Stochastic Systems

ACHIEVING RAPID RECOVERY IN AN OVERLOAD CONTROL FOR LARGE-SCALE SERVICE SYSTEMS

By Ohad Perry and Ward Whitt

We consider an automatic overload control for two large service systems modeled as multi-server queues, such as call centers. We assume that the two systems are designed to operate independently, but want to help each other respond to unexpected overloads. The proposed overload control automatically activates sharing (sending some customers from one system to the other) once a ratio of the queue lengths in the two systems crosses an activation threshold (with ratio and activation threshold parameters for each direction). To prevent harmful sharing, sharing is allowed in only one direction at any time. In this paper, we are primarily concerned with ensuring that the system recovers rapidly after the overload is over, either (i) because the two systems return to normal loading or (ii) because the direction of the overload suddenly shifts in the opposite direction. To achieve rapid recovery, we introduce lower thresholds for the queue ratios, below which one-way sharing is released. As a basis for studying the complex dynamics, we develop a new six-dimensional fluid approximation for a system with time-varying arrival rates, extending a previous fluid approximation involving a stochastic averaging principle. We conduct simulations to confirm that the new algorithm is effective for predicting the system performance and choosing effective control parameters. The simulation and the algorithm both show that the system can experience an inefficient nearly-periodic behavior, corresponding to an oscillating equilibrium (congestion collapse), if the sharing is strongly inefficient and the control parameters are set inappropriately.

1. Introduction.

1.1. *An Automatic Overload Control.* In this paper we study an automatic control to temporarily activate "emergency" measures in an uncertain dynamic environment to mitigate damage from an unexpected disruption, and then automatically return to normal operation when the disruption is over. There are two important questions: First, how and when should the control be activated? And, second, how and when should the control be released?

MSC 2010 subject classifications: Primary 60K25, ; secondary 60K30; 90B22

Keywords and phrases: service systems, overload control, congestion collapse, time-varying queues, many-server queues, recovery after overload incidents, fluid models

These issues arise in many contexts and have long been studied within the discipline of control theory [25, 45]. A familiar automatic control is a thermostat, which automatically turns on and off a heater and/or an air conditioner within a building. Since building temperature tends to change slowly relative to human temperature tolerance, conventional thermostats operate well with little concern, but special thermostats are needed for complex environments, such as in biochemical processes [3].

Another example of an automated control occurs in a large stock market exchange, such as the New York Stock Exchange (NYSE). To respond to the experience of dramatic fluctuations in prices, in 1988 the NYSE instituted trading curbs called *circuit breakers* or collars, which stop trading for a specified period in the event of exceptionally large price changes. With the increase of high-speed computer trading, these controls have become even more important and interesting since then [16].

The specific setting considered here involves two large-scale telephone call centers (or service pools within the same call center) that are designed to operate independently, but have the capability (due to both network technology and agent training) to respond to calls from the other system, even though there might be some loss in service effectiveness and efficiency in doing so. These call centers are designed and managed to separately respond to uncertain fluctuating demand and, with good practices, usually can do so effectively; see [1] for background. However, these call centers may occasionally face exceptional unexpected overloads, due to sudden surges in arrivals, extensive agent absenteeism or system malfunction (e.g., due to computer failures). It thus might be mutually beneficial for the two systems to agree to help each other during such overload incidents. We propose an automatic control for doing so. We are motivated by this call-center application, but the insights and methods should be useful in other service systems. Since we model the call centers as multi-server queues, the insights and methods may also be useful for other queueing settings.

In telecommunication systems and the Internet, the standard overload controls reduce the demand through some form of admission control (rejecting some arrivals) or otherwise restricting demand; see [4, 14, 32, 42, 49] and references therein. These controls, that reject or reduce arrivals, are especially important when the increasing load can cause the useful throughput (the "goodput") not only to reach its largest possible value, but also to actually decrease. Such anomalous behavior can occur because some of the customers "go bad." The classic telephone example is failure during the call setup process. The customer might start entering digits before receiving dial tone or abandon before the call is sent to the destination. As a consequence,

the vast majority of system resources may be working on requests that are no longer active, causing the throughput to actually decrease. In response, various effective controls have been developed [10, 27].

In contrast, here we assume that *no* arrivals will be directly turned away, although on their own initiative customers may elect to abandon from queue because they become impatient. Instead, we develop a control that automatically sends some of the arrivals to receive service from the other service pool when appropriate conditions are met. It is natural to prefer diverting instead of rejecting arrivals whenever some response is judged to be better than none at all, even if delayed. Indeed, diverting instead of rejecting arrivals is the accepted policy with ambulance diversion in response to overload in hospital emergency rooms, e.g., see [5, 9, 53] and references therein. The results here may be useful in that context as well, but then it is necessary to consider the extra delay for ambulances to reach alternative hospitals, which has no counterpart in networked call centers. (We assume that the calls can be transferred instantaneously.)

1.2. *Congestion Collapse.* An important feature of this kind of sharing, which is captured by our our model, is that the sharing may be inefficient. A simple symmetric example that we will consider in §4 has identical service rates for agents serving their own customers, but identical slower service rates when serving the other customers. With such inefficiency, the whole system will necessarily operate inefficiently, with lower throughput of both classes, if both pools are busy serving the other customers instead of their own. Nevertheless, we find that judicious sharing with our proposed overload control can be effective even with some degree of inefficiency, but care is needed in setting the control parameters. A major concern with such inefficient sharing is that the system may possibly experience *congestion collapse*, i.e., the system may reach an equilibrium with inefficient operation [43].

It is known that control schemes can cause congestion collapse; see, e.g., [11]. Within telecommunications there is a long history of congestion collapse and its prevention in the circuit-switched telephone network. More than 60 years ago, it was discovered that the capacity and performance of the network could greatly be expanded by allowing alternative routing paths [52]. If a circuit is not available on the most direct path, then the switch can search for free circuits on alternative paths. The difficulty is that these alternative paths may use more links and thus more circuits. Thus, in overload situations (the classic example being Mother's Day), the network can reach a stable inefficient operating regime, with the system congested, but far less than maximal throughput. This congestion collapse in the telephone

network was first studied by simulation [48]. The classical remedy in such loss networks is trunk reservation control, where the last few circuits on a link are reserved for direct traffic; see [12], §§4.3-4.5 of [24] and references therein.

Overload controls have also been considered for more general multi-class loss networks. In the multi-class setting, it may be desirable to provide different grades of service to different classes, including protection against overloads caused by overloads of other classes. Partial sharing controls achieving these more general goals can be achieved exploiting upper limit bounds and guaranteed minimum bounds [6]. Moreover, in [6] algorithms are developed to compute the performance associated with such complex controls, which greatly facilitates choosing appropriate control parameters. For the (different) problem we consider, we also develop a performance algorithm that can be used to set the control parameters.

Even though a call center can be regarded as a telecommunications network, our problem is quite different from the classical loss network setting discussed above. By definition, the loss network has *no* queues, so that all arrivals that cannot immediately enter service are turned away. In sharp contrast, our system turns *no* arrivals away. As a consequence, our system is more "sluggish;" it responds more slowly to changes in conditions, and presents new challenges.

For the model considered here, we show in §4 that the two call centers can indeed experience behavior that is best described as congestion collapse if the sharing is strongly inefficient and an inappropriate control is used. An unstable oscillating equilibrium is predicted by our numerical algorithm for the approximating fluid model and confirmed by simulation; see Figures 6 and 7 for the simulation and Figures 25 and 26 for the algorithm.

However, this oscillatory phenomenon is far from obvious because the stochastic model after the overload is over is an ergodic time-homogeneous CTMC with a steady-state limiting distribution. The situation that we consider in this paper is similar to the nearly periodic behavior of the $G/D/s+GI$ queue exposed in [28]. In that setting, the actual stochastic system has a well-defined limiting steady-state distribution and yet the system exhibits nearly periodic behavior over long time periods. When the scale is large, it turns out that the nearly periodic transient behavior observed in simulations is well predicted by a limiting fluid model. Unlike the stochastic model, the fluid model does not have a unique limiting steady-state. The reason for this discrepancy is that the two iterated limits (as time gets large and as the scale, determined by the arrival rate, gets large) done in different order are not equal.

In this paper we show the existence of the nearly periodic behavior (with inefficient sharing and inappropriately chosen controls), tantamount to congestion collapse, with our fluid algorithm and simulation. We provide additional mathematical support in [40] by proving that unstable oscillating equilibria can exist for a class of these fluid models.

However, this highly undesirably behavior can be avoided with reasonably chosen controls. In this paper we develop a model and an algorithm for analyzing that model that can be used to achieve the benefits of sharing while avoiding such bad behavior.

1.3. *Fixed-Queue-Ratio Controls.* Our overload control is a modification of the *Fixed-Queue-Ratio* (FQR) and more general Queue-and-Idleness-Ratio (QIR) controls proposed for routing and scheduling in a multi-class multi-pool call center under normal operating conditions in [18, 19, 20]. For the two-class two-pool X model considered here, the FQR rule sends customers to the other service pool if the ratio of the queue lengths exceeds a specified ratio. However, the theorems establishing that the FQR control is effective in [18, 19, 20] have conditions that *do not* hold for our networks here, which has a cyclic routing graph and service rates that depend on the customer class and service pool. Indeed, Example 2 of [35] shows that the X model can experience severe congestion collapse under normal loading if FQR is used. (The congestion collapse shown in [35] is different than the one mentioned above, which is due to the undesired oscillatory behavior.)

Nevertheless, in [35] we showed that the FQR control can usefully be applied as an overload control for the X model with inefficient sharing if we introduce additional activation thresholds. The *FQR control with thresholds* (FQR-T) sends customers to the other service pool if the queue ratio exceeds the activation threshold. For the X model, the FQR-T control has four parameters: a target ratio and an activation threshold for each direction of sharing. The target ratios are chosen to minimize the long-run average cost during the overload incident in an approximating stationary deterministic fluid model with a convex cost function applied to the two queues. To prevent harmful sharing, we also imposed the condition of *one-way sharing*; i.e., sharing is allowed in only one direction at any one time.

To better understand the transient behavior of the FQR-T control, in [36] we developed a deterministic fluid model to analyze the performance. That model is challenging and interesting because it is an *ordinary differential equation* (ODE) involving a stochastic *averaging principle* (AP). In [37, 38, 39] we established supporting mathematical results about the FQR-T control, including a functional weak law of large numbers (FWLLN) and

functional central limit theorem (FCLT) refinement. The previous analysis showed that the FQR-T control can rapidly respond to and mitigate an unexpected overload, while preventing sharing under normal conditions.

1.4. *New Contribution: Rapid Recovery After the Overload Is Over.* In this paper we show that FQR-T needs to be modified in order to ensure that the system recovers rapidly after an overload is over, either (i) because the two systems return to normal loading or (ii) because the direction of the overload suddenly shifts in the opposite direction. To achieve rapid recovery, we propose additional release thresholds for the shared-customers processes, below which one-way sharing is released. (We had previously recognized that such a modification of FQR-T was needed, e.g., see paragraph 3 in §2.2 of [37] and Remark B.1 in Appendix B of [38], but we now show for the first time that the modified control can be analyzed and can be effective.)

As a basis for studying such more complex dynamics, we extend our previous fluid model approximation in two ways: (i) the new fluid model is 6-dimensional instead of 3-dimensional and (ii) the model is allowed to have time-varying arrival rates and staffing functions. We also extend our previous algorithm to numerically compute the fluid solution to this more complex model. We implement the new algorithm and conduct simulations to show that the fluid model and the associated algorithm are effective in predicting system performance. Finally, we show that the new *FQR control with activation-and-release thresholds* (FQR-ART) can be effective with appropriate control parameters. We provide guidelines on choosing appropriate control parameters in the paper. The new model and algorithm can be used to confirm that a good choice has been made.

As before, for large scale in the model, there is important *state space collapse* (SSC) during overload periods with active sharing of customers. As a consequence, even though the basic stochastic process is 6-dimensional, the approximating fluid model is essentially 3-dimensional when there is active sharing instead of 6-dimensional. One of the complications in the new setting is to identify if and when SSC begins and in what direction (which queue is receiving help), and when it ends. The stochastic AP determines when SSC occurs, and how the fluid evolves during periods of SSC. When we introduce release thresholds, we also discover that to achieve good robust performance, we also need to increase the activation thresholds.

In summary, our contribution is fourfold: (i) We continue our study of the X model and demonstrate how and when it is beneficial (or harmful) to exploit system flexibility in response to an overload. (ii) We improve the previous FQR-T control designed to automatically exploit that system flex-

ibility when it is beneficial to do so by ensuring rapid recovery when the overload has ended. (iii) We develop a novel fluid model to approximate the intractable stochastic system in the time-varying environment and help determine appropriate control parameters. (iv) Finally, we design an efficient algorithm to solve that fluid model. (This paper addresses the control from an engineering perspective, as in [36]; we do not focus on underlying mathematics (prove theorems) as in [37, 38, 39].)

Simulation also plays an important role in our study. First, we use simulation to show that refinements to the FQR-T control are needed to ensure rapid recovery after the overload is over. Second, we use simulation to demonstrate that the fluid model provides a good performance approximation. Finally, we use simulation to verify that we can indeed gain important insights into complex system behavior from the fluid model, even for systems that are not overloaded, as in our examples after the overload has ended.

1.5. *Other Related Literature.*

Time-Varying Models.. A significant contribution here is extending the analysis of the transient behavior of a stationary fluid model to the analysis of (the necessarily transient behavior of) a time-varying model. When the predictable variability captured by time-varying model parameters dominates the unpredictable stochastic variability, deterministic fluid models are especially appropriate. Operationally, the deterministic fluid models tend to capture the essential performance. Mathematically, the deterministic fluid models are much easier to analyze than their stochastic extensions, such as diffusion approximations. The vast majority of the queueing literature concerns stationary models, but there have been important exceptions, e.g., [26, 33]. For related recent work, see [21, 22, 28, 29, 30, 31], and references therein.

Overloaded Systems and Fluid Models.. For other work that considers overloaded systems and fluid models, see [7, 17, 23, 46]. The authors in [7] suggest using the max-weight policy which, much like the FQR-ART control here, is easy to implement because it uses only information on the current state of the system; it stabilizes the system during normal loads and keeps the queues at target ratios when the system cannot be stabilized due to high arrival rates. In [17] overflow networks in heavy-traffic are studied in settings of co-sourcing, i.e., when firms that operate their own in-house call center overflow a *nonnegligible proportion* of the arrivals to a call center that is operated by an outsourcer. Fluid and diffusion limits are obtained via a stochastic averaging principle. The authors in [46] apply their previously introduced *shadow routing* control to overloaded parallel systems with un-

known arrival rates, and show that it maximizes the reward rate, assuming a class-dependent reward of each customer served.

Healthcare Systems.. System overloads are especially prevalent in healthcare systems, often even being the "natural state." Some facilities such as *intensive care units* (ICU's) and equipment such as magnetic resonance imaging (MRI) machines are so expensive that they are designed to be operated continuously, and exhibit long lines of waiting patients. Extreme overloads can occur with mass casualty events.

Hospitals have complex queueing dynamics, with multiple internal flows among its units in addition to exogenous arrival streams. Thus, overloads in some units of a hospital can "propagate" to other units, creating a system-wide overload. For example, when *inpatient wards* (IW) are overloaded, patients from the *emergency department* (ED) who need to be hospitalized cannot be transferred to the IW due to the unavailability of beds, creating the phenomenon of *blocked beds* in the ED, i.e., beds that are occupied by patients who finished their treatment in the ED; see, e.g., [5] for a current review. See [2] for a data-based study of queueing aspects in hospital settings as well as an extensive literature review.

