Algorithms for Time-Varying Networks of Many-Server Fluid Queues

Yunan Liu
Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC, 27695, yunan_liu@ncsu.edu
Ward Whitt
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, ww2040@columbia.edu

Algorithms are developed to calculate the performance functions for a time-varying open network of many-server fluid queues. The deterministic fluid model has time-varying external arrival rate, service capacity and non-exponential abandonment at each queue, with proportional routing from one queue to another. The model serves as an approximation for the corresponding stochastic network of many-server queues with Markovian routing, experiencing periods of overloading at the queues. Simulation experiments are conducted to confirm that the algorithms are effective in computing the performance functions and that these performance functions provide useful approximations for the corresponding stochastic models.

Key words: queues with time-varying arrival rates; nonstationary queues; queueing networks; many-server queues; deterministic fluid models, fluid approximation; nonstationary networks of fluid queues; customer abandonment; non-Markovian queues.
History: Draft: February 16, 2012.

---

1. Introduction

In this paper we develop and study alternative algorithms to calculate the (exact, deterministic) time-dependent performance of a time-varying open network of many-server fluid queues, which we call a *fluid queue network* (FQNet). We consider three different algorithms for (generalizations of) the $(G_t/M/s_t + GI)^m/M_t$ FQNet, which has m fluid queues, each with time-varying external arrival rate (the G_t), a time-varying staffing function (the s_t) with unlimited waiting space, exponential service (the M) and abandonment from queue according to a general distribution (the $+GI$), plus time-varying proportional routing from one queue to another (the final M_t). The general patience (time-to-abandon) distribution

and service distribution (which appears in one algorithm) lead to considering two-parameter performance functions at each queue, such as $Q(t, y)$, the fluid content in queue at time t that has been so for at most time y, as a function of t and y.

These FQNets are intended to serve as approximations for corresponding *stochastic queueing networks* (SQNets), where the M_t routing becomes time-varying Markovian routing; a departure from queue i at time t goes (instantaneously) next to queue j with probability $P_{i,j}(t)$, independent of the system history up to that time. In the FQNet, a proportion $P_{i,j}(t)$ of the fluid flow out of queue i at time t goes next to queue j. In the SQNet, service times and patience times are random times for individual customers; in the FQNet, they specify flow proportions; i.e., with patience cdf F_i at queue i, $F_i(t)$ represents the proportion of all fluid that abandons by time t after it joins the queue, if it has not already entered service.

The broad goal of this work is to develop tools to analyze the performance of stochastic queues with time-varying arrival rates. Motivated by large-scale service systems, we focus on many-server queues with time-varying arrival rates; see Green et al. (2007). The motivation and theory for single many-server fluid queues has been given in Liu and Whitt (2011a-e). The algorithm for a single fluid queue exploits the assumption that the model alternates between successive *overloaded* (OL) and *underloaded* (UL) intervals. That work includes extensive comparisons with simulations of stochastic models and supporting heavy-traffic limit theorems. These fluid queues with alternating OL and UL intervals tend to be relevant when the stochastic system experiences such alternating periods of overloading and underloading. That behavior commonly occurs when it is too difficult or costly to dynamically adjust staffing in response to time-varying arrival rates to precisely balance supply and demand at all times. An alternative algorithm for a fluid queue based on a random-measure perspective that does not require alternating OL and UL intervals, but so far requires constant staffing (which can be applied more generally in a piecewise-constant manner), has been developed by Kang and Pang (2011).

Following Liu and Whitt (2011b), we now consider the more general FQNets. For the associated SQNets, there are few useful analysis tools besides discrete-event stochastic simulation. We envision the FQNets here being used in performance analysis together with simulation of associated SQNets. The FQNets can be analyzed much more rapidly, and so may be used efficiently in preliminary analyses, e.g., to efficiently derive candidate staffing functions at all queues. Then simulation of SQNets can be applied to verify and refine the FQNet analysis.

Among the limited literature on SQNets with time-varying arrival rates, an important contribution was made by Mandelbaum et al. (1998), who established many-server heavy-traffic limits for Markovian SQNets, showing that FQNets and associated diffusion process refinements arise in the many-server heavy-traffic limit, in which the arrival rate and staffing are both allowed to grow; see also Mandelbaum et al. (1999a-b). Detailed analysis can also be successfully performed for *infinite-server* (IS) SQNets, having infinitely many servers at each queue. Markovian IS SQNets were studied by Massey and Whitt (1993) while IS SQNets with time-varying phase-type (PH_t) distributions were studied by Nelson and Taaffe (2004a-b). They investigated $(PH_t/PH_t/\infty)^m$ SQNets with multiple customer classes and time-varying phase-type arrival and service processes. They showed that this IS network with k classes is mathematically equivalent to k single-class IS networks, each of which is furthermore equivalent to the $PH_t/PH_t/\infty$ IS model with a modified service distribution. They therefore directly applied the numerical algorithm they first developed for the $PH_t/PH_t/\infty$ model to the $(PH_t/PH_t/\infty)^m$ SQNets. Paralleling that analysis technique, we demonstrate how the algorithm for the single $G_t/GI/s_t+GI$ fluid queue in Liu and Whitt (2011a) can be applied to the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet.

The main contribution of the present paper is to develop new algorithms for FQNets and study them. The *first* of the three algorithms considered here, referred to as Alg(FPE), is the algorithm proposed in Liu and Whitt (2011b) for the more general $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNets, allowing time-dependent service rates and abandonment times; it combines a *fixed point equation* (FPE) for the vector of total arrival rates to the queues with the algorithm for a single fluid queue from Liu and Whitt (2011a-b). We implement and study that algorithm for the first time here.

The *second* algorithm, introduced here, is for that same FQNet and is referred to as Alg(ODE). It is based on an m-dimensional *ordinary differential equation* (ODE), where m is the number of queues. This new algorithm is appealing because it is relatively easy to implement. For the special case of $m = 2$ queues, we obtain an explicit analytical solution for the arrival rate functions, using the ODE representation; see Appendix E.1.

The *third* algorithm is a new FPE-based algorithm for $(G_t/GI/s_t + GI_t)^m/M_t$ FQNets, allowing non-exponential service-time distributions at each queue, which we consider for the first time here. We refer to it as Alg(FPE,GI). Thus, we find the solution to an important new model as well as study alternative algorithms. For applications, this generalization is important because non-exponential service distributions are seen in many service systems;

e.g., see Brown et al. (2005).

We evaluate the performance of these three algorithms by implementing them and conducting simulation experiments for associated SQNets for several examples. To relate the FQNets to associated SQNets, we use many-server heavy-traffic scaling, as in Liu and Whitt (2011d-e) and references therein. Thus, for a stochastic queue indexed by *scale parameter* n, we let the arrival rate be $n\lambda(t)$ and the number of servers be $\lceil n\, s(t) \rceil$, where $\lambda(t)$ and $s(t)$ are the fluid model counterparts and $\lceil x \rceil$ is the least integer greater than or equal to x.

