quad \lambda_i = 0.99n, \quad \mu \equiv \mu_{i,i} = 1, \quad \text{and} \quad \nu \equiv \mu_{1,2} = \mu_{2,1} = 0.8, \quad (1.1)$$

where inefficiency can occur because  $\nu < \mu$  (serving the other class is less efficient). With these parameters, if each service pool served only its own class, then the system would be stable, even without abandonments.

**No Customer Abandonment.** We now show how to analyze such a symmetric  $X$  model with the SLQ routing policy. To do so, we will work in the fluid scaling, dividing by  $n$ . For that purpose, let  $z(t) \equiv Z_{i,i}(t)/n$ , be the *proportion* of agents serving their own class in each of the pools, and let  $q(t) \equiv Q_i(t)/n$ . Since we consider the systems normalized by  $n$ , we take  $\lambda \equiv \lambda_i/n$ , so that in our example above  $\lambda = 0.99$ . Because of the symmetry, we omit the class subscripts. To preserve the symmetry, we assume that the initial conditions are symmetric as well.

We now develop ODE's describing the evolution of  $z(t)$  and  $q(t)$ . The reasoning is somewhat more complicated than before, because now the queue-difference stochastic process  $D(t)$  should visit all three regions  $(-\infty, -\kappa_{2,1}]$ ,  $(-\kappa_{2,1}, \kappa_{1,2})$  and  $[\kappa_{1,2}, \infty)$ , where  $\kappa_{1,2} = \kappa_{2,1} = 1$ . We will obtain significant simplification by exploiting the symmetry.

We first find the time-dependent proportion of time that the two queues are equal. Let  $\pi(t)$  be that

proportion, i.e.,

$$\pi(t) = P(D_t(\infty) = 0), \quad t \geq 0. \quad (1.2)$$

(See §4.1 for more details.) By symmetry, the amount of time queue 1 is bigger than queue 2 is equal to the amount of time queue 1 is smaller than queue 2. Hence,  $(1 - \pi(t))/2$  is the amount of time queue 1 is bigger than queue 2.

We first write down the ODE for  $z$ . It is easy to see that

$$\dot{z}(t) = \pi(t)(1 - z(t))\nu + \frac{1 - \pi(t)}{2} [\nu - z(t)(\mu + \nu)]. \quad (1.3)$$

In equilibrium,  $\dot{z}(\infty) = 0$ , so that we get

$$z \equiv z(\infty) = \frac{\nu(1 + \pi)}{2\nu\pi + (1 - \pi)(\mu + \nu)}, \quad (1.4)$$

where  $\pi \equiv \pi(\infty)$ . In our numerical example with  $\mu = 1$  and  $\nu = 0.8$ , equation (1.4) becomes

$$z = \frac{4 + 4\pi}{9 - \pi}. \quad (1.5)$$

To find the value of  $z$  above, we need to solve for  $\pi$ . We will find an expression for  $\pi$  by approximating the *absolute* difference process between the queues:  $\{|D(t)| : t \geq 0\}$  by a BD process. For all states  $j \geq 1$ , the birth rate is  $\hat{\lambda}_j = \lambda$ , corresponding to an arrival at the larger queue, while the death rate is  $\hat{\mu}_j = \lambda + 2[z\mu + (1 - z)\nu]$ , corresponding to an arrival to the shorter queue, or any service completion (since the newly available agent will take a customer from the larger queue). There is a different birth rate when the two queues are equal. The birth rate (to make either of the queues longer) when the difference is zero is  $\hat{\lambda}_0 = 2\lambda + 2[z\mu + (1 - z)\nu]$ , where  $2\lambda$  corresponds to an arrival to either of the queues, while  $2[z\mu + (1 - z)\nu]$  corresponds to any service completion. Solving for the steady-state of this BD process, we get

$$\pi = \frac{1 - \rho}{1 - \rho + \frac{\hat{\lambda}_0}{\mu}}, \quad (1.6)$$

where

$$\rho \equiv \frac{\hat{\lambda}_j}{\hat{\mu}_j} = \frac{\lambda}{\lambda + 2[z\mu + (1 - z)\nu]}, \quad j \neq 0. \quad (1.7)$$

Hence we get

$$\pi = \frac{z\mu + (1-z)\nu}{\lambda + 2z\mu + 2(1-z)\nu}. \quad (1.8)$$

Solving the two equations (1.5) and (1.8) for  $\pi$  and  $z$  with the rates of our numerical example, we get  $z = 0.61$  ( $Z_{i,i} = 61$ ) and  $\pi = 0.32$ .

Using the values of  $z$  just determined, we can now find the deterministic fluid approximation for the evolution of the queues, after  $z$  achieves its steady-state, and is fixed. In the fluid approximation work flows to and out of the system in constant rate, hence the rate of change in the queue length is the arrival rate minus the departure rate, yielding:

$$\dot{q}(t) = \lambda - z\mu - (1-z)\nu,$$

so that

$$q(t) = q(0) + [\lambda - \nu - (\mu - \nu)z]t, \quad t \geq 0.$$

Plugging the rates from our numerical example, we get

$$\begin{aligned} q(t) &= q(0) + 0.068t, \quad \text{or} \\ Q(t) &= Q(0) + 6.8t, \quad t \geq 0. \end{aligned} \quad (1.9)$$

Figures 6 and 7 show the sample paths of  $Q_1(t)$  and  $Z_{2,1}(t)$ , starting empty, in one simulation run. After an initial transient period, the number of agents serving the other class fluctuates around 40%, while the queue grows in an approximately linear rate. In Figure 7 we plot the number in queue together with the approximating line in (1.9). The queue-length sample path and the straight line are almost indistinguishable.

Table 7 has a comparison of the approximations described above with simulation results. As before, we ran five simulations with five different random seeds. We give the averages of this five simulation runs, together with the half width confidence intervals calculated using a  $t$  random variable with 4 degrees of freedom. Since the system we are considering is completely symmetric, we only show the results for  $Q_1$  and  $Z_{1,1}$ ; the results for  $Q_2$  and  $Z_{2,2}$  are identical.

	$E[Q_1]$ slope	$E[Z_{1,1}]$	$\pi$
Approx.	6.8	61.0	0.32
Sim. results	$\pm 0.4$	$\pm 0.6$	$\pm 0.00$

Table 7: A comparison of approximations for the system performance with SLQ routing to simulation results when there is no customer abandonment.

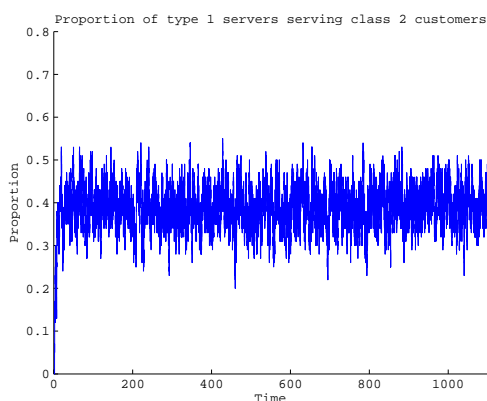


Figure 6: Sample path of  $Z_{2,1}(t)$  with SLQ, but no abandonments.