In [8], a fluid approximation of an ICU experiencing periods of overload periods is studied, in which the service rate of current ICU patients increases (is "sped-up") if the number of patients that are waiting to be admitted to the ICU exceeds a certain threshold. In turn, the sped-up patients have an increased probability of readmission to the ICU, so that alleviating overloads by employing speedup increases future overloads. The fluid model in [8] exploits an averaging principle in the spirit of [36].

1.6. *Organization of the Rest of the Paper.* In §2 we define the stochastic X model and the FQR-T and FQR-ART controls. Building on simple fluid considerations, In §3 and §4 we demonstrate the need to modify FQR-T in order to rapidly recover after the overload is over. In §3 we show why release thresholds are needed. In §4 we show that, unless precaution is taken, the release thresholds can cause congestion collapse when the system recovers from an overload. To avoid that bad behavior, the activation thresholds need to be increased beyond the FQR-T values. In §5 we develop the fluid approximation and in §6 we develop an efficient algorithm to numerically solve it. In §7 we provide numerical examples, demonstrating the effectiveness of both the FQR-ART control and the fluid model by comparing the results of the numerical algorithm for the ODE to the results of simulation experiments. Finally, in §8 we draw conclusions and suggest directions for further research.

2. The Time-Varying X Model. As depicted in Figure 2, the X model has two customer classes and two agent pools, each with many homogeneous agents working in parallel. We assume that each customer class

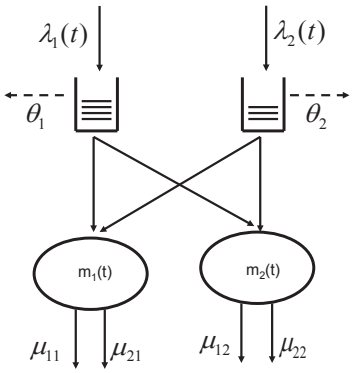

FIG 1. *The X model*

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, i.e., customers may be served at a slower rate when served in the other class pool. We assume that the service times are independent exponential random variables, with $1/\mu_{i,j}$ being the expected time for a class i customer to be served in service pool j. Each class has a buffer with unlimited capacity where customers who are not routed immediately into service upon arrival wait to be served. Within each class, customers enter service according to the first-come-first-served discipline. Customers have limited patience, so that they may abandon from the queue. The successive patience times of class i customers are i.i.d. exponential variables with mean $1/\theta_i$.

We assume that customers arrive according to independent *nonhomogeneous* Poisson processes, one for each class, with time-varying deterministic rate functions. The staffing levels are assumed to be time dependent as well, usually chosen to respond to anticipated changes in the arrival rates; see [30] and references therein. As discussed in §1 of [29], it is necessary to specify how the system responds when the staffing level of a service pool is scheduled to decrease. As in [29], we allow server switching (an agent can take over service from an agent scheduled to leave). Since service times are exponential, it thus suffices to let idle agents leave when staffing decreases, and the first agent to become idle leave when all agents are busy when staffing is scheduled to decrease.

Even though we do not prove any limit theorems as we did in [38, 39], and instead develop direct fluid models to approximate the stochastic system, we will use asymptotic considerations in our analysis. We therefore consider a sequence of X systems, as just described, indexed by a superscript n. As is standard for many-server heavy-traffic limits [38], the service rates and abandonment rates are independent of n, but the arrival rates and staffing levels increase. Specifically, for each $n \geq 1$, let $\lambda_i^n(t)$ be the arrival rate to pool i and let $m_j^n(t)$ be the number of agents in pool j at time t. For the fluid approximation, we assume that

$$\lambda_i^n(t)/n \to \lambda_i(t) \quad \text{and} \quad m_j^n(t)/n \to m_j(t) \quad \text{as} \quad n \to \infty, \tag{1}$$

uniformly in t over each bounded time interval.

As in [29], we assume that the limit functions λ_i and m_j in (1) are piecewise-smooth, by which we mean that they have only finitely many discontinuities in any finite interval, have limits from the left and right at each discontinuity point and are differentiable at all continuity points. That assumption is not restrictive for applications and supports analysis of the approximating fluid model by differential equations. For call-center applications, it usually suffices to consider piecewise-constant functions, but we allow greater generality because our methods can be applied in other settings.

Let $Q_i^n(t)$ be the number of customers waiting in the class-i buffer and $Z_{i,j}^n(t)$ be the number of class-i customers in service pool j at time t in system n. Let the associated six-dimensional vector process be

$$X^n \equiv X^n(t) \equiv (Q_i^n(t), Z_{i,j}^n(t) : i, j = 1, 2), \quad t \geq 0. \tag{2}$$

We consider controls that are functions of $X^n(t)$ at each t, making X^n a nonhomogeneous CTMC.

To define asymptotic regimes, let $\rho_i^n(t) := \lambda_i^n(t)/(\mu_{i,i} m_i^n(t))$ be the instantaneous traffic-intensity function of class i (and pool i) alone in system n at time t. By (1),

$$\rho_i^n(t) - 1 \to \beta_i(t) \quad \text{as} \quad n \to \infty, \tag{3}$$

uniformly in t over each bounded time interval. We say that class i (and pool i) is *underloaded* at time t if $\beta_i(t) < 0$, *overloaded* at time t if $\beta(t) > 0$ and *normally loaded* at time t if $\beta_i(t) = 0$.

The generality we have introduced allows for many possible scenarios, but here we restrict attention to an unexpected overload incident followed by a subsequent instantaneous switch in state, either (i) a return to normal

loading or (ii) a switch in the direction of overloading. Thus, now there are three intervals: first normally loaded, then overloaded and then a final new regime, which is either normal loading for both classes or an overload in the opposite direction. During each of these three intervals, the arrival rates and staffing functions are allowed to change.

As before, we consider the system starting at the unanticipated time when the first overload incident begins. However, now the arrival rates and staffing functions no longer need to be constant within each interval. By assumption, they have discontinuities at the beginning of the first overload incident and at the subsequent time when the overload is over. For the generality that we do consider, we exploit the fact that we know how to staff to stabilize the system in face of time-varying arrival rates under normal loading; see [29, 30] and references therein.

2.1. *The Initial FQR-T Control.* For each $n \geq 1$, the FQR-T control is based on two positive (activation) thresholds, $k^n_{1,2}$ and $k^n_{2,1}$ and the two queue-ratio parameters, $r_{1,2}$ and $r_{2,1}$ (which are chosen independent of n under (1)). We define two (centered) queue-difference stochastic processes

$$\begin{aligned} D^n_{1,2}(t) &\equiv Q^n_1(t) - k^n_{1,2} - r_{1,2}Q^n_2(t) \quad \text{and} \\ D^n_{2,1}(t) &\equiv r_{2,1}Q^n_2(t) - k^n_{2,1} - Q^n_1(t), \quad t \geq 0. \end{aligned} \tag{4}$$

As long as $D^n_{1,2}(t) < 0$ and $D^n_{2,1}(t) < 0$ we consider the system to be not overloaded so that no customers are routed to be served in the other class pool. Once one of these inequalities is violated, the system is considered to be overloaded, and sharing is initiated. For example, if $D^n_{1,2}(t) \geq 0$, then class 1 is judged to be overloaded (because then $Q^n_1 - r_{1,2}Q^n_2 \geq k^n_{1,2}$), and it is desirable to send class-1 customers to be served in pool 2. Note that $D^n_{1,2}(t) \geq 0$ does not exclude the case that class 2 is also overloaded; we can have $\beta_i(t) > 0$ for *both* i. However, once one of the thresholds is crossed, its corresponding class is considered to be "more overloaded" than the other class. (We refer to this situation as *unbalanced overloads*.) We call $k^n_{1,2}$ and $k^n_{2,1}$ *activation thresholds*, because exceeding one of these thresholds activates sharing (and not exceeding prevents sharing when it is not desired).

The behavior of X^n in (2) depends on the choice of the thresholds $k^n_{i,j}$. In particular, we want the thresholds to be large enough so that sharing will not take place if both service pools are normally loaded, and to be small enough to detect any overload quickly, and start sharing in the correct direction once the overload begins. Note that without sharing, the two pools operate as two independent $M_t/M/m^n_t + M$ (time-varying Erlang-A) models. The familiar fluid and diffusion limits for the stationary Erlang-A model give insight as

to how to choose these thresholds; e.g., see [15, 34]. In Assumption 2.4 of [38] and Assumption 3 of [39] we assumed that the activation thresholds are chosen to satisfy:

(5)
$$k^n_{i,j}/n \to 0 \quad \text{and} \quad k^n_{i,j}/\sqrt{n} \to \infty \quad \text{as} \quad n \to \infty, \quad i,j = 1,2 \quad \text{with } i \neq j.$$

The first limit in (5) ensures that overloads are detected quickly (immediately in the fluid model obtained as $n \to \infty$), whereas the second limit in (5) ensures that stochastic fluctuations of normally-loaded pools will not cause undesired sharing, since the diffusion-scaled queue in that case are of order $\sqrt{n}$.

Given that the system is designed so that sharing of customers takes place only during overloads, it is reasonable to assume that agents serve the other class customers (the so-called "shared customers") at a slower rate than they serve their own designated customers. Thus, substantial sharing is likely to reduce the effective service rate of the helping pool. In our previous work we took measures to avoid sharing in both directions simultaneously. In particular, we imposed the one-way sharing rule described in §1. However, it is evident that the one-way sharing rule may considerably slow the recovery after the overload is over. We elaborate in §3 below.

To remedy this problem, we could consider removing the one-way sharing rule altogether and rely solely on the activation thresholds to avoid undesired sharing. However, removing the one-way sharing rule makes it necessary to increase the activation thresholds substantially, increasing the time until overloads are detected. Moreover, if these thresholds are too large, then some overloads may not be detected at all, because abandonment keeps the queues from increasing indefinitely. (While there is also a need to increase the activation thresholds in our setting here, that increase is less than would be required if the one-way sharing was completely removed.) Moreover, if sharing is taking place in one direction and then immediately starts in the other direction in response to a switch in the overload, then the combined service capacity of both pools may be reduced significantly, creating a period of severe congestion in both directions. Hence, it is beneficial to avoid too much simultaneous two-way sharing. We again refer to Example 2 in [35]. Therefore, our new control relaxes the one-way sharing rule by introducing the release thresholds alluded to above. We elaborate in the following subsection.

2.2. *The Proposed FQR-ART Control.* For the reasons discussed above, we suggest a modification of the one-way sharing rule by introducing *release thresholds* (RT). For each $n \geq 1$, we introduce two strictly positive numbers

$\tau^n_{1,2}$ and $\tau^n_{2,1}$. A newly available type-2 agent is allowed to take a class-1 customer at time t only if $Z^n_{2,1}(t) \le \tau^n_{2,1}$, i.e., if the number of type-1 agents serving class-2 customers at the same time t is below $\tau^n_{2,1}$ (and of course $D^n_{1,2}(t) \ge 0$), and similarly in the other direction. (Ways to choose the parameters $\tau^n_{1,2}$ and $\tau^n_{2,1}$ will be discussed later.)

However, the new release thresholds allow small simultaneous sharing in both directions, which can slightly increase the overload. In some cases, this slight increase in the overload is sufficient to cause the system to spin out of control and start to oscillate, as we demonstrate in §4 below. In particular, the new release thresholds make activation thresholds satisfying (5) unsuitable. We therefore conclude that these activation thresholds should be positive in "fluid scale", i.e., they should be chosen so as to satisfy

$$\lim_{n\to\infty} k^n_{i,j}/n = k_{i,j} > 0, \quad i,j = 1,2. \tag{6}$$

Thus, the FQR-ART control is specified by the parameter six-tuple

$$(r_{1,2}, r_{2,1}, k^n_{1,2}, k^n_{2,1}, \tau^n_{1,2}, \tau^n_{2,1})$$

and the routing and scheduling rules which depend on the values of the two processes $D^n_{i,j}$ and $Z^n_{i,j}$, $i \ne j$, in the manner described above. Note that FQR-T requires knowing only the queue lengths $Q^n_i(t)$ at each time t (specifically, the values of the two difference processes (4)), whereas FQR-ART also requires knowledge of $Z^n_{1,2}$ and $Z^n_{2,1}$. Under either control, the X model is a (possible inhomogeneous) CTMC.

2.3. *Analysis Via Fluid Approximations.* Since the stochastic process X^n in (2) under FQR-ART is evidently too difficult to analyze exactly, we will employ a deterministic dynamical-system approximation, and refer to that approximation as "fluid approximation" or "fluid model" interchangeably. The main idea in using fluid approximations is that, for large n, $\bar{X}^n \approx x$, for some deterministic function x that is easier to analyze than the untractable stochastic process X^n. (We use the 'bar' notation throughout to denote fluid scaled processes, e.g., $\bar{X}^n \equiv X^n/n$.) In particular, the fluid counterpart of X^n in (2) is the six-dimensional deterministic function

$$x \equiv x(t) \equiv (q_i(t), z_{i,j}(t) : i,j = 1,2), \quad t \ge 0, \tag{7}$$

where q_i and $z_{i,j}$ are the fluid approximations for the stochastic processes Q^n_i and $Z^n_{i,j}$, $i,j = 1,2$. The approximation $\bar{X}^n \approx x$ should be supported by a *functional law of large numbers* (FLLN), stating that $\bar{X}^n \Rightarrow x$ as $n \to \infty$,

extending [38], but that remains to be established. (However, the FWLLN has been established for the FQR-T model in [**?**].)

In the stochastic system, customer routing depends on the values of the difference processes in (4). For example, if sharing is taking place with pool 2 helping class 1, and assuming $Z^n_{2,1} \le \tau^n_{2,1}$, the process $D^n_{1,2}$ determines which customer class a newly available type-2 agent will take. as in [36, 37, 38], that implies that the resulting fluid model is much more complicated than most fluid models in the literature. In particular, in the fluid system we cannot simply replace the process $D^n_{1,2}$ with a process

$$d_{1,2}(t) \equiv q_1(t) - k_{1,2} - r_{1,2}q_2(t), \quad t \ge 0.$$

In fact, the purpose of the control is to keep $d_{1,2}(t) = 0$ during the overload. Hence, as in [36, 37, 38], a refined asymptotic analysis of the behavior of $D^n_{1,2}$ (or $D^n_{2,1}$ during overloads in the other direction) is required. That refined analysis can be carried out thanks to a stochastic averaging principle, which replaces the processes $D^n_{i,j}$, $i, j = 1, 2$, with the long-run average behavior of corresponding limiting stochastic processes. In turn, those deterministic long-run averages determine the evolution of the fluid model; see §5 below, where the fluid equations are developed.

3. The Need to Relax the One-Way Sharing Rule. Relying on the fluid approximation, we now demonstrate why the one-way sharing rule impedes recovery after the overload incident is over. The simple fluid analysis suggests that release thresholds provide a good remedy, and helps indicate how they should be chosen.

3.1. *The Recovery Time With One-Way Sharing.* We consider two consecutive time intervals $I_1 = [t_0, t_1)$ and $I_2 = [t_1, t_2)$ with $0 \le t_0 < t_1 < t_2 \le \infty$, with the system being overloaded in opposite direction over each interval. Suppose that class 2 is overloaded over the time interval I_1 and that sharing is taking place with pool 1 helping class 2. Then, at time t_1 the loads suddenly change in such a way that sharing is required in the other direction. In particular, we assume that $\beta_1(t) \le 0$ and $\beta_2(t) > 0$ for $t \in I_1$, whereas $\beta_1(t) > 0$ and $\beta_2(t) \le 0$ for $t \in I_2$. We also assume that $z_{2,1}(t_1) > 0$.