We illustrate now with an example of a two-queue SQNet, as depicted in Figure 1 of Liu and Whitt (2011b); see §6.2 for details about this example. Figure 1 compares the fluid approximation (the dashed lines) with simulation estimates of the performance in the stochastic model (the solid lines) for $n = 4000$. We plot single sample paths of the following

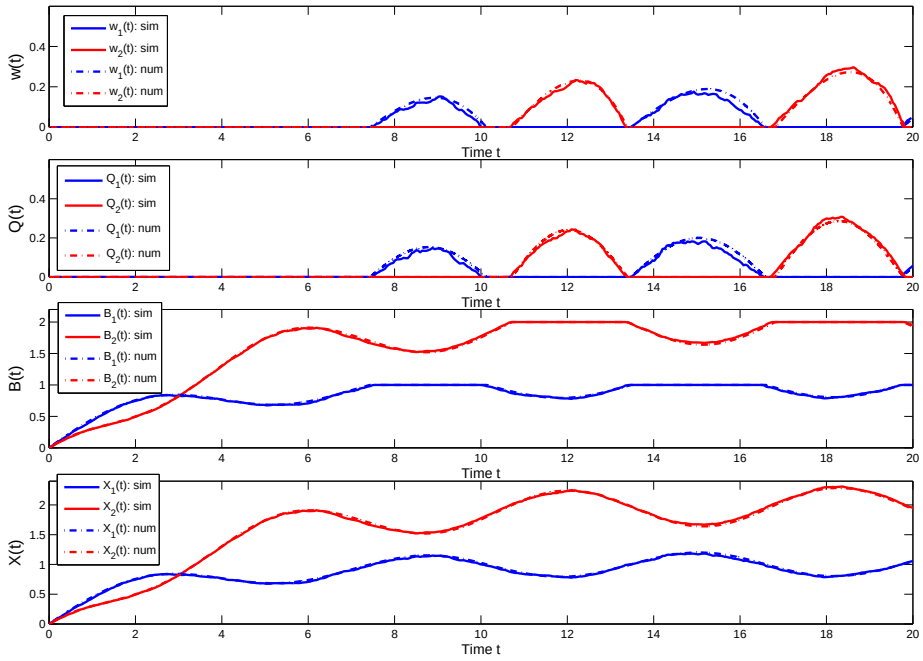

Figure 1: A comparison of performance functions in the two-queue FQNet with single sample paths from a simulation of the corresponding SQNet with scale parameter $n = 4000$.

processes: (i) the elapsed waiting time of the customer at the head of the line, $W_n(t)$, (ii) the scaled number of customers waiting in queue, $\bar{Q}_n(t) \equiv Q_n(t)/n$, (iii) the scaled number of customers in service, $\bar{B}_n(t) \equiv B_n(t)/n$, and (iv) the scaled total number of customers in the system $\bar{X}_n(t) \equiv X_n(t)/n$. For this extremely large value of n, there is little variability in the simulation sample paths. Figure 1 shows that each simulated sample path falls right on top of the the FQNet approximation. The close agreement confirms that both the numerical

algorithm and the simulation must be done correctly, and it empirically validates the many-server heavy-traffic limit.

For more realistic stochastic models with fewer servers, the fluid performance functions serve as approximations for the *mean values* of the corresponding stochastic processes. A figure nearly identical to Figure 1 (Figure 8 in the appendix) shows that the fluid model provides excellent approximations for the mean values for the same example with $n = 50$. Then the solid lines become simulation estimates of the mean of these scaled stochastic processes, obtained by averaging multiple independent sample paths.

Here is how the rest of this paper is organized. In §2 we review the definitions, assumptions and dynamics of the single $G_t/M_t/s_t + GI_t$ fluid queue discussed in Liu and Whitt (2011a-b). In §3 we review the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet and the results for it developed in Liu and Whitt (2011b). We also specify the first FPE-based algorithm Alg(FPE) there. In §4 we develop the alternative algorithm Alg(ODE) based on solving an m-dimensional ODE. In §5 we develop the new FPE-based algorithm Alg(FPE,GI) for the $(G_t/GI/s_t + GI_t)^m/M_t$ model with general service-times distributions at each queue. In §6 we demonstrate the performance of the algorithms by considering several examples. We also confirm conclusions drawn about the computational complexity there. Additional material appears in an appendix available online, including a discussion about checking for violation of staffing feasibility and the analytical solution for the arrival rates from the m-dimensional ODE for $m = 2$.

2. The $G_t/M_t/s_t + GI_t$ Fluid Queue

In this section we review the established results for the $G_t/M_t/s_t + GI_t$ fluid queue from Liu and Whitt (2011a-b); see those sources for more details.

2.1. Model Definition

There is a service facility with finite capacity and an associated waiting room or queue with unlimited capacity. Fluid is a deterministic, divisible and incompressible quantity that arrives over time. Fluid input flows directly into the service facility if there is free capacity available; otherwise it flows into the queue. Fluid leaves the queue and enters service in a first-come first-served (FCFS) manner whenever service capacity becomes available. There cannot be simultaneously free service capacity and positive queue content.

The staffing function (service capacity) s is an absolutely continuous positive function with

$$s(t) \equiv \int_0^t s'(y)\, dy, \quad t \geq 0. \tag{2.1}$$

We assume that the service capacity is exogenously specified and that it provides a hard constraint: the amount of fluid in service at time t cannot exceed $s(t)$. In general, there is no guarantee that some fluid that has entered service will not be later forced to leave without completing service, because we allow s to decrease. We directly assume that phenomenon does not occur; i.e., we directly assume that the given staffing function is *feasible*. However, Theorem 6 of Liu and Whitt (2011b) shows how to construct a minimum feasible staffing function greater than or equal to an initial infeasible staffing function. In the appendix we indicate how infeasibility can be detected whenever it occurs and corrected in each algorithm.

The total fluid input over an interval $[0, t]$ is $\Lambda(t)$, where Λ is an absolutely continuous function with

$$\Lambda(t) \equiv \int_0^t \lambda(y)\, dy, \quad t \geq 0, \tag{2.2}$$

where λ is the arrival-rate function. Service and abandonment occur deterministically in proportions. Since the service is M_t, the proportion of fluid in service at time t that will still be in service at time $t + x$ is

$$\bar{G}_t(x) = e^{-M(t,t+x)}, \quad \text{where} \quad M(t, t+x) \equiv \int_t^{t+x} \mu(y)\, dy, \tag{2.3}$$

for $t \geq 0$ and $x \geq 0$. The time-varying service-time cumulative distribution function (cdf) of a quantum of fluid that enters service at time t is $G_t \equiv 1 - \bar{G}_t(x)$; $\bar{G}_t(x)$ is the complementary cdf (ccdf). The cdf G_t has density $g_t(x) = \mu(t+x)\bar{G}_t(x)$ and hazard rate $h_{G_t}(x) = \mu(t+x)$, $x \geq 0$.

The model allows for abandonment of fluid waiting in the queue. In particular, a proportion $F_t(x)$ of any fluid to enter the queue at time t will abandon by time $t + x$ if it has not yet entered service, where F_t is an absolutely continuous cdf for each t, $-\infty < t < +\infty$, with

$$F_t(x) = \int_0^x f_t(y)\, dy, \quad x \geq 0, \quad \text{and} \quad \bar{F}_t(x) \equiv 1 - F_t(x), \quad x \geq 0. \tag{2.4}$$

Let $h_{F_t}(y) \equiv f_t(y)/\bar{F}_t(y)$ be the hazard rate associated with the patience (abandonment) cdf F_t. We assume that $f_t(y)$ is jointly measurable in t and y, so the same will be true for $F_t(y)$ and $h_{F_t}(y)$.