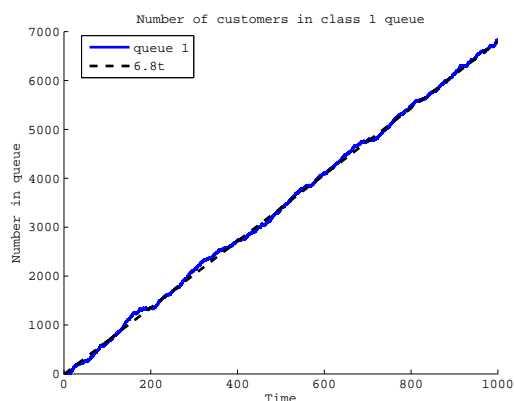


Figure 7: Sample path of  $Q_1(t)$  with SLQ, but no abandonments.

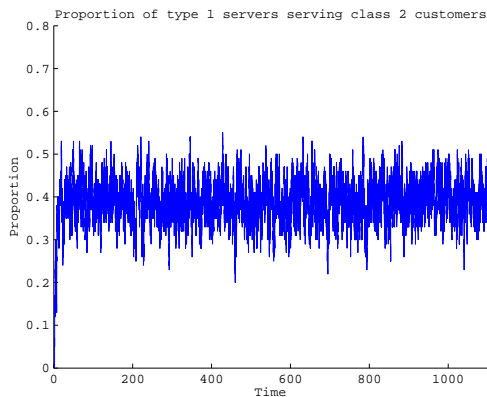


Figure 8: Sample path of  $Z_{2,1}(t)$  with SLQ, with abandonments.

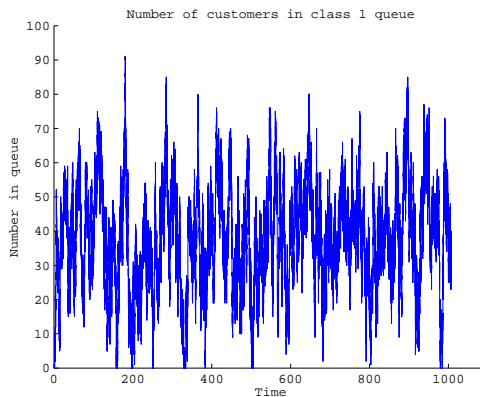


Figure 9: Sample path of  $Q_1(t)$  with SLQ, with abandonments.

**System With Abandonments.** We now consider the case of customer abandonments. When we consider the symmetric model with abandonments, we have a new ODE for the queue length:

$$\dot{q}(t) = \lambda - z(t)\mu - (1 - z(t))\nu - q(t)\theta,$$

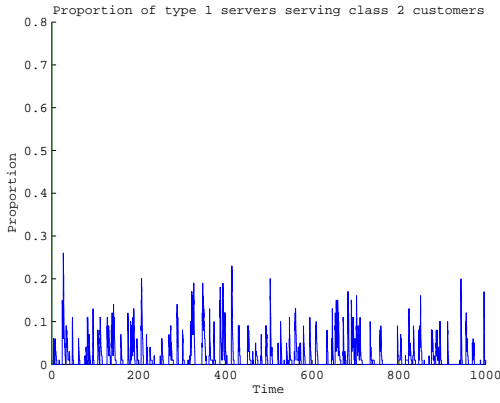


Figure 10: Sample path of  $Z_{2,1}(t)$  with FQR-T, with abandonments.

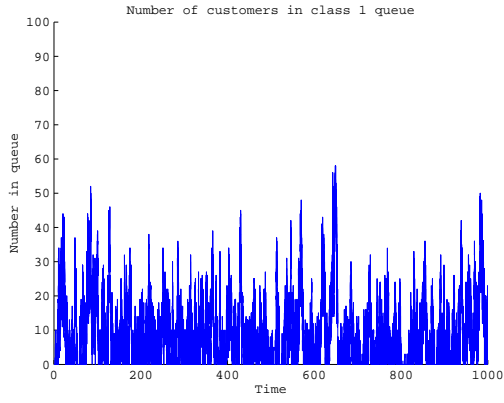


Figure 11: Sample path of  $Q_1(t)$  with FQR-T, with abandonments.

where  $\theta \equiv \theta_i$ ,  $i = 1, 2$  is the abandonment rate for both classes, which we take to be 0.2 in our example.

In steady-state, we have  $\dot{q}(t) = 0$ , and thus, for  $q \equiv q(\infty)$ ,

$$q = \frac{\lambda - z\mu - (1 - z)\nu}{\theta}. \quad (1.10)$$

An initial approximation for  $q$  can use the same value of  $z$  we obtained before without abandonments. If we use that same  $z$  in (1.5), then we get  $q = 0.34$ , which turns out to be quite accurate. However, proceeding more carefully, we can incorporate the abandonment and investigate how it changes the value of  $z$ . We see that it does so through the way it changes the value of  $\pi$ . To do the analysis, we need to consider once again the birth and death rates of the process  $|D(\infty)|$ : Let  $\hat{\lambda}_j$  be the birth rate, and  $\hat{\mu}_j$  be the death rate when the difference between the two queues is  $j$ . We then have

$$\hat{\lambda}_0 = 2\lambda + 2[z\mu + (1 - z)\nu] + 2q\theta,$$

corresponding to an arrival to either of the queues, service completion from either of the service pools or an abandonment from either of the queues. For  $j > 0$ , we have birth rates

$$\hat{\lambda}_j = \lambda + q\theta,$$

corresponding to an arrival to the longer queue, or an abandonment from the shorter queue, while the death rates are

$$\hat{\mu}_j = \lambda + 2[z\mu + (1 - z)\nu] + q\theta,$$

corresponding to an arrival to the shorter queue, service completion from either of the service pools, or an abandonment from the longer queue.

Solving the balance equations of the BD process gives us an expression for  $\pi$  in terms of  $z$  and  $q$ . Once again we get

$$\pi = \frac{1 - \rho}{1 - \rho + \frac{\hat{\lambda}_0}{\hat{\mu}}},$$

but where  $\rho$  is redefined as

$$\rho \equiv \frac{\lambda + q\theta}{\lambda + 2[z\mu + (1 - z)\nu] + q\theta}.$$

Hence

$$\pi = \frac{z\mu + (1 - z)\nu}{\lambda + 2z\mu + 2(1 - z)\nu + q\theta}.$$

From equation (1.10), we get  $q\theta = \lambda - z\mu - (1 - z)\nu$ , thus

$$\pi = \frac{z\mu + (1 - z)\nu}{2\lambda + z\mu + (1 - z)\nu}. \quad (1.11)$$

Solving the two equations (1.4) and (1.11) in the two unknowns  $\pi$  and  $z$ , we get  $z = 0.607$  and  $\pi = 0.318$  in our numerical example. Plugging that value of  $z$  in equation (1.10), we get  $q = 0.343$ , or  $Q_i = 34.3$  and  $Z_{i,i} = 60.7$ . On average, 39.3 agents are serving customers from the other class, hence the total service rate of each class reduces from  $100\mu = 100$  to  $60.7\mu + 39.3\nu = 92.14$ , which is less than the arrival rate  $\lambda_i = 99$ . That explains why the system becomes congested.