We do two different mathematical analyses. We first consider a direct fluid model analysis, and then afterwards we consider the stochastic system. A fluid approximation for the evolution of $Z^n_{1,2}$ (which we refer to as $z_{1,2}(t)$) can easily be derived using rate considerations. Since every type-1 agent who is helping a class-2 customer at time $t > t_1$ will finish service immediately after time t at a rate $\mu_{2,1}$, regardless of the value of t, due to the memoryless

property, and since there are no more class 2 customers routed to pool 1 after time t_1, we expect that $z_{2,1}$ will satisfy the ODE

$$\dot{z}_{2,1}(t) = -\mu_{2,1} z_{2,1}(t), \quad t \in I_2,$$

whose unique solution is

$$z_{2,1}(t) = z_{2,1}(t_1) e^{-\mu_{2,1} t}, \quad t \in [t_1, t_2). \tag{8}$$

As a consequence, for the fluid model, if $z_{2,1}(t_1) > 0$, then pool 1 will *never* empty, so that sharing can *never* begin in the opposite direction.

We now characterize the random time T^n after the time t_1 in the stochastic system with scale n for $Z^n_{2,1}(t)$ to first hit 0. The time required for all these customers to complete service is the maximum of $Z^n_{2,1}(t_1)$ i.i.d. exponential random variables. It is well known that the maximum of n i.i.d. exponential random variables with mean 1 is the harmonic sum $H_n \equiv \sum_{j=1}^{n}(1/j)$. Moreover, it is well known that $H_n - \log_e n \to \gamma$ as $n \to \infty$, where $\gamma \equiv 0.57721\ldots$ is the Euler-Mascheroni constant. This limit is relevant for us, because from the established FWLLN in [38], we know that having $z_{2,1}(t_1) > 0$ implies that $Z^n_{2,1}(t_1) \approx z_{2,1}(t_1)n$.

Hence, given $Z^n_{2,1}(t_1)$ and its approximate value, for large n,

$$E[T^n] = \sum_{j=1}^{Z^n_{2,1}(t_1)} \frac{1}{j \cdot \mu_{2,1}} \approx \frac{\log_e\left(Z^n_{2,1}(t_1)\right)}{\mu_{2,1}} \approx \frac{\log_e\left(n z_{2,1}(t_1)\right)}{\mu_{2,1}}. \tag{9}$$

We thus see that the expected time required for a pool to empty its shared customers after an overload is over, and no new shared customers are routed to that pool, is of order $\log_e(n)$ as $n \to \infty$.

3.2. *Choosing Appropriate Release Thresholds.* The simple considerations leading to (8) and (9) show that a large system will be slow to recover after an overload is over. That analysis also helps choose appropriate release thresholds. Indeed, the fluid model easily generates an approximate recovery time. In particular, if a release threshold of $\tau_{2,1}$ is used in the fluid model starting with $z_{2,1}(t_1)$ at time t_1, where $z_{2,1}(t_1) > \tau_{2,1} > 0$, then the release threshold will be hit at time

$$T \equiv \frac{1}{\mu_{2,1}} \log_e\left(\frac{z_{2,1}(t_1)}{\tau_{2,1}}\right).$$

The analysis above indicates that the release thresholds in stochastic system n should be of order $O(n)$ as n increases. It suffices to pick two strictly positive numbers $\tau_{1,2}$ and $\tau_{2,1}$ and let

$$\tau^n_{1,2} \equiv n\tau_{1,2} \quad \text{and} \quad \tau^n_{2,1} \equiv n\tau_{2,1}. \tag{10}$$

With the scaling in (10), the recovery time T^n in system n should be approximately a constant, independent of n.

In summary, with FQR-ART, an available type-2 agent is allowed to serve a class-1 customer only if $Z^n_{2,1}(t) \le \tau^n_{2,1}$ (or, equivalently, only if $\bar{Z}^n_{2,1}(t) \le \tau_{2,1}$), and of course $D^n_{1,2}(t) \ge 0$, and similarly in the other direction. The choice in (10) shows that the release thresholds should be proportional to n, but does not determine the proportionality constants $\tau_{1,2}$ and $\tau_{1,2}$. Further analysis shows that these can be quite small, as we show next.

3.3. *Simulation Experiments.* To illustrate the importance of the release thresholds for stochastic systems, we conducted simulation experiments, comparing the performance of a system with and without release thresholds. The results can be seen in Figures 2 and 3.

The (fixed) parameters for this simulation are

$$\begin{aligned} m^n_1 &= m^n_2 = 1000, \quad \lambda^n_1 = 1200, \quad \lambda^n_2 = 990, \quad \mu_{1,1} = \mu_{2,2} = 1, \\ \mu_{1,2} &= \mu_{2,1} = 0.5, \quad \kappa^n_{1,2} = \kappa^n_{2,1} = 100, \quad \text{and} \quad r_{1,2} = r_{2,1} = 1. \end{aligned}$$

(Here, we can think of n as being fixed and equal to 1000.) With these parameters, $\rho^n_1 = 1.2$ and $\rho^n_2 = 0.99$, where $\rho^n_i \equiv \lambda^n_i/(m^n_i\mu_{i,i})$, so that class 1 may be regarded as overloaded, whereas class 2 may be regarded as normally loaded (recall (3)).

To respond to that unbalanced overload by having pool 2 help class 1, we should have $Z^n_{1,2} > 0$ and $Z^n_{2,1} = 0$ if one-way sharing is employed. However, we initialize the system at time 0 sharing in the opposite direction, with *all* pool 1 agents serving class 2 customers. We are interested in the time it takes the stochastic process $Z^n_{2,1}$ to reach 0, so that the desired sharing can begin. Without release thresholds, the required recovery time is quite long, approximately 21 (mean service times, of their own type). In contrast, with release thresholds of only $\tau^n_{1,2} = \tau^n_{2,1} = 0.01n = 10$, that time is reduced from about 21 to about 9 service times. Thus, clearing the last 1% of the class-2 customers in pool 1 without release thresholds takes more than half the total clearing time!

We hasten to admit that we just considered an extreme example in which *all* of service pool 1 is initially busy with customers from class 2. We did so in order to convey the message that *it is the last few agents working with class* 1 *that cause the largest part of the delayed response*. In particular, the $Z^n_{2,1}$ process decreases fast at the beginning, but then the decrease rate slows down considerably.

From Figures 2 and 3, it is also easy to see what happens in less extreme cases, when $0 < Z_{2,1}(0) < m_1$. For example, if we initialize with 20% sharing

in the wrong direction, we see that, without a release threshold, the time to activate sharing in the right direction is about $21 - 4 = 17$ time units. In contrast, with release thresholds, it is about $9 - 4 = 5$ time units. (Figures 2 and 3 show that the common value 4 in these calculations is the time to go from 100% sharing in the wrong direction to only 20% sharing in the wrong direction, which would be the same in the two cases.) 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 time units).

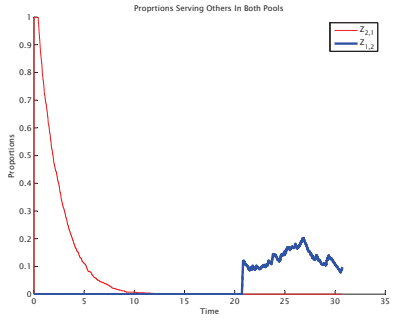

FIG 2. *Sample paths of* $\bar{Z}^n_{1,2}(t)$ *and* $\bar{Z}^n_{2,1}(t)$ *initialized incorrectly, without release thresholds.*

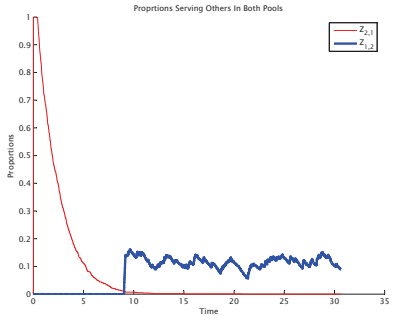

FIG 3. *Sample paths of* $\bar{Z}^n_{1,2}(t)$ *and* $\bar{Z}^n_{2,1}(t)$ *initialized incorrectly, with release thresholds* $\tau_{1,2} = \tau_{2,1} = 0.01$.

4. Congestion Collapse Due to Oscillations. The previous section dramatically showed the need for the release thresholds when the direction of the overload suddenly shifts. However, a more common case is for the two systems to simply return to normal loading, after which no sharing in either direction is desired. We now show that the release thresholds can cause serious problems when the system returns to normal loading after an overload incident if the activation thresholds are too small. In this case, there is a potential difficulty when the inefficient sharing condition holds, i.e., when $\mu_{1,1} > \mu_{2,1}$ and $\mu_{2,2} > \mu_{1,2}$, which is what we now assume. We show that, with inefficient sharing, the release thresholds combined with small activation thresholds can lead to oscillatory poor performance. We emphasize that, even though the performance is oscillatory, the model after the overload is over is a (necessarily aperiodic) positive-recurrent and stationary time-homogeneous CTMC when there is abandonment (as discussed in §1.2). In particular, our examples below are time-homogeneous CTMCs because the arrival rates and staffing levels are kept fixed.

4.1. *Simulations of Oscillating Systems with Inefficient Sharing.* The oscillatory behavior is more evident when there is no abandonment, so we start by considering a system without abandonment. We start with an extreme case having very inefficient sharing; i.e., we let $\mu_{1,1} = \mu_{2,2} = 1$, but $\mu_{1,2} = \mu_{2,1} = 0.1$. Afterwards we consider a more realistic example with customer abandonment and less efficiency loss from sharing. We consider a relatively heavily loaded symmetric system. In particular, let there be $m_1 = m_2 = 100$ agents in each pool and let the arrival rates be $\lambda_1 = \lambda_2 = 98$. Thus each class alone is stable, but if all the agents are busy in one pool with n serving the other class, then the total service rate out is $0.1k + (100 - k) = 100 - 0.9k$. When $k \geq 3$, the maximum service rate is less than the arrival rate 98, so that the rate in exceeds the maximum rate out at that instant.

We now illustrate bad behavior for poorly chosen thresholds. We make both the activation thresho.lds and the release thresholds be too small. In particular, we use ratio parameters $r_{1,2} = r_{2,1} = 1$, activation thresholds $k^n_{i,j} = 10$ and release thresholds $\tau^n_{i,j} = 1$ for $i, j = 1, 2$ and $i \neq j$. We start the system with both pools busy serving their own class, but no queues, i.e., $Z^n_{1,1}(0) = Z^n_{2,2}(0) = 100$ and $Q^n_1(0) = Q^n_2(0) = 0$. The symmetry implies that both pools and queues exhibit symmetric behavior.

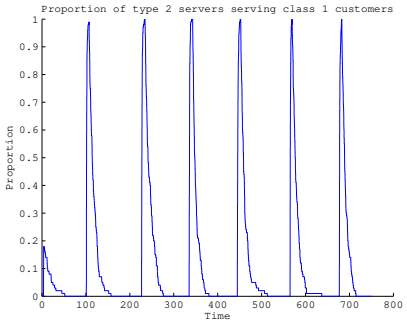

FIG 4. *Oscillations of $\bar{Z}^n_{1,2}$ in the extreme symmetric example with $\tau^n_{i,j} = 1$, $k^n_{i,j} = 10$ and no abandonment.*

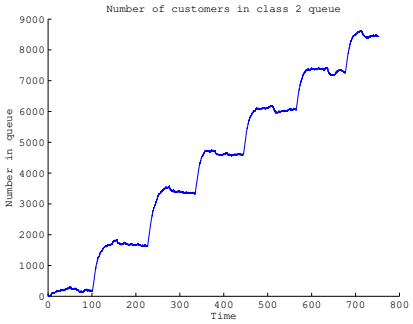

FIG 5. *Oscillating growth of $\bar{Q}^n_2$ in the extreme symmetric example with $\tau^n_{i,j} = 1$, $k^n_{i,j} = 10$ and no abandonment.*

Figures 4 and 5 show a single simulated sample path. Figure 4 shows that the proportion of agents serving customers from the other pool oscillates between 0 and 1, alternating between these two extremes over this horizon. The oscillatory behavior is occurring despite the fact that there is no sharing initially. Figure 5 shows that the queue lengths are growing in an oscillating

manner over the time interval $[0, 800]$ at an average rate of 10.

The oscillatory behavior also occurs for systems with abandonment, but it is often hard to detect, because the abandonment ensures that the stationary stochastic system after the overload has ended is stable and it dampens any oscillatory behavior. Nevertheless, the difficulty highlighted above remains with abandonment.

To demonstrate dramatically, we simulated the same system considered in the previous example, but now with the low positive abandonment rates $\theta_1 = \theta_2 = 0.01$. Figures 6 and 7 show that the oscillatory behavior remains. Moreover, Figure 7 suggests that Q_2^n (and, by symmetry, also Q_1^n) stabilizes at an overloaded oscillatory equilibrium. The oscillatory behavior in Figures 6–7 may be surprising at first, because the underlying (time-homogeneous) CTMC after the overload has ended is ergodic, as we mentioned above. Fortunately, the fluid model provides valuable insight, as we explain in §4.2.

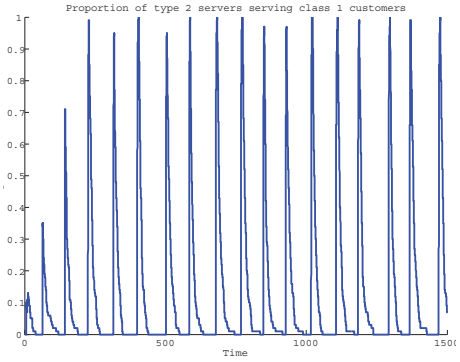

FIG 6. *Oscillations of $\bar{Z}_{1,2}^n$ in the extreme symmetric example with $\tau_{i,j}^n = 1$, $k_{i,j}^n = 10$ with abandonment.*

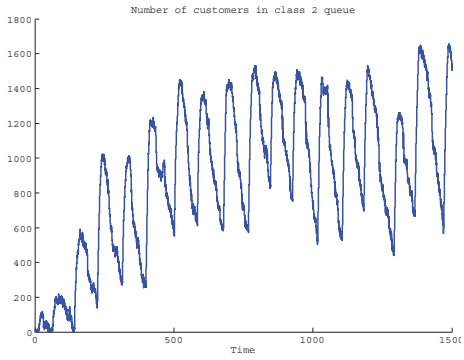

FIG 7. *Oscillating stable behavior of $\bar{Q}_2^n$ in the extreme symmetric example with $\tau_{i,j}^n = 1$, $k_{i,j}^n = 10$ with abandonment.*

We now consider a less-extreme more realistic example, in which the sharing service rates and abandonment rates are changed to $\mu_{1,2} = \mu_{2,1} = \theta_1 = \theta_2 = 0.5$. First, Figure 8 shows the proportion of shared customers over time with the previously specified activation thresholds of $k_{i,j}^n = 10$, but we now consider a system that is recovering from an overload in which pool 1 was helping class 2 customers. In particular, there are initially 20 type-1 agents helping class-2 customers. By taking this initial condition, we are considering a system that starts "worse off" than before, because it is initially overloaded. (In the other two examples, the systems were initialized empty.) We consider the time interval $[0, 100]$ to make the figures clear, but the be-

havior shown in the figures below remained for the whole duration of the simulation (which lasted for 1500 time units).