System performance is described by a pair of two-parameter deterministic functions $(\hat{B}, \hat{Q})$, where $\hat{B}(t,y)$ $(\hat{Q}(t,y))$ is the total quantity of fluid in service (in queue) at time t that has been so for a *duration* at most y, for $t \geq 0$ and $y \geq 0$. These functions will be absolutely continuous in the second parameter, so that

$$\hat{B}(t,y) \equiv \int_0^y b(t,x)\,dx \quad \text{and} \quad \hat{Q}(t,y) \equiv \int_0^y q(t,x)\,dx, \tag{2.5}$$

for $t \geq 0$ and $y \geq 0$. Performance is primarily characterized through the pair of two-parameter fluid content densities (b,q). Let $B(t) \equiv \hat{B}(t,\infty)$ and $Q(t) \equiv \hat{Q}(t,\infty)$ be the total fluid content in service and in queue, respectively. Let $X(t) \equiv B(t) + Q(t)$ be the total fluid content in the system at time t. Since service is assumed to be M_t, the performance will primarily depend on b via B. (We will not directly discuss $\hat{B}$.) The total service completion rate and abandonment rate at time t are

$$\sigma(t) \equiv \int_0^\infty b(t,x) h_{G_t}(x) dx = B(t)\mu(t), \quad t \geq 0, \tag{2.6}$$

$$\alpha(t) \equiv \int_0^\infty b(t,x) h_{F_t}(x) dx, \tag{2.7}$$

respectively. Let $S(t)$ be the total amount of fluid to complete service in the interval $[0,t]$; then

$$S(t) \equiv \int_0^t \sigma(y)\,dy = \int_0^t B(y)\mu(y)\,dy, \quad t \geq 0. \tag{2.8}$$

Since fluid in service (queue) that is not served (does not abandon or enter service) remains in service (queue), we see that the fluid content densities b and q must satisfy the equations

$$b(t+u, x+u) = b(t,x)\frac{\bar{G}_{t-x}(x+u)}{\bar{G}_{t-x}(x)} = b(t,x)e^{-M(t,t+u)}, \tag{2.9}$$

$$q(t+u, x+u) = q(t,x)\frac{\bar{F}_{t-x}(x+u)}{\bar{F}_{t-x}(x)}, \quad 0 \leq x+u < w(t), \tag{2.10}$$

for $t \geq 0$, $x \geq 0$ and $u \geq 0$, where M is defined in (2.3) and $w(t)$ is the *boundary waiting time* (BWT) at time t,

$$w(t) \equiv \inf\{x > 0 : q(t,y) = 0 \quad \text{for all} \quad y > x\}. \tag{2.11}$$

(By Assumptions 4 and 5 below, we are never dividing by 0 in (2.9) and (2.10).) Since the service discipline is FCFS, fluid leaves the queue to enter service from the right boundary of $q(t,x)$.

Let $A(t)$ be the total amount of fluid to abandon in the interval $[0,t]$ and Let $E(t)$ be the amount of fluid to enter service in $[0,t]$. Clearly, we have the *flow conservation equations*: For each $t \geq 0$,

$$Q(t) = Q(0) + \Lambda(t) - A(t) - E(t) \quad \text{and} \quad B(t) = B(0) + E(t) - S(t). \tag{2.12}$$

The abandonment satisfies

$$A(t) \equiv \int_0^t \alpha(y)\, dy, \quad \alpha(t) \equiv \int_0^\infty q(t,y) h_{F_{t-y}}(y)\, dy \tag{2.13}$$

for $t \geq 0$, where $\alpha(t)$ is the abandonment rate at time t and $h_{F_t}(y)$ is the hazard rate associated with the patience cdf F_t. (Recall that F_t is defined for t extending into the past.) The flow into service satisfies

$$E(t) \equiv \int_0^t b(u,0)\, du, \quad t \geq 0, \tag{2.14}$$

where $b(t,0)$ is the rate fluid enters service at time t. If the system is OL, then the fluid to enter service is determined by the *rate that service capacity becomes available* at time t,

$$\eta(t) \equiv s'(t) + \sigma(t) = s'(t) + B(t)\mu(t), \quad t \geq 0, \tag{2.15}$$

Then $\eta(t)$ coincides with the *maximum possible rate that fluid can enter service* at time t,

$$\gamma(t) \equiv s'(t) + s(t)\mu(t). \tag{2.16}$$

To describe waiting times, let the BWT $w(t)$ be the delay experienced by the quantum of fluid at the head of the queue at time t, already given in (2.11), and let the *potential waiting time* (PWT) $v(t)$ be the virtual delay of a quantum of fluid arriving at time t under the assumption that the quantum has infinite patience. A proper definition of q, w and v is somewhat complicated, because w depends on q, while q depends on w, but that has been done in §7 in Liu and Whitt (2011a).

We specify the initial conditions via the initial fluid densities $b(0,x)$ and $q(0,x)$, $x \geq 0$. Then $\hat{B}(0,y)$ and $\hat{Q}(0,y)$ are defined via (2.5), while $B(0) \equiv \hat{B}(0,\infty)$ and $Q(0) \equiv \hat{Q}(0,\infty)$, as before. Let $w(0)$ be defined in terms of $q(0,\cdot)$ as in (2.11). In summary, the six-tuple $(\lambda(t), s(t), \mu(t), F_t(x), b(0,x), q(0,x))$ of functions of the variables t and x specifies the *model data*. The system performance is characterized by $(b(t,x), q(t,x), w(t), v(t), \alpha(t), \sigma(t))$.

2.2. Assumptions on the Model Data

We directly assume that the initial values are finite:

Assumption 1 (*finite initial content*) $B(0) < \infty$, $Q(0) < \infty$ *and* $w(0) < \infty$.

As in Liu and Whitt (2011a-b), we consider a *smooth model*. Let $\mathbb{C}_p$ be the space of piecewise continuous real-valued functions of a real variable, by which we mean that there are only finitely many discontinuities in each finite interval, and that left and right limits exist at each discontinuity point, where the whole function is right continuous. Thus, $\mathbb{C}_p$ is a subset of $\mathbb{D}$, the right-continuous functions with left limits.

Assumption 2 (*smoothness*) $s', \lambda, f_t, f_\cdot(x), \mu, b(0,\cdot), q(0,\cdot)$ *in* $\mathbb{C}_p$ *for each* $x \geq 0$ *and* t, $-\infty < t < \infty$.

To treat the BWT w, we need to impose a regularity condition on the arrival rate function and the initial queue density, as in Assumption 10 of Liu and Whitt (2011a). Here and later we use the notation $\uparrow$ and $\downarrow$ to denote supremum and infimum, respectively, e.g.,

$$\lambda_t^\uparrow \equiv \sup_{0\leq u\leq t}\{\lambda(u)\} \quad \text{and} \quad \lambda_t^\downarrow \equiv \inf_{0\leq u\leq t}\{\lambda(u)\}. \tag{2.17}$$

These apply in the obvious way, e.g., $q^\downarrow(0,x)$ below denotes the infimum over the second variable over $[0,x]$ and $\lambda_\infty^\uparrow$ denotes the supremum over the positive halfline.

Assumption 3 (*positive arrival rate and initial queue density*) *For all* $t \geq 0$, $\lambda_t^\downarrow > 0$ *and* $q^\downarrow(0, w(0)) > 0$ *if* $w(0) > 0$.

Appendix E of Liu and Whitt (2011a) illustrates the more complicated behavior that can occur for the BWT w when $\lambda_t^\downarrow = 0$.

To ensure that the PWT v is finite, we assume bounds on the minimum staffing level and the minimum service rate, as in Assumptions 7 and 8 of Liu and Whitt (2011b).

Assumption 4 (*minimum staffing and service rate*) $s_\infty^\downarrow > 0$ *and* $\mu_\infty^\downarrow > 0$.

To treat the time-varying abandonment cdf F_t, we introduce bounds for the time-varying pdf f_t and complementary cdf $\bar{F}_t$, as in Liu and Whitt (2011b). Let

$$f^\uparrow \equiv \sup_{-\infty<t<\infty, x\geq 0}\{f_t(x)\} \quad \text{and} \quad \bar{F}^\downarrow(x) \equiv \inf_{-\infty\leq t<\infty}\{\bar{F}_t(x)\}. \tag{2.18}$$

Assumption 5 (*controlling the time-varying abandonment*) $f^{\uparrow} < \infty$ *and* $\bar{F}^{\downarrow}(x) > 0$ *for all* $x > 0$, *where* $f^{\uparrow}$ *and* $\bar{F}^{\downarrow}(x)$ *are defined in* (2.18).