In Table 8 we compare these new approximations to simulations. As before, we ran five independent simulations, and use the  $t$  distribution with 4 degrees of freedom to construct the confidence intervals.

	$E[Q_1]$	$E[Z_{1,1}]$	$\pi$
Approx.	34.3	60.7	0.32
Sim.	34.3	61.0	0.34
results	$\pm 0.8$	$\pm 0.0$	$\pm 0.01$

Table 8: A comparison of approximations for the system performance with SLQ routing to simulation results when there are customer abandonments.

Figures 8 and 9 show the sample paths of  $Q_1(t)$  and  $Z_{2,1}(t)$  taken from one simulation run. For contrast, in Figures 10 and 11 we also show the sample paths of  $Q_1(t)$  and  $Z_{2,1}(t)$  in the same system, but with

the FQR-T control using  $\kappa_{i,j} = 10$ . With FQR-T, we get  $E[Z_{1,2}] = 2.0$  and  $E[Q_1] = 9.4$ , so that the average service rate for class  $i$  is now  $98\mu + 2\nu = 99.6$ , which is larger than the arrival rate  $\lambda_1$ . Hence, with FQR-T the system remains normally loaded, even though there is some sharing.

**SLQ with One-Way Sharing** We have seen that performance degrades seriously if we drastically reduce the thresholds and eliminate the one-way sharing. It is natural to wonder what happens if we only reduce the thresholds, keeping one-way sharing. We find that the performance is not nearly as bad when we impose one-way sharing, but it still degrades significantly. We now illustrate that.

For the example above, we see that the total arrival rate is  $\lambda = \lambda_1 + \lambda_2 = 198$ , while the total rate out is 200 without sharing, but with sharing the total rate out is  $200 - 0.02(Z_{1,2}(t) + Z_{2,1}(t))$ . We thus see that the total rate in actually exceeds the total rate out whenever  $Z_{1,2}(t) + Z_{2,1}(t) > 10$ . The traffic intensity varies from 0.99 with no sharing at all to  $198/180 = 1.10$  with full sharing. We have yet to mathematically analyze the performance in this case, so we rely on simulation.

Figure 12 shows the class-1 queue-length process without abandonments over a long time interval, in particular, for  $t = 25,000$ , which corresponds to  $5 \times 10^6$  arrivals to both queues. Without abandonments, it is unclear whether the system is stable or not, but there is clearly significant congestion. We estimate that  $Z_{1,2} = Z_{2,1} \approx 3.6$ , indicating that the system is close to the critical boundary case. In Figure 13 we show both  $Z_{i,j}$  processes, during a short time interval, to make it easy to observe the way the two processes oscillate.

We are also interested in the way the system behaves if we incorporate abandonments. For this purpose we add an exponential patience distribution with rate  $\theta_i = 0.2$ , for every customer from both classes. Figure 14 shows a sample path of  $Q_1(t)$ , and figure 15 shows a sample path of  $Z_{2,1}(t)$ . The figures suggest that one-way sharing is not much worse than FQR-T, at least when  $n = 100$ . Yet, for larger systems, as the thresholds  $\kappa_{i,j}$  become larger, there will be less sharing in the balanced loading, and the advantages of the FQR-T control will become more apparent. Simulation results for this case are shown in table 9. The amount of sharing and the mean queue length are only slightly larger than the

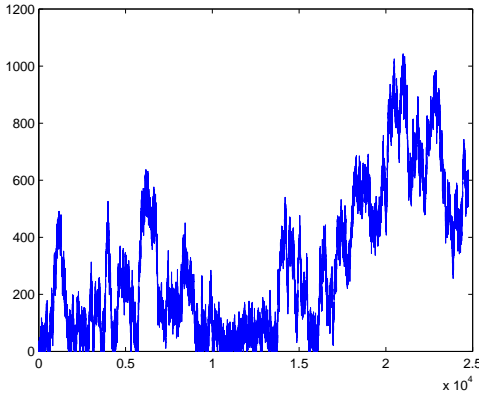


Figure 12: The queue-length process at queue 1 with SLQ modified by one-way sharing when there are no abandonments.

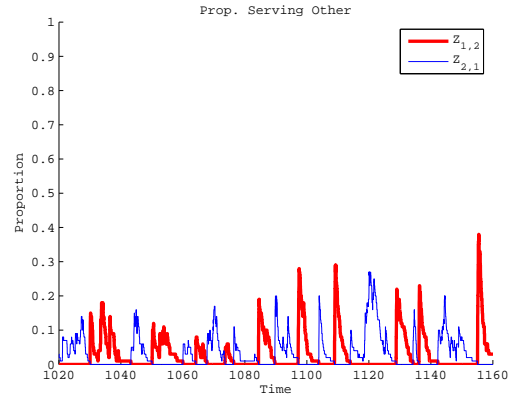


Figure 13: A plot of the  $Z_{i,j}(t)$  processes in a short time scale with SLQ modified by one-way sharing when there are no abandonments.

estimates for FQR-T given above.

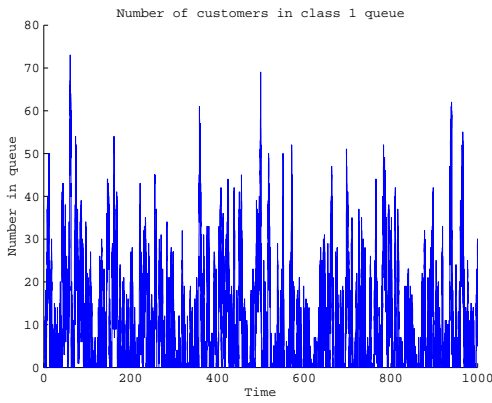


Figure 14: The queue-length process at queue 1 with SLQ modified by one-way sharing when there are abandonments.

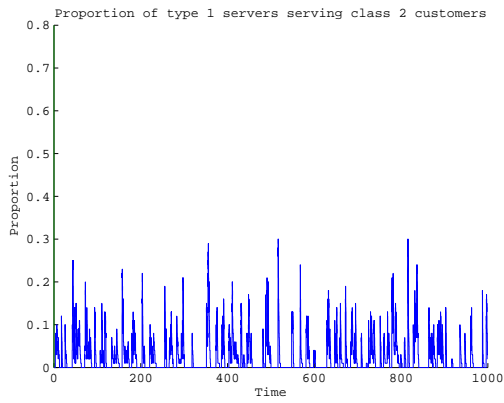


Figure 15: The  $Z_{2,1}(t)$  process with SLQ modified by one-way sharing when there are abandonments.