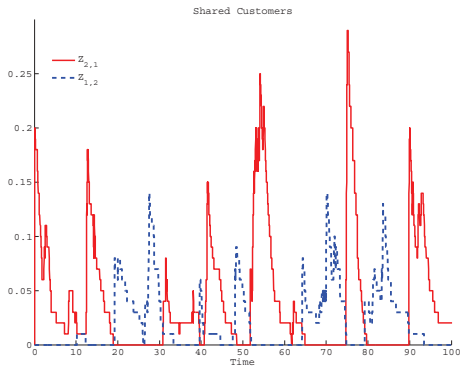

FIG 8. *Oscillations of* $\bar{Z}^n_{1,2}$ *in the more realistic symmetric example with abandonment:* $\mu_{1,1} = 1$, $\mu_{1,2} = 0.5$, $\theta_1 = 0.5$, $\tau^n_{1,2} = 1$ *and* $k^n_{i,j} = 10$.

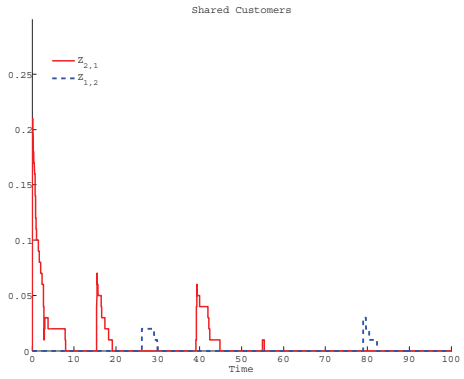

FIG 9. $\bar{Z}^n_{1,2}$ *in the more realistic example but with higher activation thresholds* $k^n_{i,j} = 35$.

In this case, substantial customer abandonment significantly dampens the sharing oscillations seen previously. Nevertheless, Figure 8 shows that the pools share repeatedly in an oscillating manner over the time interval $[0, 100]$. Although the long-run average number of agents that are helping the other class is not significant, this oscillatory behavior, is clearly undesirable. We do not show figures of the queues because they are uninformative (the oscillations are insignificant). Hence, the bad behavior in a system with a relative substantial customer abandonment may be hard to detect by only observing the queues, so that a system with no abandonment, or low abandonment rate, gives important insights.

To remedy the problem in Figure 8, we propose increasing the activation thresholds. To illustrate the potential benefit, Figure 9 shows the sharing when the activation thresholds are increased to $k^n_{i,j} = 35$, $i, j = 1, 2$, with all other parameters kept the same. Even though some customers are shared occasionally, especially just after the overload is over, the oscillatory behavior is minimal and decays quickly.

4.2. *Insight from the Deterministic Fluid Model.* In the examples we have just considered, the six-dimensional stochastic process X^n in (2) describing the system performance after the overload incident has ended is a stationary CTMC. With customer abandonment, that CTMC is necessarily stable, so that with FQR-ART and any parameter setting, the stochastic

process X^n in (2) necessarily has a unique steady-state distribution. Nevertheless, we have just seen that the system can exhibit quite complex undesirable behavior for some initial conditions if the control parameters are not set properly.

Fortunately, the fluid model we develop provides an effective means to study the complex system performance and set the control parameters. The oscillating behavior we see in the simulations looks periodic, but it is not quite; it is nearly periodic, just as in [28]. The system becomes more nearly periodic as the scale increases. In the many-server heavy-traffic limit, the stochastic process X^n approaches the deterministic solution of the fluid model we introduce next to serve as an approximation. From the algorithm for that fluid model, we see that it possesses a periodic equilibrium for some initial conditions.

As a consequence, the fluid model can be bistable; it can have a periodic equilibrium in addition to a stable equilibrium, depending on the initial conditions. Consequently, the order in which two different limits occur leads to different stories. As time increases, for any fixed scale, the stochastic process approaches its unique steady-state distribution. In contrast, as the scale increases, a properly-scaled version of the stochastic process approaches a deterministic function, which can be periodic. Thus, the fluid model provides important insight: an oscillatory fluid approximation implies that a corresponding large system experiences oscillatory behavior for prohibitively large time intervals, even though it is essentially a stationary CTMC. In [40] we prove that the fluid models can exhibit this bi-stability; Here it is verified numerically by applying the fluid algorithm.

5. The Fluid Model. The fluid model approximating the stochastic system X^n under FQR-ART is described as the solution to an ordinary differential equation (ODE), but that ODE depends on a stochastic averaging principle (AP). In this section we derive that ODE via a heuristic representation of the inhomogeneous CTMC in (2). The reasoning in the justification of the fluid model approximation parallels the heuristic engineering discussion in [36], to which we refer for more discussion. For mathematical support for that reasoning, see [37, 38].

5.1. *Representation of the Stochastic System During Overloads.* The sample paths of the queueing system can be represented in terms of its primitive processes, i.e., the arrival, abandonment and service processes, as a function of the control. Unlike traditional fluid models, in which the primitive stochastic processes are replaced by their long-run rates, the deterministic fluid model here is more involved and includes a stochastic ingredient in the

form of a stochastic AP, which we describe in detail in §5.2 below.

Even though we are not proving that the fluid model arises as a weak limit of the fluid-scaled stochastic system, we need to take asymptotic considerations in order to develop the fluid approximation. We thus start with a representation of the stochastic system during overloads, assuming that both service pools are full over an interval $[0,T]$, i.e.,

$$Z^n_{1,1}(t)+Z^n_{2,1}(t) = m^n_1(t) \quad \text{and} \quad Z^n_{2,2}(t)+Z^n_{1,2}(t) = m^n_2(t), \quad t \in [0,T]. \tag{11}$$

During the time interval $[0,T]$ no customers can enter service immediately upon arrival, and so all customers are delayed in queue. For simplicity, we first consider intervals over which the staffing functions are continuous and differentiable everywhere. In §7 we give an example of a staffing function with discontinuity; see Figure 22 below.

We represent the sample paths of X^n as random time-changes of independent unit-rate Poisson processes, as reviewed in [34]; see Equations (41)-(43) in [38] for such a representation applied to the X model operating under FQR-T. Let

$$\begin{aligned} \mathcal{A}^n_{1,2}(s) &\equiv \{\{D^n_{1,2}(s) > 0\} \cap \{Z^n_{2,1}(s) \le \tau^n_{2,1}\}\} \quad \text{and} \\ \mathcal{A}^n_{2,1}(s) &\equiv \{\{D^n_{2,1}(s) > 0\} \cap \{Z^n_{1,2}(s) \le \tau^n_{1,2}\}\}, \end{aligned} \tag{12}$$

the representation of Q^n_1 over $[0,T]$ is

$$\begin{aligned} Q^n_1(t) = N^a_1\left(\int_0^t \lambda^n_1(s)ds\right) - N^u_1\left(\theta_1\int_0^t Q^n_1(s)ds\right) \\ - N^+_1\left(\int_0^t \mathbf{1}_{\mathcal{A}^n_{1,2}(s)}\left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{1,2}Z^n_{1,2}(s) + \mu_{2,1}Z^n_{2,1}(s) + \mu_{2,2}Z^n_{2,2}(s)\right)ds\right) \\ - N^-_1\left(\int_0^t (1 - \mathbf{1}_{\mathcal{A}^n_{1,2}(s)} - \mathbf{1}_{\mathcal{A}^n_{2,1}(s)})\left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{2,1}Z^n_{2,1}(s)\right)ds\right), \end{aligned}$$

where N^a_1, N^u_1, N^+_1 and N^-_1 are mutually independent unit rate (homogeneous) Poisson processes, and $\mathbf{1}_A$ is the indicator function that is equal to 1 if event A occurs, and to 0 otherwise.

Note that the representation of Q^n_1 is essentially a flow conservation equation (based on the memoryless property of the exponential distribution). That is, the queue at time t is all those customers who arrived by that time, captured by the Poisson process N^a_1, minus all the customers that abandoned, captured by the Poisson process N^u_1, minus all those who were routed into service, as captured by the last two Poisson processes in the expression. Similar expressions hold for the other processes in X^n.

We elaborate on how the intensities of the last two Poisson processes in the right-hand side (RHS) of the representation were obtained. First, if at time $s \in [0,T]$ the event $\mathcal{A}^n_{1,2}(s)$ in (12) holds, then any newly available agent in the system will take his next customer from the head of queue 1. Since agents become available at an instantaneous rate $\sum_{i,j} \mu_{i,j} Z^n_{i,j}(s)$ at time s, we get the third component in the RHS of $Q^n_1(t)$. Next we recall that, by the routing rule of FQR-ART, if at a time $s \in [0,T]$ $\mathcal{A}^n_{2,1}(s)$ in (12) holds, then any newly available agent takes his next customer from queue 2, in which case queue 1 will not decrease due to a service completion. If neither of the events $\mathcal{A}^n_{1,2}(s)$ or $\mathcal{A}^n_{2,1}(s)$ holds at a time s, then only service completions at pool 1 will cause a decrease at queue 1 due to a customer from that queue being routed to service. That explains the last term in the RHS of the representation.

Next, we exploit the fact that each of the Poisson processes in the representation minus its random intensity constitutes a martingale (again, see [34, 38]), e.g.,

$$M^{n,u}_1 \equiv N^u_1 \left(\theta_1 \int_0^t Q^n_1(s) ds \right) - \theta_1 \int_0^t Q^n_1(s) ds$$

is a martingale. Thus, subtracting and then adding all the random intensities, and using the fact that a sum of martingales is again a martingale, we get the following representation for the processes $Q^n_1, Q^n_2, Z^n_{1,2}, Z^n_{2,1}$ (the remaining two processes $Z^n_{1,1}$ and $Z^n_{2,2}$ are determined by (11)):

(13)

$$
\begin{aligned}
Q_1^n(t) &= M_1^n(t) + \int_0^t \lambda_1^n(s)ds - \int_0^t \theta_1 Q_1^n(s)ds \\
&\quad - \int_0^t \mathbf{1}_{\mathcal{A}_{1,2}^n(s)} \left(\mu_{1,1}Z_{1,1}^n(s) + \mu_{1,2}Z_{1,2}^n(s) + \mu_{2,1}Z_{2,1}^n(s) + \mu_{2,2}Z_{2,2}^n(s)\right) ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\mathcal{A}_{1,2}^n(s)} - \mathbf{1}_{\mathcal{A}_{2,1}^n(s)}) \left(\mu_{1,1}Z_{1,1}^n(s) + \mu_{2,1}Z_{2,1}^n(s)\right) ds, \\
Q_2^n(t) &= M_2^n(t) + \int_0^t \lambda_2^n(s)ds - \int_0^t \theta_2 Q_2^n(s)ds \\
&\quad - \int_0^t \mathbf{1}_{\mathcal{A}_{2,1}^n(s)} \left(\mu_{1,1}Z_{1,1}^n(s) + \mu_{1,2}Z_{1,2}^n(s) + \mu_{2,1}Z_{2,1}^n(s) + \mu_{2,2}Z_{2,2}^n(s)\right) ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\mathcal{A}_{1,2}^n(s)} - \mathbf{1}_{\mathcal{A}_{2,1}^n(s)}) \left(\mu_{2,2}Z_{2,2}^n(s) + \mu_{1,2}Z_{1,2}^n(s)\right) ds, \\
Z_{1,2}^n(t) &= M_{1,2}^n(t) + \int_0^t \mathbf{1}_{\mathcal{A}_{1,2}^n(s)}\mu_{2,2}Z_{2,2}^n(s)ds - \int_0^t (1 - \mathbf{1}_{\mathcal{A}_{1,2}^n(s)})\mu_{1,2}Z_{1,2}^n(s)ds, \\
Z_{2,1}^n(t) &= M_{2,1}^n(t) + \int_0^t 1_{\mathcal{A}_{1,2}^n(s)}\mu_{1,1}Z_{1,1}^n(s)ds - \int_0^t (1 - \mathbf{1}_{\mathcal{A}_{2,1}^n(s)})Z_{2,1}^n(s))ds,
\end{aligned}
$$

where $M_1^n, M_2^n, M_{1,2}^n$ and $M_{2,1}^n$ are the martingale terms alluded to above. It is not hard to show that those martingales are negligible in the fluid scaling (divided by n, e.g., $\bar{M}_i^n \equiv n^{-1}M_i^n$), i.e., that $\bar{M}_i^n \Rightarrow 0$ and $\bar{M}_{i,j}^n \Rightarrow 0$ as $n \to \infty$, uniformly over $[0,T]$, $i,j = 1,2$; see, e.g., Lemma 6.1 in [38]. Hence, we consider those martingales as a negligible stochastic noise that can be ignored for the purpose of developing the fluid approximation for (13).

To replace the stochastic integral representation in (13) with a deterministic one, we need to replace the indicator functions with smooth functions. This is where the AP comes in. What we do is replace the term $1_{\{D_{1,2}^n(t)>0\}}$ by the steady state probability that an associated fast-time scale process (FTSP) is greater than or equal to 0, denoted by $\pi_{1,2}(x(t))$, which is a function of the fluid state at time t, $x(t)$. Both $x(t)$ and $\pi(x(t))$ turn out to be a continuous function of t. This complicated step requires more explanation and justification, which again was the subject of [36, 37, 38].

We give a brief account in the rest of this section and the following one. We start by assuming that there is a fluid counterpart x for X^n in (13) which is *continuous and differentiable*. (This fact can be shown to hold by a minor modification of Corollary 5.1 in [38]). For any fluid point $x(t)$, let

(14)

$$
d_{1,2}(x(t)) \equiv q_1(t) - r_{1,2}q_2(t) - k_{1,2} \quad \text{and} \quad d_{2,1}(x(t)) \equiv r_{2,1}q_2(t) - q_1(t) - k_{2,1}.
$$

We first observe that, if $d_{i,j}(x(t)) > 0$ then, since $d_{i,j}(\cdot)$ is a continuous function, $d_{i,j}$ is strictly positive over an interval, and similarly if $d_{i,j} < 0$, $i, j = 1, 2$. In such cases the indicator functions are easy to deal with because each is a constant over the interval, and equals either 1 or 0. For example, if $d_{1,2}(x(t)) > 0$ for $t \in [s_1, s_2)$, for some $0 \le s_1 < s_2 < \infty$, and in addition, $Z^n_{2,1}(t) \le \tau^n_{2,1}$ over that interval for all n large enough, then

$$1_{\mathcal{A}^n_{1,2}(t)} \equiv \mathbf{1}_{\{\{D^n_{1,2}(t)>0\}\cap\{Z^n_{2,1}(t)\le\tau^n_{2,1}\}\}} = \mathbf{1}_{\{[s_2,s_2)\}}(t) \quad \text{for all } n \text{ large enough.}$$

Hence, a careful study is required for all $x(t) = \gamma$ in the *boundary sets* defined by

$$\mathbb{B}_{1,2} \equiv \{\gamma \in \mathbb{R}_6 : d_{1,2}(\gamma) = 0\} \quad \text{and} \quad \mathbb{B}_{2,1} \equiv \{\gamma \in \mathbb{R}_6 : d_{2,1}(\gamma) = 0\} \tag{15}$$

FQR-ART aims to "pull" the fluid model to one of these two boundary sets during overloads, when sharing is actively taking place, i.e., $\mathbb{B}_{i,j}$ is the region of the state space where we aim the fluid model to be when pool j helps class i, $i, j = 1, 2$.

Unfortunately, there is no straightforward fluid counterpart to the stochastic processes $D^n_{1,2}$ and $D^n_{2,1}$ when the fluid is in the boundary sets. However, there are two related stochastic processes, operating in an infinitely faster time scale, whose behavior determines the evolution of the fluid model, as we now explain.