We analyze the fluid queue under the assumptions above by considering alternating intervals over which the system is either UL or OL, where these intervals include what is usually regarded as critically loaded. In particular, an interval starting at time t_0 with (i) $Q(t_0) > 0$ or (ii) $Q(t_0) = 0$, $B(t_0) = s(t_0)$ and $\lambda(t_0) > s'(t_0) + \sigma(t_0)$ is OL. We thus use $\mathcal{R}$ to denote the current system regime and let $\mathcal{R}(t_0) \equiv$ OL. The OL interval ends at the *OL termination time*

$$T_{OL}(t_0) \equiv \inf\{u \geq t_0 : Q(u) = 0 \quad \text{and} \quad \lambda(u) \leq s'(u) + \sigma(u)\}. \tag{2.19}$$

Case (ii) in which $Q(t_0) = 0$ and $B(t_0) = s(t_0)$ is often regarded as critically loaded, but because the arrival rate $\lambda(0)$ exceeds the rate that new service capacity becomes available, $s'(t_0) + \sigma(t_0)$, we must have the right limit $Q(t_0+) > 0$, so that there exists $\epsilon > 0$ such that $Q(u) > 0$ for all $u \in (0, 0 + \epsilon)$. Hence, we necessarily have $T_{OL}(t_0) > 0$.

An interval starting at time t_0 with (i) $Q(t_0) < 0$ or (ii) $Q(t_0) = 0$, $B(t_0) = s(t_0)$ and $\lambda(t_0) \leq s'(t_0) + \sigma(t_0)$ is UL, designated by $\mathcal{R}(t_0) = UL$. The UL interval ends at *UL termination time*

$$T_{UL}(t_0) \equiv \inf\{u \geq t_0 : B(u) = s(u) \quad \text{and} \quad \lambda(u) > s'(u) + \sigma(u)\}. \tag{2.20}$$

As before, case (ii) in which $Q(t_0) = 0$ and $B(t_0) = s(0)$ is often regarded as critically loaded, but because the arrival rate $\lambda(t_0)$ does not exceed the rate that new service capacity becomes available, $\eta(t_0) \equiv s'(t_0) + \sigma(t_0)$, we must have the right limit $Q(t_0+) = 0$. The UL interval may contain subintervals that are conventionally regarded as critically loaded; i.e., we may have $Q(t) = 0$, $B(t) = s(t)$ and $\lambda(t) = s'(t) + \sigma(t)$. For the fluid models, such critically loaded subintervals can be treated the same as UL subintervals. However, unlike an overloaded interval, we cannot conclude that we necessarily have $T_{UL}(t_0) > 0$ for a UL interval. Moreover, even if $T_{UL}(t_0) > 0$ for each UL interval, we could have infinitely many switches between OL intervals and UL intervals in a finite interval. Thus we make assumptions to ensure that those pathological situations do not occur. We define the interval termination time

$$T_{\mathcal{R}}(t_0) \equiv T_{OL}(t_0)\, \mathbf{1}_{\{\mathcal{R}(t_0)=\text{OL}\}} + T_{UL}(t_0)\, \mathbf{1}_{\{\mathcal{R}(t_0)=\text{UL}\}}. \tag{2.21}$$

As discussed in Liu and Whitt (2011a), for engineering applications it is reasonable to directly assume that there are only finitely many switches between OL and UL intervals

in each finite time interval, but it is unappealing mathematically. In §3 of Liu and Whitt (2011b) we provided sufficient conditions based directly on the model parameters for there to be only finitely many switches between OL intervals and UL intervals in each finite time interval. In particular, we showed that it suffices to impose regularity conditions on the function $\zeta(t) \equiv \lambda(t) - s'(t) - s(t)\mu(t)$, $t \geq 0$. Let $Z_{\zeta,T}$ be the subset of zeros of the function ζ in $[0,T]$ and let $|A|$ be the cardinality of a set A. Theorem 2 of Liu and Whitt (2011b) shows that the number of switches between overloaded and underloaded intervals is finite in each finite interval if $|Z_{\zeta,T}| < \infty$ for each $T > 0$.

Assumption 6 (*controlling the number of switches*) *For all* $T > 0$, $|Z_{\zeta,T}| < \infty$.

In §3 of Liu and Whitt (2011b) we also showed that a sufficient condition for $|Z_{\zeta,T}| < \infty$ for each $T > 0$ is for the functions λ, s and μ to be piecewise polynomials (with finitely many discontinuities in each finite interval). Assumption 6 is also easy to verify in other settings, as we illustrate here with sinusoidal functions. *We assume that all assumptions in this section are in force throughout the paper.*

2.3. The Performance Formulas

In Liu and Whitt (2011a-b) we showed how the system performance expressed via the basic functions $\hat{\mathcal{P}}(t) \equiv (b(t,\cdot), q(t,\cdot))$ depends on the model data $\mathcal{D} \equiv \left(\lambda, s, \mu, F, \hat{\mathcal{P}}(0)\right)$. From the basic performance vector $\hat{\mathcal{P}}$, we easily compute the associated vector of all performance functions

$$\mathcal{P}(t) \equiv \left(\hat{\mathcal{P}}(t), w(t), v(t), B(t), Q(t), X(t), \sigma(t), S(t), \alpha(t), A(t), E(t)\right) \tag{2.22}$$

via the definitions in §2.1. We quickly review the main results for the basic functions (b, q, w, v); see Liu and Whitt (2011a-b) for more details.

For the fluid model with unlimited service capacity ($s(t) \equiv \infty$ for all $t \geq 0$), starting at time 0,

$$\begin{aligned} b(t,x) &= e^{-M(t-x,t)}\lambda(t-x)1_{\{x \leq t\}} + e^{-M(0,t)}b(0,x-t)1_{\{x>t\}}, \\ B(t) &= \int_0^t e^{-M(t-x,t)}\lambda(t-x)\,dx + B(0)e^{-M(0,t)}, \quad t \geq 0. \end{aligned} \tag{2.23}$$

where M is defined in (2.3). If, instead, a finite-capacity system starts UL, then the same formulas apply over the interval $[0,T)$, where $T \equiv \inf\{t \geq 0 : B(t) > s(t)\}$, with $T = \infty$ if the infimum is never obtained.