	$E[Q_1]$	$E[Z_{2,1}]$
Average	11.0	2.8
conf. int.	$\pm 0.8$	$\pm 0.2$

Table 9: Simulation results for the SLQ using one way sharing, when there are customers abandonments with rate  $\theta = 0.2$ .

## B. The Advantage of Lower Thresholds

It turns out that the one-way sharing restriction can cause problems in very large systems, making it take too long to shift from sharing in one direction to share in the opposite direction when that becomes desirable. To avoid that difficulty, we can also include lower thresholds  $\tau_{1,2}$  and  $\tau_{2,1}$ : An available type-2 agent is allowed to serve a class-1 customer only if the proportion of type-1 agents serving class-2 customers is below  $\tau_{2,1}$  (and of course  $D(t) \geq \kappa_{1,2}$ ). And similarly in the other direction. It suffices for these lower thresholds to be quite small, e.g., about 1% of the number of servers.

Suppose the system is initialized with  $n$  servers from service-pool 1 serving class-2 customers, but class-1 is more overloaded than class-2. (This can happen if the arrival rates change suddenly.) In that circumstance, queue 1 may grow well above queue 2, but we have to wait until there are no longer class-2 customers in pool-1 before sharing can be activated. The mean time to wait until pool-1 has no more class-2 customers is

$$\sum_{j=1}^n \frac{1}{j \cdot \mu_{2,1}} \approx \frac{\log(n)}{\mu_{2,1}} \rightarrow \infty \quad \text{as } n \rightarrow \infty. \quad (2.1)$$

To activate sharing more quickly, we can modify the FQR-T control to include *lower thresholds*, as specified above. The importance of the lower thresholds can be seen in Figures 16 and 17. These figures show simulation results of an extreme example, to illustrate the value of the lower thresholds in large systems. The parameters for this simulation are

$$\begin{aligned} m_1 &= m_2 = 1000, & \lambda_1 &= 1200, & \lambda_2 &= 990, & \mu_{1,1} &= \mu_{2,2} = 1, & \mu_{1,2} &= \mu_{2,1} = 0.5, \\ \kappa_{1,2} &= \kappa_{2,1} = 100, & \text{and } r &= 1. \end{aligned}$$

With these parameters, queue 1 is overloaded, while queue 2 is underloaded. To respond to that unbalanced overload, we should have  $Z_{1,2} > 0$  and  $Z_{2,1} = 0$ . However, we initialize the system with sharing in the opposite way. Indeed, we consider an extreme example in which *all* of service pool 1 is initially busy with customer from class-2, and none of the type-2 agents are busy serving class-1 customers. We are interested in the time it takes the stochastic process  $Z_{2,1}(t)$  to reach 0, so that the desired sharing can begin. With lower thresholds of only  $\tau_{1,2} = \tau_{2,1} = 0.01$ , that time is reduced from about 21 mean

service times to about 9 service times. Thus, clearing the last 1% without lower thresholds takes more than half the time.

From Figures 16 and 17, it is also easy to see what happens in less extreme cases, such as we have been considering in the main paper. For example, consider the base case discussed in §6 with results in Table 1. In that case with  $\lambda_1 = 1.3n$  and  $\lambda_2 = 0.9n$ , we have  $Z_{1,2} \approx 0.2n$ . Hence, it is reasonable to assume that overload in one direction might lead to about 20% of the agents in pool 2 serving class-1 customers.

From that perspective, we might consider starting with only 20% sharing in the wrong direction. From Figures 16 and 17, we see that, without a lower threshold, the time to activate sharing in the right direction is about  $21 - 4 = 17$ . In contrast, with lower thresholds, it is about  $9 - 4 = 5$ . When we start with a lower percentage of agents sharing the wrong way, the difference becomes even more dramatic, because we eliminate a common initial period (here of length 4).

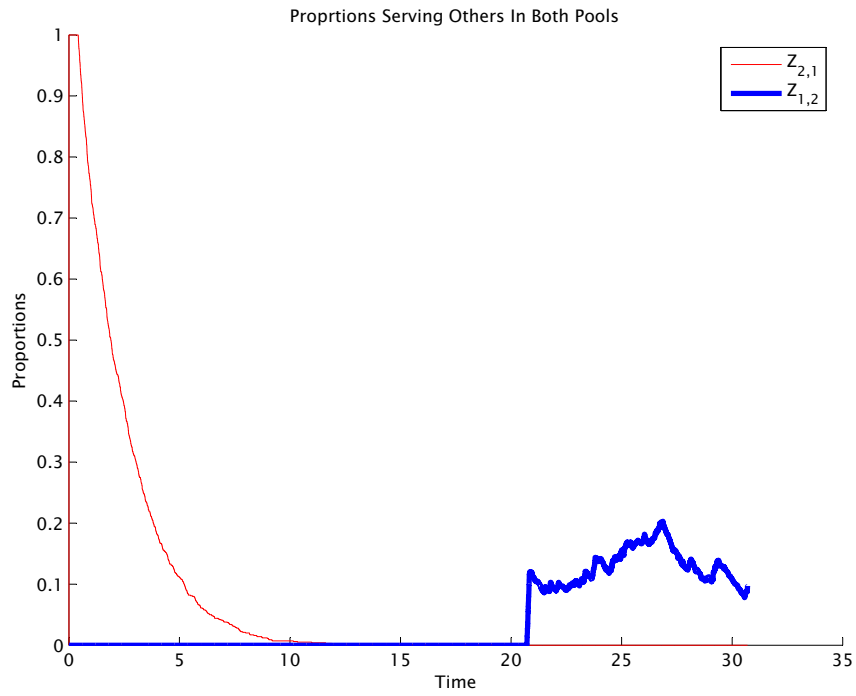


Figure 16: Sample paths of  $Z_{1,2}(t)$  and  $Z_{2,1}(t)$  initialized incorrectly, without lower thresholds.