5.2. *A Stochastic Averaging Principle.* For the discussion now, assume that $x(t) \in \mathbb{B}_{1,2}$ and consider $D^n_{1,2}$. To be able to apply the results in [38], we assume (for now) that the arrival rates are fixed (the arrival processes are homogeneous Poisson processes) and that $Z^n_{2,1} < \tau_{2,1}$, so that routing is determined solely on the value of $D^n_{1,2}$. In particular, sharing can take place if $D^n_{1,2}(t) > 0$. Then, by Theorem 4.5 in [38],

$$D^n_{1,2}(t) \Rightarrow D_{1,2}(x(t), \infty) \quad \text{in } \mathbb{R} \quad \text{as } n \to \infty, \tag{16}$$

where $D_{1,2}(\gamma, \cdot) \equiv \{D_{1,2}(\gamma, s) : s \ge 0\}$ is a CTMC associated with $\gamma \in \mathbb{R}_6$ whose distribution is determined by the value γ. (There is a different process for each γ.)

An analogous result holds for $D^n_{2,1}$ when $x(t) \in \mathbb{B}_{2,1}$. The notation $D_{i,j}(\gamma, \infty)$ stands for a random variable that has the steady-state distribution of the CTMC $D_{i,j}(\gamma, \cdot)$. Loosely speaking, $D^n_{i,j}$ moves so fast when $x(t)$ is in $\mathbb{B}_{i,j}$, that it reaches its steady state instantaneously as $n \to \infty$. Hence, we call $D_{i,j}(\gamma, \cdot)$ the *fast-time-scale process* (FTSP) associated with the point γ, or simply the FTSP.

Since we are interested in analyzing the indicator functions in (13), we first define for all $\gamma \in \mathbb{R}_6$

$$D_{i,j}(\gamma,\cdot) \equiv +\infty \quad \text{if} \quad d_{i,j}(\gamma) > 0 \quad \text{and} \quad D_{i,j}(\gamma,\cdot) \equiv -\infty \quad \text{if} \quad d_{i,j}(\gamma) < 0.$$

Next, we define

$$\begin{aligned} &\pi_{1,2}(\gamma) \equiv P(D_{1,2}(\gamma,\infty) > 0), \quad \text{for} \quad \gamma \in \mathbb{B}_{1,2} \quad \text{and} \\ &\pi_{2,1}(\gamma) \equiv P(D_{2,1}(\gamma,\infty) > 0), \quad \text{for} \quad \gamma \in \mathbb{B}_{2,1}. \end{aligned} \tag{17}$$

Now, by Theorem 4.1 in [38], which was proved for the process $D^n_{1,2}$ when $x \in \mathbb{B}_{1,2}$, and assuming that $Z^n_{2,1}(s) \le \tau^n_{2,1}$ over $[t_1,t_2]$ for all n large enough, we have that, as $n \to \infty$,

$$\int_{t_1}^{t_2} \mathbf{1}_{\mathcal{A}^n_{1,2}(s)} ds \equiv \int_{t_1}^{t_2} \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\le\tau^n_{2,1}\}\}} ds \Rightarrow \int_{t_1}^{t_2} \pi_{1,2}(x(s)) ds.$$

Similarly, if $x \in \mathbb{B}_{2,1}$ over an interval $[t_3,t_4]$, and $Z^n_{1,2}(s) \le \tau^n_{1,2}$ for all n large enough over that interval, we have

$$\int_{t_3}^{t_4} \mathbf{1}_{\mathcal{A}^n_{2,1}(s)} ds \equiv \int_{t_3}^{t_4} \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\le\tau^n_{1,2}\}\}} ds \Rightarrow \int_{t_3}^{t_4} \pi_{2,1}(x(s)) ds.$$

The convergence in both equations above holds uniformly.

We called these limits a "stochastic averaging principle", or simply an *averaging principle* (AP), since the process $D^n_{i,j}(t)$ is replaced by the *long-run average* behavior of the corresponding FTSP $D_{i,j}(x(t),\cdot)$ for each time t over the appropriate interval.

In the FQR-ART settings, the AP holds under the assumption that $Z^n_{i,j}$ lies below the appropriate release threshold over the interval $[t_1,t_2]$ for all n large enough (i.e., with probability converging to 1 as $n \to \infty$). If $Z^n_{i,j}$ is larger than the appropriate release threshold for all n large enough (again, with probability converging to 1) over $[t_1,t_2]$, then the limit of the integral considered above is clearly the 0 function. It remains to rigorously prove convergence theorems at points at which $Z^n_{i,j}(t) = \tau^n_{i,j} + o_P(n)$, where $o_P(n)$ denotes a random variable satisfying $o_P(n)/n \Rightarrow 0$ as $n \to \infty$. However, it is not hard to determine what the dynamics of the limit should be at such points if the limit exists. That is the basis for our heuristic fluid model approximation below.

5.3. *Representation via an ODE.* The heuristic limiting arguments above lead to the following fluid approximation for the X system under FQR-ART during overload periods. Considering an interval $[0,T]$ for which

$$z_{1,1}(t) + z_{2,1}(t) = m_1(t) \quad \text{and} \quad z_{2,2}(t) + z_{2,2}(t) = m_2(t) \quad \text{for all} \quad t \in [0,T], \tag{18}$$

together with an initial condition $x(0)$, the fluid model of X^n is the solution $x \equiv \{x(t) : t \geq 0\}$ over $[0, T]$ to the ODE:

(19)
$$\begin{aligned}
\dot{q}_1(t) &= \lambda_1(t) - \theta_1 q_1(t) - \Pi_{1,2}(x(t))\left(\mu_{1,1}z_{1,1}(t) + \mu_{1,2}z_{1,2}(t) + \mu_{2,1}z_{2,1}(t) + \mu_{2,2}z_{2,2}(t)\right) \\
&\quad - \left(1 - \Pi_{1,2}(x(t)) - \Pi_{2,1}(x(t))\right)\left(\mu_{1,1}z_{1,1}(t) + \mu_{2,1}z_{2,1}(t)\right), \\
\dot{q}_2(t) &= \lambda_2(t) - \theta_2 q_2(t) - \Pi_{2,1}(x(t))\left(\mu_{1,1}z_{1,1}(t) + \mu_{1,2}z_{1,2}(t) + \mu_{2,1}z_{2,1}(t) + \mu_{2,2}z_{2,2}(t)\right) \\
&\quad - \left(1 - \Pi_{1,2}(x(t)) - \Pi_{2,1}(x(t))\right)\left(\mu_{2,2}z_{2,2}(t) + \mu_{1,2}z_{1,2}(t)\right), \\
\dot{z}_{1,2}(t) &= \Pi_{1,2}(x(t))\mu_{2,2}z_{2,2}(t) - (1 - \Pi_{1,2}(x(t)))\mu_{1,2}z_{1,2}(t), \\
\dot{z}_{2,1}(t) &= \Pi_{2,1}(x(t))\mu_{1,1}z_{1,1}(t) - (1 - \Pi_{2,1}(x(t)))\mu_{2,1}z_{2,1}(t), \\
\dot{m}_1(t) &= \dot{z}_{1,1}(t) + \dot{z}_{2,1}(t), \\
\dot{m}_2(t) &= \dot{z}_{2,2}(t) + \dot{z}_{1,2}(t),
\end{aligned}$$

where, for $\pi_{i,j}(x(t))$ in (17), $i, j = 1, 2$,

$$\Pi_{i,j}(x(t)) := \begin{cases} \pi_{i,j}(x(t)) & \text{if } z_{j,i}(t) < \tau_{j,i}, \\ 0 & \text{otherwise.} \end{cases}$$

We remark that the ODE (19) can be equivalently represented by an integral equation resembling (13), but with the negligible martingale terms omitted, all the stochastic processes replaced by their fluid counterparts, and the indicator functions replaced by the appropriate $\Pi_{i,j}$ functions.

In practice we do not a-priori know the value of T, and there is a need to make sure that the ODE is a valid approximation for the stochastic system. We consider the ODE (19) valid (i.e., a legitimate representation of the evolution of the system) as long as the following two conditions are satisfied: (i) the two queues are strictly positive; (ii) if a queue is equal to 0 at some time $t \geq 0$, then the derivative of that queue is nonnegative at time t (so that the queue is nondecreasing at this time). When the ODE (19) is not valid, then other fluid models should be employed to approximate the system. We discuss such scenarios in §5.4 below.

We elaborate on Condition (ii). Consider, for example, the ODE for q_1 and assume that $q_1(t) = 0$ and $\dot{q}_1(t) < 0$ for some $t \geq 0$. Necessarily $\Pi_{1,2}(x(t)) = 0$, because $d_{1,2}(x(t)) \leq 0$, and the assumption that $\dot{q}_1(t) < 0$ implies that

$$\lambda_1(t) - (1 - \Pi_{2,1}(x(t)))(\mu_{1,1}z_{1,1}(t) + \mu_{2,1}z_{2,1}(t)) < 0. \tag{20}$$

In addition, since all the class-1 arrivals must immediately enter service (for otherwise, the queue will be increasing), it also holds that

$$\dot{z}_{1,1}(t) = \lambda_1(t) - \mu_{1,1}z_{1,1}(t).$$

Hence,

$$\begin{aligned}\dot{z}_{1,1}(t)+\dot{z}_{2,1}(t) &= \lambda_1(t)-\mu_{1,1}z_{1,1}(t)\\ &\quad +\Pi_{2,1}(x(t))\mu_{1,1}z_{1,1}(t)-(1-\Pi_{2,1}(x(t)))\mu_{2,1}z_{2,1}(t)\\ &= \lambda_1(t)-(1-\Pi_{2,1}(x(t)))(\mu_{1,1}z_{1,1}(t)+\mu_{2,1}z_{2,1}(t))\\ &< 0,\end{aligned} \tag{21}$$

where the inequality follows from (20).

Now, since $\dot{m}_1(t)=\dot{z}_{1,1}(t)+\dot{z}_{2,1}(t)$, we see that pool 1 can remain full just after time t only if $m_1(t)$ happens to decrease exactly as in (21). However, q_1 is becoming negative, so that the ODE is not valid. On the other hand, if (21) holds (which ODE (19) enforces to be equal to $\dot{m}_1(t)$) and $q_1(t)=0$, then necessarily $\dot{q}_1(t)<0$, so that the queue is becoming negative. In either case, we see that the ODE is valid as an approximation for the stochastic system when $q_1(t)=0$ only if pool 1 can be kept full without enforcing q_1 to become negative. Similar reasonings hold for the q_2 and m_2 processes.

5.4. *The Fluid Model When There is No Active Sharing.* The ODE for the fluid model above was developed for all cases for which both pools are full, i.e., (18) holds. This is the main case because systems are typically designed to operate with very little extra service capacity (if any), and is the primary case when overloads occur. Nevertheless, the system may go through periods in which at least one of the pools is underloaded. Hence, we now briefly describe the fluid models for underloaded pools.

Consider an interval $I\subset[0,\infty)$. If no sharing takes place and $z_{1,2}(t)=z_{2,1}(t)=0$ for all $t\in I$, then the two classes operate as two independent single-pool models (with time-varying parameters and staffing) over that interval I, to which fluid limits are easy to establish. Specifically, assuming without loss of generality, that $I=[0,s)$ for some $0<s<\infty$, the fluid dynamics of both classes obey the ODE

$$\begin{aligned}\dot{q}_i(t) &= (\lambda_i(t)-\mu_{i,i}z_{i,i}(t)-\theta_i q_i(t))\mathbf{1}_{\{q_i(t)\geq 0\}}\\ \dot{z}_{i,i}(t) &= \begin{cases}\dot{m}_i(t) & \text{if } q_i(t)>0,\\ \lambda_i(t)\mathbf{1}_{\{z_{i,i}(t)\leq m_i(t)\}}-\mu_{i,i}z_{i,i}(t) & \text{if } q_i(t)=0.\end{cases}\end{aligned} \tag{22}$$

In the *time-invariant case*, when the arrival rates and staffing functions are fixed constants, the unique solution for a given initial condition to the ODE

in (22) is easily seen to be

$$
\begin{aligned}
q_i(t) &= \left(\frac{\lambda_i - \mu_{i,i} m_i}{\theta_i} + \left(q_i(0) - \frac{\lambda_i - \mu_{i,i} m_i}{\theta_i} \right) e^{-\theta_i t} \right) \vee 0, \\
z_{i,i}(t) &= \begin{cases} m_{i,i} & \text{if } q_i(t) > 0, \\ \frac{\lambda_i}{\mu_{i,i}} + \left(z_{i,i}(0) - \frac{\lambda_i}{\mu_{i,i}} \right) e^{-\mu_{i,i} t} & \text{if } q_i(t) = 0. \end{cases}
\end{aligned} \tag{23}
$$

where $a \vee b \equiv \max\{a, b\}$ and $(q_1(0), q_2(0), z_{1,1}(0), z_{2,2}(0))$ is a deterministic vector in $[0, \infty)^2 \times [0, m_1] \times [0, m_2]$.

If $z_{1,2}(s_0) > 0$ (or $z_{2,1}(s_0) > 0$) for some $s_0 \geq 0$ and there is no active sharing over the interval $[s_0, s_1)$, then $z_{1,2}$ ($z_{2,1}$) is strictly decreasing over that interval. Then $z_{i,j}$, $i \neq j$, satisfies the ODE

$$\dot{z}_{i,j}(t) = -\mu_{i,j} z_{i,j}(t), \quad s_0 \leq t < s_1$$

which is the same as the ODE for $z_{i,j}$ in (19) with $\Pi_{i,j} = 0$.

Remark 5.1 A proof of existence of a unique solution to the ODE (19) following the lines of [37] requires showing that the RHS is a local Lipschitz continuous function of x and is piecewise continuous in t. We do not prove such a result here, but it is important to consider arrival rates and staffing functions that ensure that the right side of the ODE satisfies the piecewise continuity condition in the time argument.

6. Solving the ODE. To appreciate that the algorithm cannot be a routine solution of an ODE, observe that computing the solution to (19) requires computing the two steady-state probabilities $\pi_{1,2}(x(t))$ and $\pi_{2,1}(x(t))$ for all times t and states $x(t) \in \mathbb{R}_6$. Simplification is achieved when $r_{1,2} = r_{2,1} = 1$, because the FTSP's $D_{i,j}(x(t), \cdot)$, $i, j = 1, 2$, become simple *birth-and-death* (BD) processes. To facilitate the discussion we thus consider this simpler case and refer to §6.2 in [37] for the treatment of the FTSP $D_{1,2}$ as a *quasi-birth-and-death process* (QBD) when the ratio parameters are not equal to 1. (In [37] FQR-T is studied with one overload incident, with pool 1 receiving help, but the same method can be applied to $D_{2,1}$ with sharing in the opposite direction.)