For the fluid model in an OL interval, $B(t) = s(t)$ and

$$\begin{aligned} b(t,x) &= (s'(t-x) + s(t-x)\mu(t-x))e^{-M(t-x,t)}1_{\{x\le t\}} \\ &\quad + b(0,x-t)e^{-M(0,t)}1_{\{x>t\}}. \end{aligned} \tag{2.24}$$

Let $\tilde{q}(t,x)$ be $q(t,x)$ during an OL interval $[0,T]$ under the assumption that no fluid enters service from queue. During an OL interval,

$$\tilde{q}(t,x) = \lambda(t-x)\bar{F}_{t-x}(x)1_{\{x\le t\}} + q(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\}}; \tag{2.25}$$

$$\begin{aligned} q(t,x) &= \tilde{q}(t-x,0)\bar{F}_{t-x}(x)1_{\{x\le w(t)\wedge t\}} + \tilde{q}(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\le w(t)\}} \\ &= \lambda(t-x)\bar{F}_{t-x}(x)1_{\{x\le w(t)\wedge t\}} + q(0,x-t)\frac{\bar{F}_{t-x}(x)}{\bar{F}_{t-x}(x-t)}1_{\{t<x\le w(t)\}}. \end{aligned}$$

We characterize the BWT w appearing in the formula for q above by equating the quantity of new fluid admitted into service in the interval $[t, t+\delta)$ to the amount of fluid removed from the right boundary of $q(t,x)$ that does not abandon in the same interval $[t, t+\delta)$. By careful analysis (Theorem 3 of Liu and Whitt (2011a)), that leads to the nonlinear first-order ODE

$$w'(t) = \Omega(t, w(t)) \equiv 1 - \frac{\gamma(t)}{\tilde{q}(t, w(t))}, \tag{2.26}$$

for γ in (2.16), where $w'(t)$ denotes the derivative. (By Assumptions 3, 4 and 5, we are not dividing by 0 in (2.25) and (2.26). More detail on the structure of w is given in Liu and Whitt (2011a). Overall, w is continuously differentiable everywhere except for finitely many t.) We compute the end of an OL interval by letting it be the first time t that $w(t) = 0$ and $\lambda(t) \le s'(t) + s(t)\mu(t)$. During an OL interval, the PWT v is finite and is the unique function in $\mathbb{D}$ satisfying the equation

$$v(t - w(t)) = w(t) \quad \text{for all} \quad t \ge 0. \tag{2.27}$$

2.4. The Fluid Algorithm for Single Queues (FASQ)

The results above yield an efficient algorithm to compute the basic performance four tuple (b, q, w, v) over a finite interval $[0, T]$. First, for each UL interval, we compute b directly via (2.23), terminating the first time we obtain $B(t) > s(t)$. Second, for each OL interval, we compute b via (2.24), $\tilde{q}$ via (2.25) and then the BWT w by solving the ODE (2.26). We consider terminating the OL interval when $w(t) = 0$. We actually do terminate the OL

interval if also $\lambda(t) \leq s'(t) + s(t)\mu(t)$. The proof of Theorem 5 in Liu and Whitt (2011a) provides an elementary algorithm to compute v during an OL interval from (2.27) once w has been computed. Theorem 6 of Liu and Whitt (2011a) shows that v satisfies its own ODE under additional regularity conditions.

The key step beyond direct computation is to control the switching between UL and OL intervals. That can be done by selecting a fixed *switching step size* ΔT over which to perform all calculations before checking to see if there is a regime change. Starting at time t in regime $\mathcal{R}(t)$, the calculations are performed over the interval $[t, t + \Delta T]$. Then the algorithm finds the first time s in $(t, t + \Delta T]$ at which there is a regime change, if any, and that becomes the new initial time t. If the switching step size ΔT is too large, then there can be much wasted computation. Otherwise, the algorithm tends to be insensitive to the choice of ΔT, as we show in Appendix C.

A formal statement of the single-queue algorithm appears in Appendix C. For a time interval $[0, T]$ with $\mathcal{S}$ regime switches, examples show that the running time of FASQ tends to be linear in both T, for fixed $\mathcal{S}$, and $\mathcal{S}$, for fixed T, and independent of ΔT, provided that ΔT is suitably small, e.g., if $\Delta T \leq T/\mathcal{S}$, assuming that the switching points are approximately uniformly distributed throughout the interval $[0, T]$. Thus, for a fixed density of switches per time, the run time should be $O(T^2)$, because $\mathcal{S}$ would be proportional to T. These observations are illustrated by a numerical example in Appendix C.

3. The $(G_t/M_t/s_t + GI_t)^m/M_t$ Fluid Network

We now review the $(G_t/M_t/s_t + GI_t)^m/M_t$ FQNet introduced by Liu and Whitt (2011b) and the FPE-based algorithm to compute all transient performance functions proposed there.

3.1. Model Properties

There are m queues, where each queue has model parameters as given in §2. In addition, a proportion $P_{i,j}(t)$ of the fluid output from queue i at time t is routed immediately to queue j, and a proportion $P_{i,0}(t) \equiv 1 - \sum_{j=1}^{m} P_{i,j}(t) \leq 1$ is routed out of the network (departs having successfully completed all required service). Consistent with the terminology, we assume that $P(t)$ is sub-stochastic for each t.

Assumption 7 (*proportional routing*) *The routing matrix function for proportional routing,* $P : [0, \infty) \to \mathbb{R}^{m^2}$, *is in* $\mathbb{C}_p$ *and* $\sum_{j=1}^{m} P_{i,j}(t) \leq 1$ *for each* $t \geq 0$ *and* i, $1 \leq i \leq m$.

Since we have a deterministic fluid model, it is elementary to treat the basic network operations of superposition and splitting: If two input streams are combined to form a single input (superposition), then the arrival rate functions are simply added. If one stream with arrival rate function λ is split, such that a proportion $p(t)$ of that stream goes into a new split stream at time t, then the arrival-rate function of the split stream is λ_p, where $\lambda_p(t) \equiv \lambda(t)p(t)$, $t \geq 0$; just like λ, the splitting proportion can be time-dependent. Similarly, if the departure flow from one queue becomes input to another, then the resulting arrival-rate function is σ; (We do not let the abandonment flow from one queue become input to another, but if we did, then the resulting arrival-rate function would be α.) However, converting departure rate or abandonment rate into new input rate is more complicated when feedback is allowed. We discuss that case now, for departures only.

As is usual with open queueing networks, there is an external exogenous arrival rate function to each queue (from outside the network, which could be null at some queues) and there is a total arrival rate function to each queue (which we simply call the arrival rate function), taking into account the flow from other queues. Let the external arrival rate function into queue j be denoted by $\lambda_j^{(0)}$; let the arrival rate function into queue j be denoted by λ_j. The model data for the $G_t/M_t/s_t + GI_t$ fluid queues directly provides the external arrival rate functions $\lambda_j^{(0)}$ (with the superscript 0 now added), while the arrival rate function itself satisfies a system of *traffic rate equations*. In particular,

$$\lambda_j(t) = \lambda_j^{(0)}(t) + \sum_{i=1}^{m} \sigma_i(t) P_{i,j}(t), \tag{3.1}$$

where

$$\sigma_i(t) = B_i(t)\mu_i(t), \quad t \geq 0. \tag{3.2}$$

Equations (3.1) and (3.2) produce a system of equations, with λ_j depending upon σ_i for $1 \leq i \leq m$, while σ_i in turn depends on λ_i for each i, because B_i depends on λ_i. The formulas for B_i as a function of λ_i have been given in 2.3, provided that we know the current system status, i.e., whether the queue is OL or UL. That requirement is the major source of complexity.

Since (3.1) is a linear equation, it can be written in matrix notation as $\lambda = \lambda^{(0)} + \sigma\, P$ by omitting the argument t as below, provided that the product σP is interpreted as in (3.1). Moreover, we can combine (3.1) and (3.2) to express λ as the solution of a fixed point equation mapping $\mathbb{C}_p^m$ over $[0, T]$ into itself. To see this, note that $B_i(t)$ in (3.2) is

a function of $\lambda_i(u)$, $0 \le u < t$, and the model data (only needed for queue i). Hence the vector $B(t) \equiv B_1(t), \ldots, B_m(t)$ is a function of λ over $[0,t)$ and the model data. Hence we can express (3.1) and (3.2) abstractly as $\lambda = \Psi(\lambda)$, where $\Psi(x)(t)$ depends on its argument x only over $[0,t]$ for each $t \ge 0$. Here the function Ψ depends on all the model data $(\lambda_i^{(0)}, s_i, \mu_i, F_{i,\cdot}, b_i(0,\cdot), q_i(0,\cdot), P)$, $1 \le i \le m$.