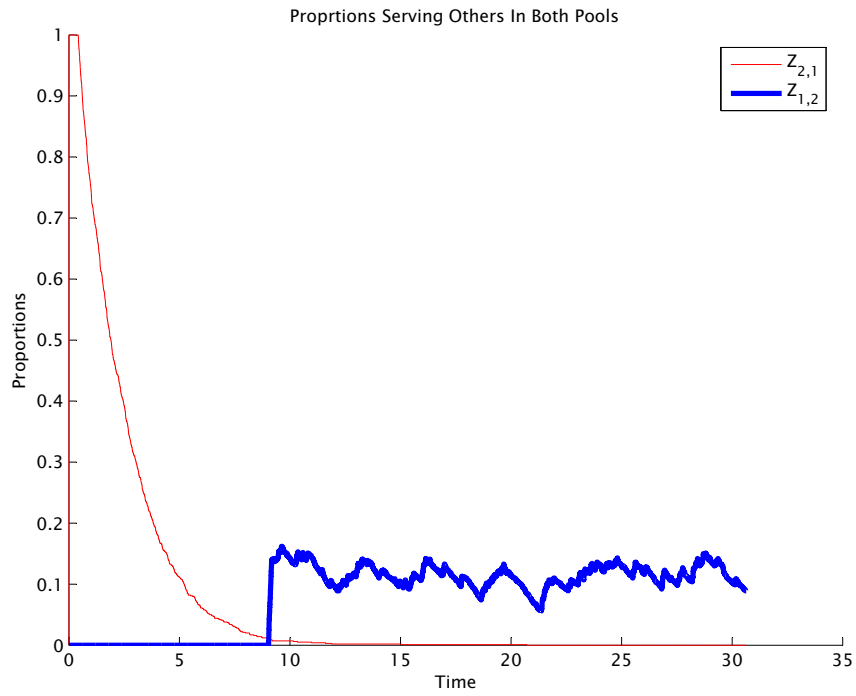


Figure 17: Sample paths of  $Z_{1,2}(t)$  and  $Z_{2,1}(t)$  initialized incorrectly, with lower thresholds  $\tau_{1,2} = \tau_{2,1} = 0.01$ .

## C. More Comparisons with Simulations

In this section we present additional simulation results in order to give a better picture of the way that the FQR-T routing policy performs.

### C.1. Different Primary Service Rates

In the main paper we assumed that the primary service rates for the two classes are identical, i.e., that  $\mu_{1,1} = \mu_{2,2}$ . Here we perform simulations to show what happens when they are not equal. In particular, here we assume that

$$\mu_{1,1} = 1 < 2 = \mu_{2,2}. \quad (3.1)$$

There are different cases, depending on what we assume for the service rates for serving the other class. Indeed, there are two main cases: we can assume that service-pool 2 is uniformly faster or we can assume that class-1 tasks are uniformly harder (take longer). That is, the differences can be determined primarily by the agents or primarily by the customers. We consider those two cases in turn. In all cases, we consider variants of the same base case with  $n = 100$ . In particular, we have

$$m_1 = m_2 = 100, \quad \theta_1 = \theta_2 = 0.2 \quad \text{and} \quad \kappa_{1,2} = \kappa_{2,1} = 10. \quad (3.2)$$

We choose the service rates  $\mu_{i,j}$  to represent different cases, and we choose the arrival rates  $\lambda_i$  to make one class overloaded and the other class underloaded, just as in the main case with unbalanced overloads.

#### C.1.1. One Service Pool Is Uniformly Faster

In some cases, one service pool might be faster than the other since it may consist of better trained agents. To represent this first case, we let

$$\mu_{1,2} = 1.6 \quad \text{and} \quad \mu_{2,1} = 0.8 \quad (3.3)$$

There are now two further subcases, depending on which class is overloaded. In the first subcase, class 1 is overloaded, while pool-2 is normally loaded. In particular, we let  $\lambda_1 = 130$  and  $\lambda_2 = 190$ . Note that since  $\mu_{2,2} = 2$ , service pool 2 is indeed underloaded with this arrival rate. In the second subcase

we let class 2 be the overloaded one. To achieve that, we let  $\lambda_1 = 90$  and  $\lambda_2 = 230$ . The results are shown in Table 10.

fluid	$Q_1$ overloaded			$Q_2$ overloaded		
perf. meas.	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.
$E[Q_1]$	65.6	61.7	64.2 $\pm 2.3$	55.6	59.8	59.2 $\pm 2.2$
$E[Q_2]$	55.6	60.4	59.9 $\pm 2.7$	65.6	62.2	62.3 $\pm 2.2$
$E[Z_{1,2}]$	10.6	11.0	11.1 $\pm 0.2$	21.1	21.9	21.8 $\pm 0.5$
distribution	$Q_1$ overloaded			$Q_2$ overloaded		
perf. meas.	approx.		sim.	approx.		sim.
$std(Q_\Sigma)$	40.0		38.8 $\pm 2.8$	40.0		39.4 $\pm 3.3$
$std(Q_1)$	20.0		20.5 $\pm 1.7$	20.0		19.7 $\pm 1.7$
$std(Q_2)$	20.0		20.8 $\pm 1.5$	20.0		20.7 $\pm 1.6$

Table 10: A comparison of the fluid approximations for the steady-state performance measures with simulation results when pool-2 agents are uniformly faster. In both cases  $\mu_{1,1} = 1$ ,  $\mu_{2,2} = 2$ ,  $\mu_{1,2} = 1.6$ ,  $\mu_{2,1} = 0.8$ ,  $\theta_i = 0.2$  and  $\kappa_{i,j} = 10$ . On the LHS  $Q_1$  is overloaded:  $\lambda_1 = 130$ ,  $\lambda_2 = 190$ . On the RHS  $Q_2$  is overloaded:  $\lambda_1 = 90$ ,  $\lambda_2 = 230$ .

### C.1.2. Service of One Class Takes Uniformly Longer

We now consider a system in which class-1 customers are harder to handle; they require more service time on average. In this case we let

$$\mu_{1,2} = 0.8 \quad \text{and} \quad \mu_{2,1} = 1.6 \quad (3.4)$$

Again there are two further subcases, depending on which class is overloaded. In the first subcase, class 1 is overloaded, while pool-2 is normally loaded:  $\lambda_1 = 130$  and  $\lambda_2 = 190$ . In the second subcase, class 2 is overloaded, while class 1 and pool-1 are normally loaded:  $\lambda_1 = 90$  and  $\lambda_2 = 230$ . The results are shown in table 11 below.

An important observation is that sharing in the first case, when  $Q_1$  is overloaded, is actually worse than

not sharing at all from the perspective of total queue length. Without sharing,  $Q_1 \approx 150$  and  $Q_2 \approx 0$ , thus the proportion of customers lost due to abandonments is approximately  $\theta_1 Q_1 = 0.2 \cdot 150 = 30$ . In contrast, with sharing, both queues are bigger than 90, thus the proportion of customers lost is larger than  $0.2 \cdot 180 = 36$ . Moreover, the total queue length is larger.

Another observation is that the variances when  $Q_1$  is overloaded are higher than the approximation, whereas the variances in the other case are smaller. These two features tend to make sharing in the first case undesirable. In cases like this we might want to have different thresholds, to make sure we share quickly when  $Q_2$  is overloaded, and possibly not share at all when  $Q_1$  is overloaded.

fluid	$Q_1$ overloaded			$Q_2$ overloaded		
perf. meas.	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.
$E[Q_1]$	95.7	91.2	93.6 $\pm 1.6$	23.1	25.6	26.6 $\pm 1.4$
$E[Q_2]$	85.7	95.2	94.3 $\pm 0.7$	33.1	29.1	32.3 $\pm 1.2$
$E[Z_{1,2}]$	13.6	14.5	14.4 $\pm 0.2$	14.6	15.1	15.0 $\pm 0.4$
distribution	$Q_1$ overloaded		$Q_2$ overloaded			
perf. meas.	approx.	sim.	approx.	sim.		
$std(Q_\Sigma)$	40.0	41.1 $\pm 1.0$	40.0	33.3 $\pm 1.4$		
$std(Q_1)$	20.0	20.6 $\pm 0.3$	20.0	16.5 $\pm 0.7$		
$std(Q_2)$	20.0	21.8 $\pm 0.5$	20.0	18.4 $\pm 0.7$		

Table 11: A comparison of the fluid approximations for the steady-state performance measures with simulation results when class-1 customers take longer to serve. In both cases  $\mu_{1,1} = 1$ ,  $\mu_{2,2} = 2$ ,  $\mu_{1,2} = 0.8$ ,  $\mu_{2,1} = 1.6$ ,  $\theta_i = 0.2$  and  $\kappa_{i,j} = 10$ . On the LHS  $Q_1$  is overloaded:  $\lambda_1 = 130$ ,  $\lambda_2 = 190$ . On the RHS  $Q_2$  is overloaded:  $\lambda_1 = 90$ ,  $\lambda_2 = 230$ .