For simplicity, we again start by assuming that the arrival processes are homogenous Poisson processes, having constant arrival rates λ_1 and λ_2 over $[0, T]$, and that the staffing functions are also fixed over that time interval at m_1 and m_2. Recall that $D_{i,j}(\gamma, \cdot) \equiv \infty$ if $d_{i,j}(\gamma) > 0$ and $D_{i,j}(\gamma, \cdot) \equiv -\infty$ if $d_{i,j}(\gamma) < 0$, and let $\mathbb{A}_{1,2}$ and $\mathbb{A}_{2,1}$ be the subsets of $\mathbb{R}_6$ in which the FTSP's $D_{1,2}(\gamma, \cdot)$ and $D_{2,1}(\gamma, \cdot)$ are positive recurrent, i.e.,

$$\mathbb{A}_{1,2} \equiv \{\gamma \in \mathbb{B}_{1,2} : 0 < \pi_{1,2}(\gamma) < 1\} \quad \text{and} \quad \mathbb{A}_{2,1} \equiv \{\gamma \in \mathbb{B}_{2,1} : 0 < \pi_{2,1}(\gamma) < 1\}. \tag{24}$$

By definition, if the fluid model at time t is in $\mathbb{A}_{i,j}$, i.e., $x(t) \in \mathbb{A}_{i,j}$, then $d_{i,j}(x(t)) = 0$. However, if $d_{i,j}(x(t)) = 0$, then $x(t)$ is not necessarily in $\mathbb{A}_{i,j}$, because the FTSP $D_{i,j}(x(t), \cdot)$ may be transient (drift to $+\infty$ or $-\infty$) or null recurrent; in particular, *The evolution of the fluid model is determined by the distributional characteristics of the FTSP's* $D_{1,2}$ *and* $D_{2,1}$. Hence, even before we try to compute $\pi_{i,j}(x(t))$, which is necessary in order to solve the ODE (19), there is a need to determine whether $x(t)$ is in one of the sets $\mathbb{A}_{1,2}$ or $\mathbb{A}_{2,1}$. We focus on $D_{1,2}$, with the analysis of $D_{2,1}$ being similar.

To determine the behavior of the FTSP $D_{1,2}$ it is again helpful to think of x as a fluid limit of the fluid-scaled sequence $\{\bar{X}^n : n \geq 1\}$ and to recall that $D_{1,2}$ was achieved as a limit of $D^n_{1,2}$ without any scaling; see (16). (See also Theorem 4.4 in [38] which provides a process-level limit relating $D_{1,2}$ and $D^n_{1,2}$.) Hence, both processes are defined on the same state space, which, for $r_{1,2} = 1$, is $\mathbb{Z} \equiv \{\ldots, -1, 0, 1, \ldots\}$.

Now, for a fixed $x(t)$, when $D_{1,2}(x(t), \cdot) = m > 0$, the birth and death rates of the FTSP are, respectively,

$$\lambda^+(x(t), m) \equiv \lambda_1 + \theta_2 q_2(t),$$
$$\mu^+(x(t), m) \equiv \lambda_2 + \mu_{1,1} z_{1,1}(t) + \mu_{1,2} z_{1,2}(t) + \mu_{2,1} z_{2,1}(t) + \mu_{2,2} z_{2,2}(t) + \theta_1 q_1(t).$$

In analogy to the (non-Markov) process $D^n_{1,2} = Q^n_1 - Q^n_2 - k^n_{1,2}$, $\lambda_+(x(t), m)$ corresponds to an increase of $D_{1,2}$ due to arrival to queue 1 plus an abandonment from queue 2 (since either one of these two events cause an increase by 1 of $D^n_{1,2}$ in the stochastic system). Since any other event causes $D^n_{1,2}$ to decrease by 1, due to the scheduling rules of FQR-ART, we get the expression for $\mu^+(x(t), m)$.

Next, if $D_{1,2}(x(t), m) = m \leq 0$, the birth and death rates are, respectively,

$$\lambda^-(x(t), m) \equiv \lambda_1 + \mu_{2,2} z_{2,2}(t) + \mu_{1,2} z_{1,2}(t) + \theta_2 q_2(t),$$
$$\mu^-(x(t), m) \equiv \lambda_2 + \mu_{1,1} z_{1,1}(t) + \mu_{2,1} z_{2,1}(t) + \theta_1 q_1(t).$$

Again, whenever $D^n_{1,2}$ is non-positive and sharing is taking place with pool 2 helping class 1, a "birth" occurs if there is an arrival to queue 1 or an abandonment from queue 2, or if there is a service completion in pool 2 (since then a newly available type-2 agent takes his next customer from queue 2). Similarly, a "death" occurs if there is an arrival to class 2, an abandonment from queue 1, or a service completion in pool 1.

We see that the FTSP $D_{1,2}(x(t), \cdot)$ is a two-sided $M/M/1$ queue, i.e., it behaves like an $M/M/1$ queue with "arrival rate" $\lambda^+(x(t), m)$ and "service rate" $\mu^+(x(t), m)$ for all $m > 0$, and behaves like a different $M/M/1$ queue with "arrival rate" $\mu^-(x(t), m)$ and "service rate" $\lambda^-(x(t), m)$, for all $m \leq 0$.

Thus, for

$$\delta^+(\gamma) \equiv \lambda^+(\gamma,\cdot) - \mu^+(\gamma,\cdot) \quad \text{and} \quad \delta^-(\gamma) \equiv \lambda^-(\gamma,\cdot) - \mu^-(\gamma,\cdot), \quad \gamma \in \mathbb{B}_{1,2},$$

the set $\mathbb{A}_{1,2}$ can be characterized via

$$\mathbb{A}_{1,2} \equiv \{\gamma \in \mathbb{B}_{1,2} : \delta^+(\gamma) < 0 < \delta^-(\gamma)\}.$$

Next, letting $T^+(\gamma)$ and $T^-(\gamma)$ denote, respectively, the busy period of the $M/M/1$ in the positive region and the busy period of the $M/M/1$ in the negative region, and using simple alternating renewal arguments for the renewal process $D_{1,2}(\gamma,\cdot)$, we have

$$\pi_{1,2}(\gamma) = \frac{E[T^+(\gamma)]}{E[T^+(\gamma)] + E[T^-(\gamma)]}, \tag{25}$$

where, from basic $M/M/1$ theory,

$$E[T^{\pm}(\gamma)] = \frac{1}{\mu^{\pm}(\gamma) - \lambda^{\pm}(\gamma)}.$$

Note that if $d_{1,2}(\gamma) = 0$ but $\gamma \notin \mathbb{A}_{1,2}$, then $\pi_{1,2}(\gamma)$ is equal to either 1 or 0. In particular,

(26)
$$\text{if} \;\; \delta^+(\gamma) \geq 0, \;\; \text{then} \;\; \pi_{1,2}(\gamma) = 1 \;\; \text{and if} \;\; \delta^-(\gamma) \leq 0 \;\; \text{then} \;\; \pi_{1,2}(\gamma) = 0.$$

There are no other options, since for any $\gamma = x(t)$ for which both pools are full (as is required for the ODE (19) to be valid), it holds that

$$\delta^-(x(t)) - \delta^+(x(t)) = 2(\mu_{1,2} z_{1,2}(t) + \mu_{2,2} z_{2,2}(t)) > 0,$$

where the inequality above follows from the fact that $z_{1,2}(t) + z_{2,2}(t) = m_2(t) > 0$.

We see that the sets $\mathbb{A}_{i,j}$ and the computation of $\pi_{i,j}(\cdot)$ are completely determined by the staffing, arrival rates, service and abandonment rates for any given point $\gamma \in \mathbb{R}_6$, where the only points that require careful analysis are those in one of the two sets $\mathbb{B}_{i,j}$. However, recall that we have assumed for simplicity that the arrival rates and staffing functions are not time dependent. If, instead, the arrival rates or the staffing functions are time dependent, then the distribution of the FTSP $D_{i,j}(x(t),\cdot)$ is also time dependent. In particular, given a $\gamma \in \mathbb{R}_6$ we cannot determine whether $D_{1,2}(\gamma,\cdot)$ is positive recurrent or not, since that may depend on the time $t \in [0,T]$. Thus, the sets at which the FTSP's are ergodic are themselves

time dependent. Hence, for a full analysis, we would need to consider sets of the form $\{\mathbb{A}_{i,j}(t) : t \in [0,T]\}$, where

$$\mathbb{A}_{i,j}(t) \equiv \{(\gamma,t) \in \mathbb{B}_{i,j} \times \mathbb{R}_+ : \delta^+(\gamma,t) < 0 < \delta^-(\gamma,t)\}, \tag{27}$$

where $\delta^+(\gamma,t)$ and $\delta^-(\gamma,t)$ are the drifts of the FTSP $D_{1,2}(\gamma,\cdot)$ at the point γ at time t. Fortunately, for the purpose of solving the ODE, we do not actually need to characterize the sets $\{\mathbb{A}_{i,j}(t) : t \in [0,T]\}$, because we can determine whether $D_{i,j}(x(t),\cdot)$ is ergodic at each time t as we solve the ODE.

6.1. *A Numerical Algorithm to Solve the ODE.* Given the ODE in (19) with a fully specified RHS at each t, we compute the solution x over an interval $[0,T]$ by employing the classical Euler method, combined with the AP. Given a step size h and the time T, the number of iterations needed is $N \equiv T/h$. Let $\dot{x} = \Psi(x)$, where $\Psi(x)$ is the RHS of the appropriate ODE, e.g., if both pools are full, then $\Psi(x)$ is the RHS of (19). Given $x(0)$, we can compute $x(h)$ using the first Euler step: $x(h) = x(0) + h\Psi(x(0))$. Given $x(h)$ we can compute $\Pi_{1,2}(x(h))$ and $\Pi_{2,1}(x(h))$, if needed, and then compute $x(2h)$ using the second Euler step. In general, the solution to the ODE is computed via

$$x((k+1)h) = x(kh) + h\Psi(x(kh)), \quad 0 \le k \le N,$$

where at each step, if $x(kh) \in \mathbb{B}_{1,2}$ or $x(kh) \in \mathbb{B}_{2,1}$, we can compute $\Pi_{1,2}(kh)$ and $\Pi_{2,1}(kh)$ as explained above.

The algorithm just described remains unchanged when the ratio parameters are general (not equal to 1), except that the sets $\mathbb{A}_{i,j}$ and the computations of $\pi_{i,j}$ are more complicated (the FTSP's are no longer BD processes). We refer to [37] for these more complicated settings.

To evaluate the RHS in each step, we use the analysis in §6, starting at a given initial condition $x(0)$, since we can now determine the value of $\Pi_{i,j}(x(t))$ for each $t \ge 0$. For example, if at a time $t \ge 0$ $d_{1,2}(x(t)) = 0$, then we check whether (27) holds, so that $x(t) \in \mathbb{A}_{1,2}(t)$. If $z_{2,1}(t) \le \tau_{2,1}$, then $\Pi_{1,2}(x(t)) = \pi_{1,2}(x(t))$ and it can be computed using (25). If $z_{2,1}(t) > \tau_{2,1}$, then $\Pi_{2,1}(t) = 0$. If $d_{1,2}(x(t)) = 0$ but $x(t) \notin \mathbb{A}_{1,2}(t)$, i.e., if (27) does not hold, then we can determine the value of $\pi_{1,2}(x(t))$, and thus of $\Pi_{1,2}(x(t))$, by computing the drifts of the FTSP and employing (26) (replacing the drifts in (26) with the time dependent drifts as in (27)). Similarly we can compute the value of $\Pi_{2,1}(x(t))$ whenever $d_{2,1}(x(t)) = 0$.

In all other regions of the state space for which both pools are full, i.e., $z_{i,j}(t) + z_{j,i}(t) = m_j(t)$, $i \ne j$, we can easily determine the value of $\pi_{1,2}(x(t))$ by considering whether $d_{i,j}(x(t))$ is bigger or smaller than 0. For example,

if at time $t \geq 0$ $d_{1,2}(x(t)) > 0$, then $\pi_{1,2}(x(t)) = 1$ and if $d_{1,2}(x(t)) < 0$, then $\pi_{1,2}(x(t)) = 0$. This, together with the value of $z_{2,1}(t)$, immediately gives the value of $\Pi_{1,2}(x(t))$.

We need to use other fluid equations when at least one of the two pools is not full. If, for example $z_{1,1}(t) + z_{2,1}(t) < m_1(t)$, then necessarily $q_1(t) = 0 < k_{1,2}$, so that

$$\dot{z}_{1,2}(t) = -\mu_{1,2} z_{1,2}(t) \quad \text{and} \quad \dot{z}_{1,1}(t) = \lambda_1(t) - \mu_{1,1} z_{1,1}(t).$$

The evolution of $z_{2,1}$ in this case is determined by whether $q_2(t) < k_{2,1}$ or $q_2(t) \geq k_{2,1}$. In the first case $z_{2,1}(t)$ must be strictly decreasing at time t if it is positive, or remain at 0 otherwise. In the latter case, when $q_2(t) \geq k_{2,1}$, the excess fluid - that is not routed to pool 2 and does not abandon, if such excess fluid exists - is flowing to pool 1. We thus have $\dot{z}_{2,1}(t)$ is equal to

(28)
$$\begin{array}{ll} -\mu_{2,1} z_{2,1}(t) & \text{if } q_2(t) < k_{2,1} \\ -\mu_{2,1} z_{2,1}(t) + (\lambda_2(t) - \mu_{2,2} z_{2,2}(t) - \mu_{1,2} z_{1,2}(t) - \theta_2 k_{2,1})^+ & \text{if } q_2(t) = k_{2,1} \end{array}$$

Similar reasonings lead to the fluid model of $z_{1,2}$ when pool 1 is full, but pool 2 has spare capacity.

If both pools have spare capacity at time t, then $q_1(t) = q_2(t) = 0$ and

$$\dot{z}_{i,j}(t) = -\mu_{i,j} z_{i,j}(t) \quad \text{and} \quad \dot{z}_{i,i}(t) = \lambda_i - \mu_{i,i} z_{i,i}(t), \quad i, j = 1, 2, \quad i \neq j.$$

Remark 6.1 If at iteration $k \geq 0$ the solution lies outside the set $\mathbb{B}_{1,2} \cup \mathbb{B}_{2,1}$, then due to the discreteness of the algorithm, there is a need to ensure that the boundary is not missed in the following iterations. Hence, if in the k^{th} iteration $d_{1,2}(x(kh)) > 0$ (< 0) and in the $(k+1)^{st}$ iteration $d_{1,2}((x(k+1)h)) < 0$ (> 0), then the boundary $d_{1,2}$ necessarily was missed, because the fluid is continuous, and so we set $d_{1,2}((x(k+1)h)) = 0$. We then check whether $x((k+1)h) \in \mathbb{A}_{1,2}((k+1)h)$, compute $\pi_{1,2}(x(k+1)h)$ and use its value to compute the value in the $(k+2)^{nd}$ iteration. It is significant that *we do not force the solution to be on the boundary*, e.g., we do not compute $q_1((k+1)h)$ and use its value to compute $q_2((k+1)h)$ via

$$q_2((k+1)h) = q_1((k+1)h) - k_{1,2}. \tag{29}$$

We solve the six-dimensional ODE in (19), and if indeed (29) holds whenever it should, then we have a good indication that the algorithm works. That is, we can check at which iteration the boundary $\mathbb{B}_{1,2}$ was hit, and then observe if $q_1(t) - q_2(t) = k_{1,2}$ over an interval for which we have indication that this should hold. (Of course, the solution to the algorithm might leave the boundary for legitimate reasons, i.e., because the fluid model leaves it.)

7. Numerical Examples. We now study three examples. The first two are piecewise-continuous models, whereas the third is for a general time-varying model. In all three examples the system starts empty, so that we also check the numerical algorithm in periods when (18) does not hold, as in §5.4.

We compare the numerical solutions to the ODE to simulations, to see how well the fluid model approximates stochastic systems. In the first two examples we simulate three systems, each can be considered as a component in a sequence $\{\bar{X}^n : n \geq 1\}$. In the smallest system we take 50 agents in each service pool, in the middle one there are 100 agents in a pool, and the largest has 400 agents in each pool, i.e., we simulate $\bar{X}^n$ for $n = 50, 100, 400$. That allows us to observe the "convergence" of the stochastic system to the fluid approximation. We plot the fluid and simulation results together, normalized to $n = 10$. (E.g., for the system with 400 agents in each pool we divide all processes by 40.)