Assumption 8 (*finitely many switches between different regimes*) *There are finite many switches between OL and UL intervals in each finite interval* $[0,T]$ *for every queue* i *of the* $(G_t/M_t/s_t)^m + GI_t$ *fluid network.*

Under Assumption 8, the operator Ψ above is a monotone contraction operator on the m-dimensional product space $\mathbb{C}_p^m$, by Theorem 10 of Liu and Whitt (2011b). Therefore, we can approach this system recursively. If we do so with initial vector $\tilde{\lambda} = \lambda^{(0)}$, the vector of external arrival rate functions, then the recursion has an important practical interpretation. Then the k^{th} iterate $\lambda_j^{(k)}$ is the arrival rate of fluid that has previously experienced k transitions in the fluid network. With this notation, we can write the recursive formulas

$$\lambda_j^{(n)}(t) = \Psi^{(n)}(\lambda^{(0)})_j(t) = \lambda_j^{(0)}(t) + \sum_{i=1}^{m} \sigma_i^{(n-1)}(t) P_{i,j}(t), \quad n \ge 1, \tag{3.3}$$

where

$$\sigma_i^{(n)}(t) = B_i^{(n)}(t)\mu_i(t) \quad n \ge 0. \tag{3.4}$$

Since we necessarily have $\lambda_i^{(1)} \ge \lambda_i^{(0)}$ for each i, this recursion converges monotonically to the fixed point λ.

3.2. The FPE-Based Algorithm Alg(FPE)

The algorithm Alg(FPE) consists of two successive steps: (i) solving the traffic-rate equations (3.1) and (3.2) and (ii) solving for the performance vector $(b, q, w, v, \sigma, \alpha)$ at each queue using the algorithm in §2. For step (i), we start with an initial vector of arrival rate functions, which can a rough estimate of the final arrival rate functions or the given external arrival rate functions. We then apply the performance formulas in §2.3 to determine the performance functions B_i and σ_i at each queue to determine a new vector of arrival rate functions. We then iteratively calculate successive vectors of arrival rate functions until the difference (measured in the supremum norm over a bounded interval) is suitably small. Then we apply step (ii).

Given a desired duration T of an interval $[0, T]$, we specify the following input data: (i) the model parameter vector

$$\left(\lambda^{(0)}, s, G, F, \mathcal{P}(0)\right) \equiv \left(\lambda_i^{(0)}(t), s_i(t), G_i, F_i, \mathcal{P}_i(0), 1 \le i \le m,\, t \in [0, T]\right), \tag{3.5}$$

where the initial performance vector (at time 0) of queue i, $1 \le i \le m$, is

$$\mathcal{P}_i(0) \equiv (b_i(0, \cdot), q_i(0, \cdot), B_i(0), Q_i(0), w_i(0), v_i(0), \alpha_i(0), \sigma_i(0))\,;$$

and (ii) the algorithm parameters: the iteration *error tolerance parameter* (ETP) ϵ and the *switching step size* (SSS) ΔT, both assumed to be strictly positive. (We assume that the switching step size is the same for all queues, which usually provides little loss of generality.) We give a formal statement of the algorithm in the appendix.

From the structure of the algorithm, we can directly determine the *computational complexity* (computer-dependent required run time) $\mathcal{C}_{FPE} \equiv \mathcal{C}_{FPE}(\epsilon, T, m, \mathcal{S})$ as a function of the ETP ϵ, number of queues m, length of the time interval T, and the number of regime switches per queue $\mathcal{S}$, but we will also confirm it in numerical examples.

Let $\mathcal{I} \equiv \mathcal{I}(\epsilon)$ be the number of *iterations* of the FPE as a function of the ETP ϵ. Roughly, we need to apply the FASQ for each of the m queues $\mathcal{I}$ times, although the full FASQ is not needed in the steps before the final one needed to compute the actual performance functions at each queue. Let $\mathcal{S}_i$ be the number of regime switches at queue i over $[0, T]$. Thus the overall complexity should be $\mathcal{C}_{FPE} = O(\mathcal{I}T \sum_{i=1}^{m} S_i)$. Assuming that $\mathcal{S}_i \approx \mathcal{S}$ for all i, with the switches at different queues occurring at different times, that yields $\mathcal{C}_{FPE}(\mathcal{I}, m, T, \mathcal{S}) = O(\mathcal{I}Tm\mathcal{S})$. Moreover, $\mathcal{I}(\epsilon) = O(\log(1/\epsilon)$ where ϵ is the ETP, because the convergence to the fixed point in successive iterations is geometrically fast. Thus, we conclude that

$$\mathcal{C}_{FPE} \equiv \mathcal{C}_{FPE}(m, T, \mathcal{S}, \epsilon) = O(mT\mathcal{S}\log(1/\epsilon)). \tag{3.6}$$

Unfortunately, unlike the other parameters, the number of regime switches per queue, $\mathcal{S}$, cannot be directly observed from the model data. However, if the model parameters, such as λ and s, are periodic functions with periods τ_λ and τ_s, then the total number of switchings is usually bounded by $2T/\tau_\lambda + 2T/\tau_s$, so that we may regard $\mathcal{S} = O(T)$, making $\mathcal{C}_{FPE}(m, T, \epsilon) = O(mT^2 \log(\epsilon))$. See the examples in §6.

4. The Alternative ODE-Based Algorithm Alg(ODE)

Now we develop the new algorithm Alg(ODE) for the $(G_t/M_t/s_t+GI_t)^m/M_t$ FQNet. Again, the key is to compute *total arrival rates* (TARs) for all queues and then treat each of these queues independently. In some special cases, analytic formulas are available.

4.1. Finding the Total Arrival Rate (TAR) Vector

Instead of solving the fixed-point equation as in §3 to find the TARs, we now solve an m-dimensional ODE. To do that, we need to work over subintervals where all queues are in specified regimes. So now we consider successive switching times for *any* queue in the network. We recursively solve the ODE in each of these intervals. The key is to characterize and update the system regime in different intervals and recursively advance in t. We describe the system regime at t with two sets: $\mathcal{U}(t)$ (the set of indices of queues that are UL) and $\mathcal{O}(t)$ (the set of indices of queues that are OL). In other words,

$$\begin{aligned}\mathcal{U}(t) &\equiv \{1 \le i \le m : B_i(t) \le s_i(t),\, Q_i(t) = 0\}\\ \mathcal{O}(t) &\equiv \{1 \le i \le m : B_i(t) = s_i(t),\, Q_i(t) > 0\}.\end{aligned}$$

Of course, $\mathcal{U}(t)$ is simply the complement of $\mathcal{O}(t)$ within the set $\{1,\ldots,m\}$.