## C.2. Extreme Differences Between The Two Classes

The remaining simulation experiments are designed to test the limits of the FQR-T control. First, we see how well our approximations perform when the classes are very different. To illustrate, here we let

the abandonment rates for the two classes be very different. In particular, now we assume that  $\theta_1 \gg \theta_2$ . (Recall that our diffusion approximations in §7 exploited equal abandonment rates in order to justify an exact analysis.) In the numerical example in §8 we saw that in the overloaded case, when  $\mu_{i,j} \neq \mu_{i,i}$ , our normal approximations for the steady-state distributions were quite a good approximation for the true distributions of the queues.

To see what happens with very different abandonment rates, we modify the base case by letting  $\theta_1 = 1.0$  and  $\theta_2 = 0.1$ . The numerical example we consider has the following rates:

$$\lambda_1 = 1.3n, \quad \lambda_2 = 0.9n, \quad \mu_{i,i} = 1, \quad \mu_{i,j} = 0.8, \quad \theta_1 = 1, \quad \theta_2 = 0.1 \quad \text{and} \quad \kappa_{i,j} = 0.1n. \quad (3.5)$$

In Figures 18 and 19 we show histograms of the distributions of  $Q_1$  and  $Q_2$ , respectively. Two features appear in the histograms, which did not appear in the previous examples. First, both queues have a mass at zero. Second, the distribution of  $Q_1$  changes at a neighborhood of  $Q_1 = 10$ , i.e., when  $Q_1 \approx \kappa_{1,2}$ . This jump in the distribution of  $Q_1$  occurs because  $Q_2$  has a large mass at zero. At such times,  $Q_1$  tends to be in the neighborhood of  $\kappa_{1,2}$ . That is when customers from  $Q_1$  are sent to service pool 2.

These two features are not accounted for in our approximations, and thus our approximations in this case are not as accurate as for previous cases. Nevertheless, as can be seen from the simulation results in Tables 12 and 13, the approximations work remarkably well. The standard-deviation approximations are very similar to the simulation results. It is just when we take a closer look at the distributions, and consider the quantiles, that we see our approximations are not nearly exact, especially for  $Q_2$ , which has a large mass at zero, hence has a “less normal” distribution.

We should point out that the degradation in the performance of the approximation here is largely due to class 1 becoming much less overloaded. Reasoning as for (3.1), we see that without any sharing the class-1 queue length would be  $Q_1 \approx 150$  when  $\theta_1 = 1.0$ , but only  $Q_1 \approx 30$  when  $\theta_1 = 0.1$ . The approximation would perform much better if we had taken the “easier case” with the abandonment rates in (3.5) switched to  $\theta_1 = 0.1$  and  $\theta_2 = 1.0$ , because then the system would have been even more overloaded.

Since  $D$  receives only integer values, we take the linear interpolation to approximate its distribution.

See §8 for more details.

	n=25			n=100			n=400		
perf. meas.	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.	2 equ.	3 equ.	sim.
$E[Q_1]$	5.3	4.7	6.0 $\pm 0.0$	21.1	20.3	20.9 $\pm 0.4$	84.4	83.6	83.9 $\pm 1.9$
$E[Q_1/n]$	0.211	0.19	0.24 $\pm 0.0$	0.211	0.203	0.209 $\pm 0.004$	0.211	0.209	0.209 $\pm 0.004$
$E[Q_2]$	2.3	10.5	6.9 $\pm 0.1$	11.1	20.9	19.2 $\pm 0.4$	44.4	55.0	54.6 $\pm 2.0$
$E[Q_2/n]$	0.111	0.42	0.27 $\pm 0.01$	0.111	0.201	0.192 $\pm 0.004$	0.111	0.137	0.136 $\pm 0.005$
$E[D]$	–	–5.9	–0.9 $\pm 0.1$	–	–0.6	1.7 $\pm 0.3$	–	28.7	29.3 $\pm 0.3$
$\kappa_{1,2} - E[D]$	–	15.9	3.9 $\pm 0.1$	–	10.6	8.3 $\pm 0.3$	–	11.3	10.7 $\pm 0.3$
$E[Z_{1,2}]$	2.7	3.5	3.1 $\pm 0.0$	11.1	12.1	12.0 $\pm 0.4$	44.4	45.5	44.9 $\pm 1.0$
$E[Z_{1,2}/n]$	0.111	0.14	0.12 $\pm 0.00$	0.111	0.121	0.120 $\pm 0.004$	0.111	0.114	0.112 $\pm 0.003$

Table 12: A comparison of the fluid approximations for the steady-state performance measures with simulation results with very different abandonment rates. Here,  $\lambda_1 = 1.3n$ ,  $\lambda_2 = 0.9n$ ,  $\mu_{1,1} = \mu_{2,2} = 1$ ,  $\mu_{1,2} = 0.8$ ,  $\theta_1 = 0.1$ ,  $\theta_2 = 1$  and  $\kappa_{1,2} = 0.1n$ .