The following parameters are used for all three simulations:

$\mu_{1,1} = \mu_{2,2} = 1$; $\mu_{1,2} = \mu_{2,1} = 0.8$, $\theta_1 = \theta_2 = 0.5$. In addition, we take $r_{1,2} = r_{2,1} = 1$. We take $k^n_{1,2} = k^n_{2,1} = 0.3n$; $\tau^n_{1,2} = \tau^n_{2,1} = 0.02n$, so that, for $n = 50, 100, 400$, we have $k^n_{1,2} = k^n_{2,1} = 15, 30, 120$ and $\tau^n_{1,2} = \tau^n_{2,1} = 1, 2, 8$, respectively.

7.1. *A Single Overload Incident.* The first example aims to check whether FQR-ART detects overloads automatically when they occur and starts sharing in the right direction, and whether, once an overload incident is over, FQR-ART avoids oscillations. In particular, over the time interval $[0, 60]$ the arrival rates are as follows: $\lambda^n_2 = n$ throughout that time interval. Over $[0, 20)$ and $[40, 60]$ the arrival rate to pool 1 is $\lambda^n_1 = n$. Hence, both pools are normally loaded during these two subintervals. However, during the interval $[20, 40)$ the arrival rate of class 1 changes to $\lambda^n_1 = 1.4n$, so that, during $[20, 40)$ the system is overloaded, and pool 2 should be helping class 1.

We compare the solution to the fluid equations, solved using the algorithm, to an average of 1000 independent simulation runs for the three cases $n = 50, 100, 400$. The results are shown in Figures 10-12 below. In addition Figure 13 plots $q_1 - r_{1,2}q_2 - k_{1,2}$. Since shortly after time 20 the value is 0 in Figure 13, we have a strong indication that the numerical solution is correct, because during most of the overload period, when sharing takes place, it should hold that $d_{1,2}(x(t)) = 0$.

The simulation experiments indicate that the fluid model approximates well the mean behavior of the system even for relatively small systems, e.g., when $n = 50$. Of course, the accuracy of the approximation grows as n

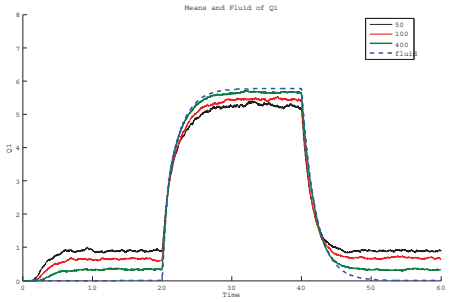

FIG 10. *comparison of the fluid model to simulations of* $10\bar{Q}_1^n$ *for* $n = 50$, 100 *and* 400 *with a single overload*

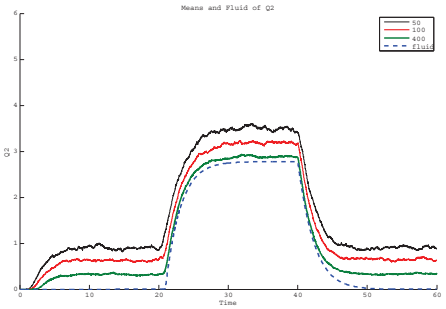

FIG 11. *comparison of the fluid model to simulations of* $10\bar{Q}_2^n$ *for* $n = 50$, 100 *and* 400 *with a single overload*

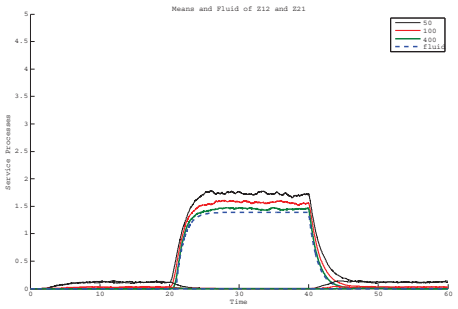

FIG 12. *comparison of the fluid model to simulations of* $10\bar{Z}_{1,2}^n$ *for* $n = 50$, 100 *and* 400 *with a single overload*

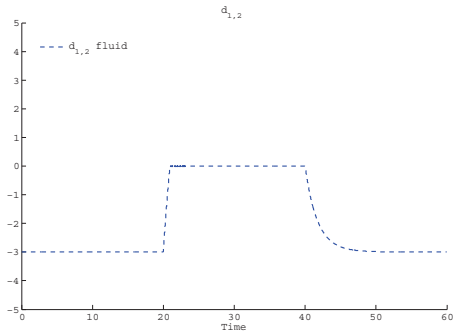

FIG 13. *plot of* $q_1 - r_{1,2}q_2 - k_{1,2}$ *in the single overload example*

becomes larger. The simulation experiments show that FQR-ART quickly detects the overload and the correct direction of sharing. Moreover, the control ensures that there are no oscillations, as in §4.

Another observation is that when the system is normally loaded and there is no sharing, the fluid model, which has null queues, does not describe the queues well. In those cases there is an increased importance to stochastic refinements for the queues. If there is only negligible sharing, as FQR-ART ensures, then such stochastic refinements are well approximated by diffusion limits for the Erlang A model, as in [15].

7.2. *Switching Overloads.* In the second example we consider an overloaded system, with pool 1 being overloaded initially, and with the direction of overload switching after some time, making pool 2 overloaded. Specifi-

cally, we let the arrival rates be $\lambda_1^n = 1.4n$ and $\lambda_2^n = n$ over $[0, 20)$, and $\lambda_1^n = n$, $\lambda_2^n = 1.4n$ on $[20, 40]$. The results are plotted in Figures 14-16. Figure 17 plots $q_1 - r_{1,2}q_2 - k_{1,2}$ and $r_{2,1}q_2 - q_1 - k_{2,1}$.

Once again, the fact that the appropriate difference process equals to 0 shortly after the corresponding overload begins is an indication that the solution to the ODE is correct, since each queue is calculated via the averaging principle, without forcing the relations $d_{1,2}(x(t)) = 0$ and $d_{2,1}(x(t)) = 0$.

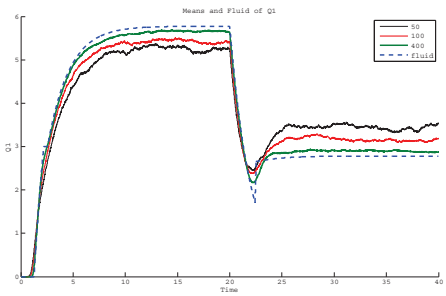

FIG 14. *comparison of the fluid model to simulations of* $10\bar{Q}_1^n$ *for* $n = 50$, 100 *and* 400 *with the switching overloads*

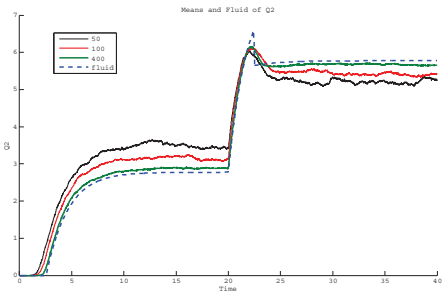

FIG 15. *comparison of the fluid model to simulations of* $10\bar{Q}_2^n$ *for* $n = 50$, 100 *and* 400 *with the switching overloads*

As in the figures in §7.1, it is easily seen from the figures above that the fluid model approaches a fixed point, so long as the arrival rates are fixed. Then, once a change in the rates occurs, the fluid goes through a new transient period until it relaxes in a new fixed point.

7.3. *General Non-stationary Model with Switching Overloads.* We next test our algorithm in a more challenging time-varying example. This example is unrealistic in call-center setting, because the arrival rates and staffing functions are not likely to change so drastically, but it demonstrates the robustness of our fluid model and of the algorithm.

We assume that the arrival rate to pool 1 over the time period $[0, 20)$ is sinusoidal. We further assume that management anticipated the basic sinusoidal pattern of the arrival rate, but did not anticipated the magnitude, so that pool 1 is overloaded. To specify the staffing with the sinusoidal arrival rate, we assume that staffing follows the appropriate *infinite-server* approximation; see, e.g., Equation (9) in [13]. The purpose of that staffing rule in our setting, is to stabilize the system at a fixed point eventually, as in the examples above. In particular, for $t \in [0, 20]$, we let

$$\lambda_1^n(t) = 1.3n + 0.1n\sin(t) \text{ and } m_1^n(t) = n + 0.05n[\sin(t) - \cos(t)];$$

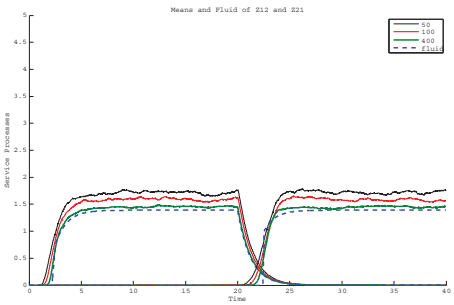

FIG 16. *comparison of the fluid model to simulations of* $10\bar{Z}^n_{1,2}$ *for* $n = 50$, 100 *and* 400 *with the switching overloads*

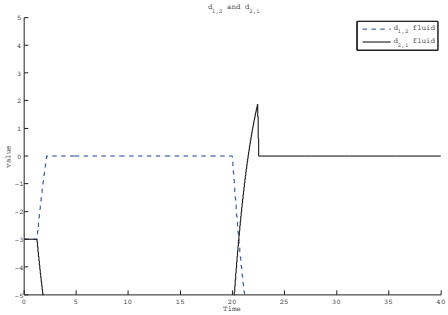

FIG 17. *the two fluid difference processes with the switching overloads*

$$\lambda^n_2(t) = n \text{ and } m^n_2(t) = n.$$

Then, on the time interval $[20, 40]$ the overload switches, with pool 2 becoming overloaded and experiencing a sinusoidal arrival rate. However, we now take fixed staffing in both service pools. In particular, the parameters over the second overload interval $[20, 40]$ are

$$\lambda^n_1(t) = n \text{ and } m^n_1(t) = n; \quad \lambda^n_2(t) = 1.1n + 0.1n\sin(t) \text{ and } m^n_2(t) = n.$$

Thus, we test two overload settings in this example. In the first interval, we can see whether the fluid approximation stabilizes. Since there is sharing of class-1 customers, previous results such as in [29] do not apply directly to our case. In the second interval, we expect to see a sinusoidal behavior of the system, because the staffing in both pools is fixed. In particular, the fluid model should not approach a fixed point after the switch at time $t = 20$.

We compare the fluid approximation to simulations for $n = 100$ and $n = 400$. Figures 18–21 demonstrate the effectiveness of the fluid model and the numerical algorithm. As expected, the fluid over $[0, 20)$ approaches a fixed point, and exhibits a sinusoidal behavior after $t = 20$, with the accuracy of the fluid approximation increasing in the scale parameter n.

As was mentioned above, the fluid model requires special care when the staffing functions are decreasing; we refer again to [29]. Figure 22 shows the actual number of agents in Pool 1 for the case $n = 100$ (the average of the 1000 simulations), and the staffing function $m^n_1(t)$ given above. Clearly, the fluid model follows the actual staffing closely. We further note that there is a downward jump in the staffing function at time $t = 20$. In the fluid model, we simply eliminated the appropriate amount of staffing from the pool, together with the fluid that was processed with that removed capacity

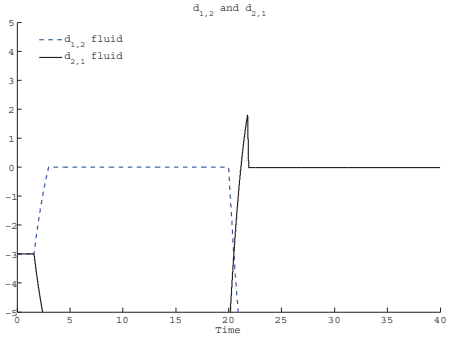

FIG 18. *the two fluid difference functions* $d_{1,2}$ *and* $d_{2,1}$ *with the switching sinusoidal overloads*

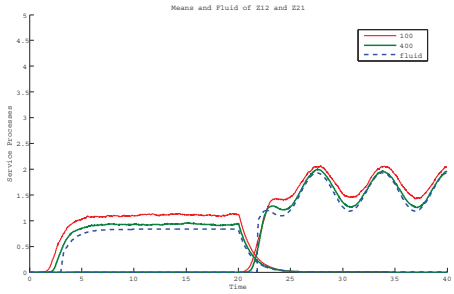

FIG 19. *comparison of the fluid model to simulations of* $10\bar{Z}^n_{1,2}$ *and* $10\bar{Z}^n_{2,1}$ *for* $n = 100$ *and* 400 *with the switching sinusoidal overloads*

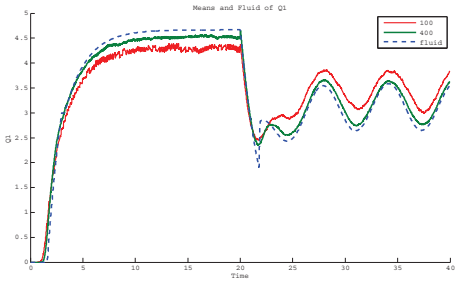

FIG 20. *comparison of the fluid model to simulations of* $10\bar{Q}^n_1$ *for* $n = 100$ *and* 400 *with the switching sinusoidal overloads*

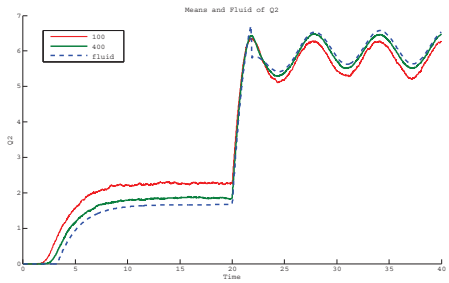

FIG 21. *comparison of the fluid model to simulations of* $10\bar{Q}^n_2$ *for* $n = 100$ *and* 400 *with the switching sinusoidal overloads*

(this fluid in service is lost). However, in the simulation, agents are removed only when they are done serving, so there is no jump in the actual staffing at $t = 20$, and no customer in service is lost. Nevertheless, the fluid model with the jump is clearly a good approximation for the stochastic model with no jump. This behavior is to be expected, since there are many service completions over short time intervals in large systems.

7.4. *The Oscillatory Model.* Our final examples show that the fluid model can also predict the bad oscillatory behavior. Here we consider the fluid model of the examples shown in Figures 4–7 in §4. In particular, the parameters are $\mu_{1,1} = \mu_{2,2} = 1$, $\mu_{1,2} = \mu_{2,1} = 0.1$, $\lambda_1 = \lambda_2 = 98$, $m_1 = m_2 = 100$ and $\tau_{i,j} = 0.01$ and $k_{i,j} = 10$, $i, j = 1, 2$. Figures 23 and 24 show the

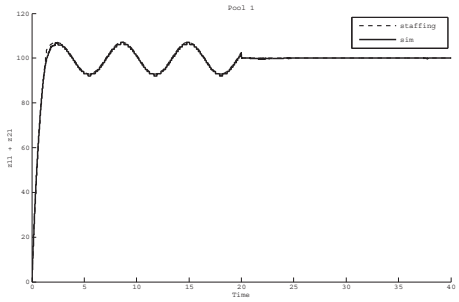

FIG 22. *Fluid vs. simulations: Number of agents in pool* 1

fluid solution to the system with no abandonment (in (19) we simply plug $\theta_1 = \theta_2 = 0$), whereas Figures 25 and 26 show the fluid solution to (19) with $\theta_1 = \theta_2 = 0.01$.