Given $\mathcal{U}(t)$ and $\mathcal{O}(t)$, consider $1 \le i \le m$. (i) If Queue i is UL, i.e., $i \in \mathcal{U}(t)$, flow conservation implies that

$$B_i'(t) = \lambda_i^{(0)}(t) + \sum_{j\in\mathcal{U}(t)} \mu_j(t)P_{j,i}(t)B_j(t) + \sum_{k\in\mathcal{O}(t)} \mu_k(t)P_{k,i}(t)s_k(t) - \mu_i(t)B_i(t).$$

If $i \in \mathcal{O}(t)$, $B_i(t) = s_i(t)$. We partition and regroup the indices of queues so that $\mathbf{B}(t) \equiv [\mathbf{B}_\mathcal{U}(t), \mathbf{B}_\mathcal{O}(t)]^T$, $\lambda(t) \equiv [\lambda_\mathcal{U}(t), \lambda_\mathcal{O}(t)]^T$, $\lambda^{(0)}(t) \equiv [\lambda_\mathcal{U}^{(0)}(t), \lambda_\mathcal{O}^{(0)}(t)]^T$, $\mu(t) \equiv [\mu_\mathcal{U}(t), \mu_\mathcal{O}(t)]^T$, $\mathbf{s}(t) \equiv [\mathbf{s}_\mathcal{U}(t), \mathbf{s}_\mathcal{O}(t)]^T$, $\mathbf{\Gamma}_\mathcal{U}(t) \equiv diag(\mu_\mathcal{U}(t))$, $\mathbf{\Gamma}_\mathcal{O}(t) \equiv diag(\mu_\mathcal{O}(t))$, $\mathbf{\Gamma}(t) \equiv diag(\mathbf{\Gamma}_\mathcal{U}(t), \mathbf{\Gamma}_\mathcal{O}(t))$,

$$\mathbf{P}(t) \equiv \begin{array}{c} \\ \mathcal{U} \\ \mathcal{O} \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{U} & \mathcal{O} \end{array} \\ \left[\begin{array}{c|c} \mathbf{P}_{\mathcal{U}\mathcal{U}}(t) & \mathbf{P}_{\mathcal{U}\mathcal{O}}(t) \\ \hline \mathbf{P}_{\mathcal{O}\mathcal{U}}(t) & \mathbf{P}_{\mathcal{O}\mathcal{O}}(t) \end{array}\right] \end{array},$$

where $\mathbf{P}_{\mathcal{U}\mathcal{U}}(t)$ ($\mathbf{P}_{\mathcal{O}\mathcal{U}}(t)$, $\mathbf{P}_{\mathcal{U}\mathcal{O}}(t)$, and $\mathbf{P}_{\mathcal{O}\mathcal{O}}(t)$) denotes the transition probability from a state in $\mathcal{U}$ ($\mathcal{O}, \mathcal{U}$, and $\mathcal{O}$) to a state in $\mathcal{U}$ ($\mathcal{U}$, $\mathcal{O}$, and $\mathcal{O}$) at time t. Let $\mathbf{P}_{\mathcal{O}\mathcal{U}}(t) = \mathbf{P}_{\mathcal{U}\mathcal{O}}(t) = \mathbf{P}_{\mathcal{O}\mathcal{O}}(t) = \phi$ when $\mathbf{P}_{\mathcal{U}\mathcal{U}}(t) = \mathbf{P}(t)$ (i.e., all queues are UL) and let $\mathbf{P}_{\mathcal{O}\mathcal{U}}(t) = \mathbf{P}_{\mathcal{U}\mathcal{O}}(t) = \mathbf{P}_{\mathcal{U}\mathcal{U}}(t) = \phi$ when $\mathbf{P}_{\mathcal{O}\mathcal{O}}(t) = \mathbf{P}(t)$ (i.e., all queues are OL), where ϕ denotes an empty matrix (with rank 0).

Therefore, in matrix notation, we have

$$\mathbf{B}'_{\mathcal{U}}(t) \;=\; \mathbf{C}(t)\cdot\mathbf{B}_{\mathcal{U}}(t)+\mathbf{D}(t), \quad \text{and} \quad \mathbf{B}_{\mathcal{O}}(t)=\mathbf{s}_{\mathcal{O}}(t), \tag{4.1}$$

where

$$\begin{aligned}\mathbf{D}(t) &\equiv \lambda^{(0)}_{\mathcal{U}}(t)+\mathbf{P}^T_{\mathcal{O}\mathcal{U}}(t)\Gamma_{\mathcal{O}}(t)\mathbf{s}_{\mathcal{O}}(t)\\ \mathbf{C}(t) &\equiv \left(\mathbf{P}^T_{\mathcal{U}\mathcal{U}}(t)-\mathbf{I}\right)\mathbf{\Gamma}_{\mathcal{U}}(t).\end{aligned}$$

If the service rates and the routing probability matrix are independent of time: $\mu_i(t)=\mu_i$ and $P_{i,j}(t)=P_{i,j}$, i.e., the model becomes the $(G_t/M/s_t+GI_t)^m/M$ network, then $\mathbf{\Gamma}_{\mathcal{U}}\equiv\mathbf{\Gamma}_{\mathcal{U}}(t)=diag(\mu_{\mathcal{U}})$, $\mathbf{C}\equiv\mathbf{C}(t)=\left(\mathbf{P}^T_{\mathcal{U}\mathcal{U}}-\mathbf{I}\right)\mathbf{\Gamma}_{\mathcal{U}}$, and (4.1) has the unique solution

$$\mathbf{B}_{\mathcal{U}}(t) \;=\; e^{-\mathbf{C}\,t}\left(\int_0^t e^{-\mathbf{C}\,u}\mathbf{D}(u)du+\mathbf{B}(0)\right).$$

In all cases, the TAR vector

$$\lambda(t)=\lambda^{(0)}(t)+\mathbf{P}^T(t)\mathbf{\Gamma}(t)\cdot\mathbf{B}(t). \tag{4.2}$$

When $m=2$, analytic formulae are available, see Appendix E.1.

4.2. The Overall Algorithm and its Complexity

Just as for FASQ in §2, the key step beyond direct computation is to control the switching between regimes. Since each queue can be either UL or OL, there are overall 2^m different network regimes. We say that the system changes its regime at some time if one or more of the queues changes its regime, either from UL to OL or from UL to OL. We provide the following regime termination time

$$\begin{aligned}T_{\mathcal{R}}(t_0) &\equiv T_1(t_0)\wedge T_2(t_0), \quad \text{where} \qquad (4.3)\\ T_1(t_0) &\equiv \inf\{t>t_0 : \text{some } i\in\mathcal{O}(t_0) \text{ s.t. } Q_i(t)=0, \lambda_i(t)\le\sigma_i(t)\},\\ T_2(t_0) &\equiv \inf\{t>t_0 : \text{some } j\in\mathcal{U}(t_0)\,{}_{\text{s.t.}}\; B_j(t)=s_j(t), \lambda_j(t)>\sigma_j(t)\},\end{aligned}$$

with t_0 being the starting time of the desired interval and the infimum of an empty set understood to be infinity.

Within each regime, we use an ODE to compute the TARs $\lambda_i(t)$ and the service content functions $B_i(t)$, based on (4.1) and (4.2). Given the TARs at all queues, we use the FASQ to calculate the performance functions. We give a formal algorithm statement in §E.

The computational complexity clearly depends largely on the computational complexity of the ODE solver. Fortunately the ODE's arising in the present context tend not to be computationally difficult; e.g., they are rarely stiff. Let $\mathcal{O}_{ode}(m,t)$ be the computational complexity for solving an m-dimensional ODE over an interval of length t. For the conventional solvers we use (see §6.1), we should have approximately $\mathcal{O}_{ode}(m,t) = O(mt)$. From the structure of algorithm Alg(ODE), we can determine the computational complexity $\mathcal{C}_{ODE} \equiv \mathcal{C}_{ODE}(T,m,\mathcal{S})$, as a function of the number of queues m, length of the time interval T and the number of regime switches per queue, $\mathcal{S}$, but we will also confirm it in numerical examples. As before in §3.2, the parameter pair (m,T) is directly observable, but $\mathcal{S}$ is not. Let $\mathcal{S}_i$ be the number of regime switches at queue i over $[0,T]$. Hence the total number of regime switches for any queue in the network is $\sum_{i=1}^{m} \mathcal{S}_i$. Assuming that $\mathcal{S}_i \approx \mathcal{S}$ for all i as before, we see that the ODE must be solved $m\mathcal{S}$ times over subintervals, whose combined length is T. In addition, there is some computational cost of carrying out the switching in each regime switch. For the ODE portion of the algorithm, the computational complexity is

$$\mathcal{O}(m,\mathcal{S},T) = \sum_{j=1}^{m\mathcal{S}} \mathcal{O}(m,T_i) \quad \text{where} \quad \sum_{j=1}^{m\mathcal{S}} T_i = T. \tag{4.4}$$

Hence, the overall computational complexity for the ODE solver is $O(mT)$. But we must factor in the regime switching, which has computational effort proportional to the number of network regime switches, which is $O(m\mathcal{S})$. Assuming that these components each contribute significantly we get an overall computational complexity

$$\mathcal{C}_{ODE} \equiv \mathcal{C}_{ODE}(T,m,\mathcal{S}) = O(m^2\mathcal{S}T). \tag{4.5}$$

We find that formula (4.5) is consistent with numerical examples. e.g., see Figure 2 in §6.3.