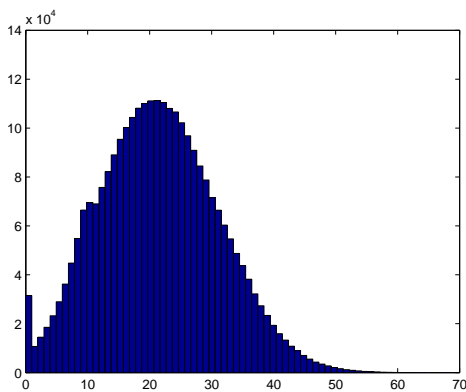


Figure 18: Histogram for  $Q_1$  when  $\theta_1 \gg \theta_2$

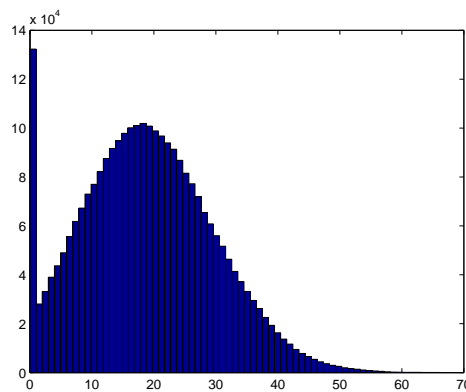


Figure 19: Histogram for  $Q_2$  when  $\theta_1 \gg \theta_2$

		n=25		n=100		n=400	
perf. meas.		Approx.	Sim.	Approx.	Sim.	Approx.	Sim.
$std(Q_\Sigma)$		10	9.1 $\pm 0.1$	20	19.7 $\pm 0.7$	40	39.9 $\pm 2.2$
$std(\hat{Q}_\Sigma)$		2	1.8	2	1.97	2	2.0
$std(Q_1)$		5	4.9 $\pm 0.1$	10	9.9 $\pm 0.2$	20	19.8 $\pm 1.0$
$std(\hat{Q}_1)$		1	1.0	1	0.099	1	1.0
$std(Q_2)$		5	6.2 $\pm 0.1$	10	11.6 $\pm 0.3$	20	21.5 $\pm 1.1$
$std(\hat{Q}_2)$		1	1.2	1	0.116	1	1.1
$\hat{Q}_1$ quantiles	0.05	-1.65	-1.21 $\pm 0.01$	-1.65	-1.55 $\pm 0.04$	-1.65	-1.62 $\pm 0.06$
	0.25	-0.68	-0.81 $\pm 0.01$	-0.68	-0.45 $\pm 0.79$	-0.68	-0.43 $\pm 0.07$
	0.75	0.68	0.60 $\pm 0.01$	0.68	0.64 $\pm 0.04$	0.68	0.66 $\pm 0.03$
	0.95	1.65	1.8 $\pm 0.01$	1.65	1.68 $\pm 0.09$	1.65	1.66 $\pm 0.12$
$\hat{Q}_2$ quantiles	0.05	-1.65	-1.38 $\pm 0.02$	-1.65	-1.90 $\pm 0.06$	-1.65	-1.41 $\pm 0.86$
	0.25	-0.68	-1.10 $\pm 0.11$	-0.68	-0.86 $\pm 0.04$	-0.68	-0.75 $\pm 0.06$
	0.75	0.68	0.82 $\pm 0.02$	0.68	0.75 $\pm 0.07$	0.68	0.70 $\pm 0.06$
	0.95	1.65	2.30 $\pm 0.13$	1.65	2.07 $\pm 0.07$	1.65	1.80 $\pm 0.12$
centered $D$ quantiles	0.05	-28.5	-15.6 $\pm 0.7$	-33.5	-25.2 $\pm 0.6$	-36.5	-32.6 $\pm 1.1$
	0.25	-13.5	-7.0 $\pm 0.0$	-15.5	-12.4 $\pm 0.7$	-16.5	-15.4 $\pm 0.7$
	0.75	-2.5	-1.0 $\pm 0.0$	-3.5	-2.0 $\pm 0.0$	-3.5	-3.0 $\pm 0.0$
	0.95	0.5	6.6 $\pm 0.7$	-0.5	1.0 $\pm 0.0$	-0.5	0.0 $\pm 0.0$

Table 13: A comparison of the fluid approximations for the steady-state performance measures with simulation results with very different abandonment rates. Here,  $\lambda_1 = 1.3n$ ,  $\lambda_2 = 0.9n$ ,  $\mu_{1,1} = \mu_{2,2} = 1$ ,  $\mu_{1,2} = 0.8$ ,  $\theta_1 = 1$ ,  $\theta_2 = 0.1$  and  $\kappa_{1,2} = 0.1n$ .

### C.3. Challenging Intermediate Cases

More challenging cases occur when the parameter values put the system on the boundary between when sharing is desired and not desired. In this section we consider such a boundary case. To do so, we suppose that  $Q_1^{alone} \approx \kappa_{1,2}$  while  $Q_2$  is critically (normally, but heavily) loaded. This scenario can be regarded as an intermediate case, because we should have sharing if  $Q_1^{alone} > \kappa_{1,2}$ , while we should not have sharing if  $Q_1^{alone} < \kappa_{1,2}$ . We thus should anticipate that neither SSC nor the independent-queue approximation will be especially accurate.

The specific model we consider has  $n = 400$  with  $m_i = n = 400$  servers in each service pool, and the following parameters:

$$\lambda_1 = 441, \quad \lambda_2 = 398, \quad \mu_{i,i} = 1, \quad \mu_{i,j} = 0.8, \quad \theta_i = 1 \quad \text{and} \quad \kappa_{i,j} = 40. \quad (3.6)$$

Note that a simplified fluid approach would consider this system as one with spare capacity, just as in §6.3, since service-pool 2 has two extra servers that can potentially serve 1.6 class-1 customers per unit time, whereas  $Q_1$  has just one “extra arrival” per unit time (when we consider the fact that  $Q_1$  must be at least 40 before the sharing is activated). However, unlike the case we have considered in §6.3,  $Q_2$  is critically loaded, and thus becomes overloaded when class-1 customers are served in service-pool 2.

Figures 20 and 21 show histograms of the distributions of the two steady-state queue lengths. We see that both distributions have a mass at zero, and are far from normal. In Figure 23 we reduce the vertical axes to make it easier to observe the shape of the distribution of  $Q_2$ . Figure 22 is a plot of the sample paths of the two queue-length processes over a short time interval, both centered about their steady-state means. We observe that even in this case there is a strong dependency between the two queues, and that the SSC assumption is not far from reality. In fact, it seems that when both queues are positive, they move together. It is only only when  $Q_2(t) = 0$  and  $Q_1(t) > 0$  that  $Q_1(t)$  moves separately.

With these parameters in (3.6), we see that the two-equation fluid approximation in (3.5) fails badly. First, we cannot find the desired fluid approximations for the  $Q_i$  and  $Z_{1,2}$  using the two equations in (3.5), since the system is operating in the spare-capacity regime. Indeed, if we use (3.5), then we get

$Q_1 = 39.7$  and  $Q_2 = -0.3$ . It is also easy to see that the spare-capacity approximations do not apply here. If we use equation (3.7), then we get that  $Z_{1,2} = 2.5$  which makes class-2 overloaded, and so there is no spare capacity in service pool 2. We can modify (3.7), and assume  $Q_1 = 40$  (and not 39) since pool-2 is heavily loaded. This will give us  $Z_{1,2} = 1.25$  and  $Q_2 = 0$ . However, that result is far from the simulation results, as can be seen in Table 14.

On the other hand, we see that the three-equation approximation in (5.2) actually yields something reasonable. It is in cases like this that we really see the value of the more complex three-equation approximation in (5.2). Here this refined approximation is needed in order to obtain a reasonable approximation.