However, the initial conditions here are different than in Figures 4–7. We now take $z_{1,1}(0) = m_1 = 100$ and $z_{1,2}(0) = m_2 - z_{2,2}(0) = 20$. The reason is that, if the fluid is initialized with no sharing and no queues, then its components $(q_1, q_2, z_{1,2}, z_{2,1})$ are fixed at $(0, 0, 0, 0)$, i.e., there is never any sharing, and the fluid queues are constant at zero. However, if it is initialized at states with some sharing, then it may get stuck at an oscillatory equilibrium, as shown in Figures 23 – 26. In particular, this is a numerical example that the fluid model may be *bi-stable*, namely, have two very different stationary behaviors. To which stationary behavior the fluid ends up converging depends on the initial condition.

This fluid bi-stability property has two immediate implications to the stochastic system. First, once an overload incident is ending, with substantial sharing taking place, the system may start to oscillate. Indeed, this is the case in the example shown in Figures 8. The second implication is that the no-sharing equilibrium may be unstable in practice, because stochastic noise can eventually "push" the system out of this equilibrium, and cause it to oscillate. That was demonstrated in Figures 4 and 5 in §4. (Recall that the initial condition of the example in §4 was of an empty system. In particular, with no sharing initially.) Note also that the time scale in Figures 23 and 24 is shorter than in Figures 25 and 26. As for the corresponding figures in §4, the time scale of the second example is longer to make it clear that the system with abandonment converges to an oscillatory equilibrium.

8. Conclusions. In this paper we studied a time-varying X model experiencing periods of overloads. While our previous FQR-T control is effective

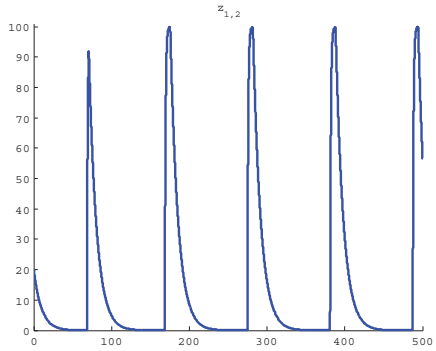

FIG 23. *Oscillations $z_{1,2}(t)$ in the fluid model of the extreme example with no abandonment.*

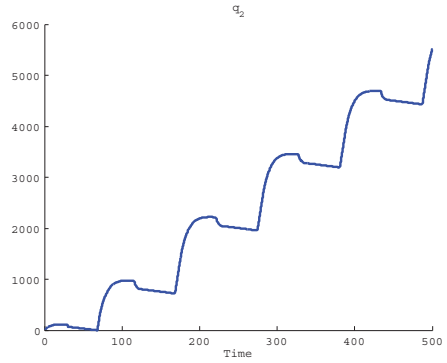

FIG 24. *Oscillating growth of the content $q_2(t)$ in the fluid model of the extreme example with no abandonment.*

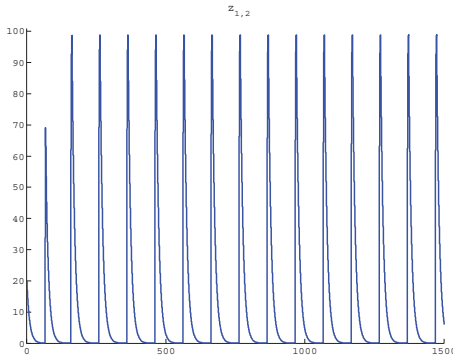

FIG 25. *Oscillations $z_{1,2}(t)$ in the fluid model of the extreme example with abandonment.*

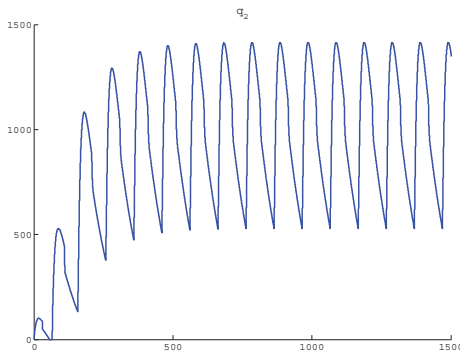

FIG 26. *Oscillating growth of the content $q_2(t)$ in the fluid model of the extreme example with abandonment.*

in automatically responding quickly to unexpected overloads, the examples in §3 and §4 show that it needs to be modified to recover rapidly after the overload is over, due to either a return to normal loading or a sudden change in the direction of the overload. We thus proposed the *fixed-queue-ratio with activation-and-release-thresholds* (FQR-ART) control. With FQR-ART, the one-way sharing rule is relaxed by adding the lower release thresholds. To avoid oscillations of the service process, which in turn can cause congestion collapse, we indicated that the activation thresholds also need to be increased, being asymptotically of order $O(n)$ as in (6) instead of $o(n)$, as

in (5) with FQR-T.

We then extended the fluid model developed in [36, 37, 38] based on the stochastic averaging principle to cover a more general time-varying environment. and developed the corresponding algorithm to numerically compute the performance functions in that fluid model. Simulation experiments indicate that this fluid model captures the main dynamics of the system, even in extreme cases, as the one considered in (7.3). Thus the fluid model can be used to ensure that the control parameters of FQR-ART are set properly.

There are many directions for future research. First, it remains to investigate the performance of FQR-ART in more complex time-varying scenarios. Second, it remains to establish theoretical properties of the new fluid model, paralleling [37]. Third, it remains to establish many-server heavy-traffic limits in this more general setting, paralleling [38, 39].

Fourth, and most important for engineering applications, it remains to extend the sharing mechanism to more than two systems. With more than two systems, there are more possible overload scenarios. One scenario involves only a single system experiencing an overload. For that scenario, assistance might be provided by several other systems, at less cost to each. The previous results for multi-class multi-pool systems in [18, 19, 20] indicate that such an extension should be possible. For example, one system might be judged to be overloaded, and assistance from others might be activated, perhaps with help only provided to the one system experiencing the overload (the analog of one-way sharing), if its queue length exceeds a specified proportion of the total queue length plus some activation threshold. Then sharing, with help only provided to the one overloaded system, might aim to keep the queue length of the overloaded system close to its target proportion. But then, as proposed here, evidently release thresholds should be used to ensure rapid recovery after the overload is over.

Acknowledgement

The authors received support from NSF grants CMMI 1066372 and 1265070.

REFERENCES

[1] Aksin, Z., M. Armony, V. Mehrotra. 2007. The modern call center: a multi-disciplinary perspective on operations management research. *Production Oper. Management*, **16** (6) 655–688.

[2] Armony, M., S. Israelit, A. Mandelbaum, Y. Marmor, Y. Tseytlin, G. Yom-tov. 2010. Patient flow in hospitals: a data-based queueing-science perspective. *Working paper*, New York University.

[3] Baier, V., R. Födisch, A. Ihring, E. Kessler, J. Lerchner, G. Wolf, J. M. Köhler, M. Nietzch, M. Krugel. 2006. Highly sensitive thermopile heat power sensor for microfluid calorimetry of biochemical processes. *Sensors and Actuators A* **123-124** (23) 354–359.

[4] Berger, A, W., W. Whitt. 1998. Effective bandwidths with priorities. *IEEE/ACM Transactions on Networking*, **6** (4), 447–460.

[5] Boyle, A., K. Beniuk, I. Higginson, P. Atkinson. 2012. Emergency department crowding: Time for interventions and policy evaluations. *Emergency Medicine J.* **29** 460–466.

[6] Choudhury G. L., K. K. Leung, W. Whitt. 1995. Efficiently providing multiple grades of service with protection against overloads in shared resources. *AT&T Technical Journal*, **74** (4), 50–63.

[7] Chan, C. W., M. Armony, N. Bambos. 2011. Fairness in overloaded parallel queues. *Working paper*, Columbia University, arXiv:1011.1237v2

[8] Chan, C. W., G. Yom-Tov, G. Escobar. 2011. When to use speedup: an examination of intensive care units with readmissions. *Working paper*, Columbia University.

[9] Deo, S., I. Gurvich. 2011. Centralized versus decentralized ambulance diversion: a network perspective. *Mangement Sci.* **57** (7), 1300–1319.

[10] Doshi, B., H. Heffes. 1986. Overload performance of several processor queueing disciplines for the M/M/1 queue. *IEEE Transactions on Communications*, **34** (6), 538–546.

[11] Erramilli, A., L. J. Forys. 1991. Oscillations and chaos in a flow model of a switching system. *IEEE journal on Selected Areas in Communications*, **9** (2), 171–178.

[12] Feinberg, E. A., M. I. Reiman. 1994. Optimality of randomized trunk reservation. *Prob Eng. Inf. Sci.* **8** (4) 463–489.

[13] Feldman, Z., Mandelbaum, A., Massey, W. A., Whitt, W. 2008. Staffing of time-varying queues to achieve time-stable performance. *Management Sci.* **54** (2), 324–338.

[14] Floyd, S., K. Fall. 1999. Promoting the use of end-to-end congestion control in the Internet *IEEE/ACM Transactions on Networking*, **7** (4), 458–472.

[15] Garnett, O., A. Mandelbaum, M. I. Reiman. 2002. Designing a call center with impatient customers. *Manufacturing Service Oper. Management*, **4** (3), 208–227.

[16] Goldstein, M. A., K. A. Kavajecz. 2004. Trading strategies during circuit breakers and extreme market movements. *Journal of Financial Markets*, **7** 301–333.

[17] Gurvich, I., O. Perry. 2012. Overflow networks: Approximations and implications to call-center outsourcing. *Oper. Res.*, **60** (4), 996–1009.

[18] Gurvich, I., W. Whitt. 2009a. Queue-and-idleness-ratio controls in many-server service systems. *Math. Oper. Res.* **34** (2) 363–396.

[19] Gurvich, I., W. Whitt. 2009b. Scheduling flexible servers with convex delay costs in many-server service systems. *Manufacturing Service Oper. Management* **11** (2) 237–253.

[20] Gurvich, I., W. Whitt. 2010. Service-level differentiation in many-server service systems via queue-ratio routing. *Oper. Res.* **58** (2) 316–328.

[21] Hampshire, R. C., O. B. Jennings, W. A. Massey. 2009. A time-varying call center design via Lagrangian mechanics. *Probability in the Engineering and Informational Sciences*, **23** (2), 231–259.

[22] Hampshire, R. C., W. A. Massey, Q. Wang. 2009. Dynamic pricing to control loss systems with quality of service targets. *Probability in the Engineering and Informational Sciences*, **23** (2), 357–383.

[23] Harrison J. M., A. Zeevi. 2005. A method for staffing large call centers using stochastic fluid models. *Manufacturing Service Oper. Management*, **7** (1), 20–36.

[24] Kelly, F. P. 1991. Loss networks. *Ann. Appl. Probab.* **1** (3), 319–378.

[25] Khalil, H. K. 2002. *Nonlinear Systems.* Prentice Hall, New Jersey.

[26] Koopman, B. O. 1972. Air-terminal queues under time-dependent conditions. *Operations Research*, **20** (6) 1089–1114.

[27] Körner U. 1991. Overload control of SPC systems. *International Teletraffic Congress, ITC 13*, Copenhagen, Denmark.

[28] Liu, Y., W. Whitt. 2011. Nearly periodic behavior in the the overloaded G/D/S+GI Queue. *Stochastic Systems* **1** (2) 340–410.

[29] Liu, Y., W. Whitt. 2012a. The $G_t/GI/s_t + GI$ many-server fluid queue. *Queueing Systems*, **71** (4) 405–444.

[30] Liu, Y., W. Whitt. 2012b. Stabilizing customer abandonment in many-server queues with time-varying arrivals. *Operations Research*, **60** (6) 1551–1564.

[31] Liu, Y., W. Whitt. 2012c. A many-server fluid limit for the $G_t/GI/s_t + GI$ queueing model experiencing periods of overloading. *Operations Research Letters*, **40** 307–312.

[32] Low, S. H., F. Paganini, J. C. Doyle. 2002. Internet congestion control. *Control Systems*, **22** (1), 28–43.

[33] Newell, G. F. 1982. *Applications of Queueing Theory*, Chapman Hall, London.

[34] Pang, G., Talreja, R., Whitt, W., 2007. Martingale proofs of many-server heavy-traffic limits for Markovian queues. *Probability Surveys*, **4**, 193–267.

[35] Perry, O., W. Whitt. 2009. Responding to unexpected overloads in large-scale service systems. *Management Sci.*, **55** (8), 1353–1367.

[36] Perry, O., W. Whitt. 2011a. A fluid approximation for service systems responding to unexpected overloads. *Oper. Res.*, **59** (5), 1159–1170.

[37] Perry, O., W. Whitt. 2011b. An ODE for an overloaded X model involving a stochastic averaging principle. *Stochastic Systems*, **1** (1), 17–66.

[38] Perry, O., W. Whitt. 2013a. A fluid limit for an overloaded X model via a stochastic averaging principle. *Math, Oper. Res.* **13** (2), 294-349.

[39] Perry, O., W. Whitt. 2013b. Diffusion approximation for an overloaded X model via a stochastic averaging principle. *Queueing Systems*, published online May 24, 2013.

[40] Perry, O., W. Whitt. 2014. The dynamics of a symmetric X fluid model with inefficient sharing. Paper in preparation.

[41] Powell, E.S., R. K. Khare, A. K. Venkatesh, B. D. Van Roo, J. G. Adams, G. Reinhardt. 2012. The relationship between inpatient discharge timing and emergency department boarding. *Journal of Emergency Medicine*, **42** (2), 186–196.

[42] Schulzrinne, H., J. F. Kurose, D. Towsley. 1990. Congestion control for real-time traffic in high-speed networks. *IEEE proceeding in Ninth Annual Joint Conference of the IEEE Computer and Communication Societies*, 543–550.

[43] Shah, D., D. Wischik. 2011. Fluid models of congestion collapse in overloaded switched networks. *Queueing Systems* **69** 121–143.

[44] Shi, P., M. Chou, J. G. Dai, D. Ding, J. Sim. 2012. Hospital Inpatient Operations: Mathematical Models and Managerial Insights. *Working paper*

[45] Sontag, E. D. 1998. *Mathematical Control Theory*, second edition, Springer, New York.

[46] Stolyar, A. L., T. Tezcan. 2011. Shadow-routing based control of flexible multiserver pools in overload. *Oper. Res.* **59** (6), 1427–1444.

[47] Teschl, G. 2009. *Ordinary Differential Equations and Dynamical Systems*, Universität Wien. Available online: www.mat.univie.ac.at/∼gerald/ftp/book-ode/ode.pdf

[48] Weber, J. H. 1964. A simulation study of routing control in communication networks. *Bell System Tech. J.* **43** 2639–2676.

[49] Wei, D. X., C. Jin, S. H. Low, S. Hegde. 2006. FAST TCP: motivation, architecture, algorithms, performance. *IEEE/ACM Transactions on Networking*, **14** (6), 1246–1259.

[50] Whitt, W. 2002. *Stochastic-Process Limits*, New York, Springer, 2002.

[51] Whitt, W. 2004. Efficiency-driven heavy-traffic approximations for many-server queues with abandonments. *Management Sci.* **50** (10) 1449–1461.

[52] Wilkinson, R. I. 1956. Theory for toll traffic engineering in the U.S.A. *Bell System Tech. J.* **35** 421–513.

[53] Yankovic, N., S. Glied, L. V. Green, M. Grams. 2010. The impact of ambulance diversion on heart attack deaths. *Inquiry* **47** (1) 81–91.

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208 E-mail: ohad.perry@northwestern.edu
Department of Industrial Engineering and Operations Research, Columbia University New York, NY, 10027 E-mail: ww2040@columbia.edu