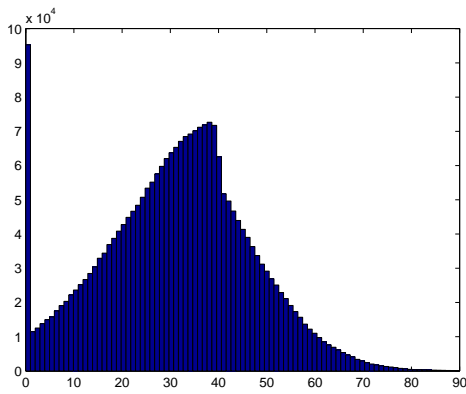


Figure 20: A histogram of  $Q_1$  in the intermediate case.

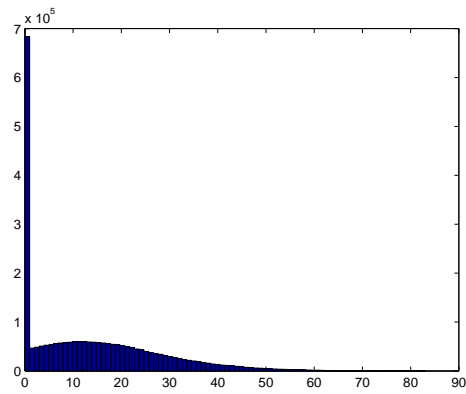


Figure 21: A histogram of  $Q_2$  in the intermediate case.

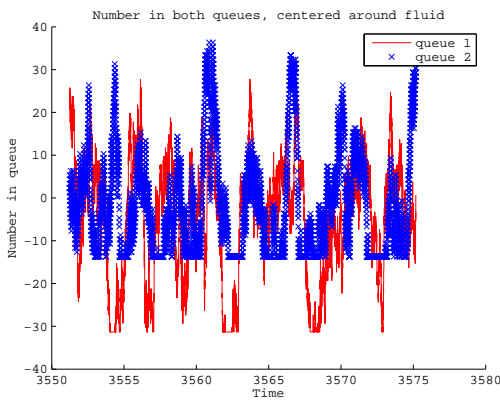


Figure 22: A plot of the queues centered about their fluid.

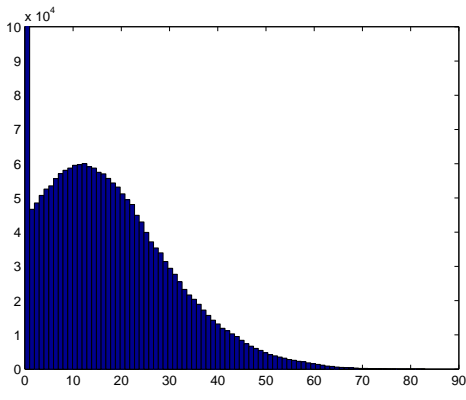


Figure 23: A closer look at  $Q_2$  in the intermediate case.

fluid parameters	spare	3 equ.	sim
$E[Q_1]$	40	27.1	31.5 $\pm 0.7$
$E[Q_2]$	0	15.4	14.2 $\pm 1.0$
$E[Z_{1,2}]$	1.6	17.4	12.6 $\pm 0.4$
distribution	spare	SSC	sim.
$std(Q_\Sigma)$	20.5	29.0	24.4 $\pm 0.6$
$std(Q_1)$	20.5	14.5	15.6 $\pm 0.5$
$std(Q_2)$	—	14.5	13.7 $\pm 0.3$

Table 14: A comparison of the fluid approximations with simulation results for the steady-state performance measures in the intermediate case. In the “spare” column we solve equation (3.7) with a slight modification, taking  $Q_1 = 40$  as described above. This makes  $Q_2 = 0$ , and  $Q_\Sigma = Q_1$ , hence both have the same standard-deviations.

**Lower Arrival Rates.** The effectiveness of the three-equation approximation in the boundary case with  $\lambda_1 = 441$  shows that it should also be not too bad for even lower arrival rates. We look at that now. Table 15 below gives results for three cases with lower arrival rates for class 1. In all three cases, we have kept the same parameter values as in (3.6), except that we change  $\lambda_1$ . Now we consider  $\lambda_1 = 430, 420$  and  $415$ . As the load on  $Q_1$  becomes smaller, the three-equation approximation, and the SSC assumption, become less accurate. Overall, we see that the three-equation fluid approximation for  $E[Q_1]$  and the SSC standard-deviation approximations work pretty well at the boundary ( $\lambda_1 = 441$ ) and even slightly below the boundary ( $\lambda_1 = 430$ ), but then they deteriorate. However, the independent-queue approximation is then good for  $E[Q_1]$ .

For  $\lambda_1 = 415$ , it seems that the independent assumption gives better approximations for the distributions. In the table we also include the value of  $E[Z_{2,1}]$  since as the loads get smaller, we start seeing more sharing in the “wrong” direction. This makes our approximations even less accurate, since we assume that  $Z_{2,1} = 0$  in our approximations.

For the standard deviations, the SSC approximations remain pretty good for the individual queues, while the independent approximation is pretty good for the total queue length. Although  $Q_2$  operates in the OED regime when both queues are independent, we approximate its fluid at zero, hence we approximate its standard deviation as being zero. We could do better in the independent case, using the QED approximations for  $Q_2$  from Garnett et al. (2002). That would evidently make the independent approximations perform well for  $\lambda_1 = 415$ .

fluid	$\lambda_1 = 430$			$\lambda_1 = 420$			$\lambda_1 = 415$		
perf. meas.	ind.	3 equ.	sim.	ind.	3 equ.	sim.	ind.	3 equ.	sim.
$E[Q_1]$	30	18.8	24.9 $\pm 1.0$	20	9.8	18.2 $\pm 1.1$	15	7.7	15.9 $\pm 1.1$
$E[Q_2]$	0	12.0	10.8 $\pm 0.5$	0	2.1	8.7 $\pm 0.6$	0	3.8	8.6 $\pm 0.8$
$E[Z_{1,2}]$	0	14.0	8.1 $\pm 0.8$	0	4.1	4.4 $\pm 0.6$	0	5.8	3.1 $\pm 0.3$
$E[Z_{2,1}]$	0	0	0.07 $\pm 0.05$	0	0	0.19 $\pm 0.12$	0	0	0.34 0.14
distribution	$\lambda_1 = 430$			$\lambda_1 = 420$			$\lambda_1 = 415$		
perf. meas.	ind.	SSC	sim.	ind.	SSC	sim.	ind.	SSC	sim.
$std(Q_\Sigma)$	20.1	28.8	22.6 $\pm 0.7$	20.5	28.6	19.8 $\pm 0.9$	20.4	28.5	20.2 $\pm 1.0$
$std(Q_1)$	20.1	14.4	15.4 $\pm 0.3$	20.5	14.3	14.4 $\pm 0.5$	20.4	14.3	14.4 $\pm 0.3$
$std(Q_2)$	0	14.4	12.8 0.6	0	14.3	11.4 $\pm 0.5$	0	14.3	11.9 $\pm 0.7$

Table 15: A comparison of the fluid approximations for the steady-state performance measures based on the three equations in (5.2) with simulation results with reduced arrival rates for class 1.