5. Allowing GI Service Distributions: Alg(FPE,GI)

We now generalize the model, allowing the service distribution at each queue to be GI instead of M; for motivation, see Brown et al. (2005). We need a new algorithm because neither the FPE-based algorithm Alg(FPE) in §3 nor the ODE-based algorithm Alg(ODE) in §4 is directly applicable. For simplicity, we focus on the $(G_t/GI/s_t + GI)^m/M_t$ FQNet, where the service and patience distributions are not time varying; the analysis easily can be generalized to $(G_t/GI_t/s_t + GI_t)^m/M_t$. As part of the model data, we let $(G_i, 1 \le i \le m)$ be the general service cdf's of the $(G_t/GI/s_t + GI)^m/M_t$ FQNet and let $\bar{G}_i \equiv 1 - G_i$ be the associated ccdf's; e.g., $\bar{G}_i(x) = e^{-\mu_i x}$ for M service.

5.1. A New FPE for the TAR Vector

The key is to obtain the TAR $\lambda_i(t)$ for $1 \leq i \leq m$ and $0 \leq t \leq T$. Once $\lambda_i(t)$ is obtained, the single-queue algorithm for GI service developed in Liu and Whitt (2011a) can be applied to compute all other performance measures; see §8 and Appendix G there. This single-queue algorithm for GI service is a generalization of FASQ, which requires solving another FPE to find the rate that fluid enters service $b(t,0)$ (which we call the *rate into service* (RIS)) during each OL interval. For M service, this FPE for RIS simplifies to (2.24) with $x = 0$.

We next analyze the transient dynamics of the $(G_t/GI/s_t + GI)^m/M_t$ model at arbitrary time t assuming the knowledge of the current system status. We refer to the explicit formulas for $b(t,x)$ developed in Liu and Whitt (2011a) during our analysis. The formulas for $q(t,x)$ and $w(t)$ are identical to those in §2.

Consider a queue j that is UL, i.e., $j \in \mathcal{U}(t)$, from Proposition 2 of Liu and Whitt (2011a) we have that (as a generalization of (2.23))

$$\begin{aligned} b_j(t,x) &= \bar{G}_j(x)\lambda_j(t-x)\mathbf{1}_{\{x \leq t\}} + \frac{\bar{G}_j(x)}{\bar{G}_j(x-t)} b_j(0,x-t)\mathbf{1}_{\{x>t\}}, \\ \sigma_j(t) &= \int_0^\infty b_j(t,x) h_{G,j}(x)dx \\ &= \int_0^t g_j(x)\lambda_j(t-x)dx + \int_0^\infty \frac{g_j(x+t)}{\bar{G}_j(x)} b_j(0,x)dx. \end{aligned} \tag{5.1}$$

Note that the above formula for queue j is in terms of the TAR λ_i which is unknown.

Consider a queue k that is OL, i.e., $k \in \mathcal{O}(t)$. From equations (17)-(20) of Liu and Whitt (2011a), we obtain

$$\sigma_k(t) = b_k(t,0) - s_k'(t), \tag{5.2}$$

where the RIS $b_k(t,0)$ satisfies the FPE (as a generalization of (2.24))

$$b_k(\cdot,0) = \Phi(b_k(\cdot,0)), \tag{5.3}$$

with

$$\begin{aligned} \Phi(y)(t) &\equiv \hat{a}_k(t) + \int_0^t y(t-x)g_k(x)dx, \\ \hat{a}_k(t) &\equiv s_k'(t) + \int_0^\infty \frac{b_k(0,y)g_k(t+y)}{\bar{G}_k(y)}dy. \end{aligned}$$

Moreover, we have shown in Theorem 2 of Liu and Whitt (2011a) that Φ is a contraction operator under mild conditions, which thus implies that the FPE (5.3) has a unique solution.

We note that the RIS for an OL queue depends on the rate at which the service capacity becomes available (defined in (2.15)) and is independent of the TAR, unlike during an UL regime. Hence, having $\sigma_k(t)$ and $b_k(t,0)$ available (by solving the FPE (5.3) and (5.2)) for all OL queues (i.e., for all $k \in \mathcal{O}(t)$), the TAR of queue i satisfies the following traffic-rate equation

$$\begin{aligned}\lambda_i(t) &= \lambda_i^{(0)}(t) + \sum_{k\in\mathcal{O}(t)} P_{k,i}(t)\sigma_k(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\sigma_j(t) \\ &= \hat{\gamma}_i(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\left(\int_0^t g_j(x)\lambda_j(t-x)dx\right), \end{aligned} \tag{5.4}$$

where

$$\hat{\gamma}_i(t) \equiv \lambda_i^{(0)}(t) + \sum_{k\in\mathcal{O}(t)} P_{k,i}(t)\sigma_k(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\int_0^\infty \frac{g_j(x+t)}{\bar{G}_j(x)} b_j(0,x)dx,$$

with $\hat{\gamma}_i$ not depending on the TAR and determined by the FPE (5.3) and the second equality holding by (5.1).

Equation (5.4) expresses the TAR vector λ as the solution of a FPE, i.e.,

$$\lambda = \mathcal{J}(\lambda), \tag{5.5}$$

where $\mathcal{J} : \mathbb{D}^m \to \mathbb{D}^m$ with

$$\mathcal{J}(u)_i(t) \equiv \hat{\gamma}_i(t) + \sum_{j\in\mathcal{U}(t)} P_{j,i}(t)\left(\int_0^t g_j(x)u_j(t-x)dx\right), \quad 1 \le i \le m, \tag{5.6}$$

where $u \equiv (u_1, \ldots, u_m) \in \mathbb{D}^m$. Under regularity conditions, we can show that there exists a unique solution to equation (5.4) by applying the Banach contraction theorem. We will use the complete (nonseparable) normed space $\mathbb{D}^m$ with the uniform norm over the interval $[0,T]$, i.e.,

$$\|u\|_T \equiv \sum_{i=1}^m \sup_{0\le t\le T} |u_i(t)|. \tag{5.7}$$

Theorem 5.1 (*TAR for GI service*) *Assume the system regime does not change in a small interval* $[0,T]$*, then the operator* $\mathcal{J}$ *in* (5.6) *is a monotone contraction operator on* $\mathbb{D}^n$ *with norm defined in* (5.7).

Proof. Assume that $T > 0$ is small enough so that the system regime does not change, i.e., $\mathcal{U}(t) = \mathcal{U}$ and $\mathcal{O}(t) = \mathcal{O}$ for $0 \le t \le T$. Then

$$\begin{aligned}
\|\mathcal{J}(u_1) - \mathcal{J}(u_2)\|_T &= \sum_{i=1}^{m} \sup_{0\le t\le T} \left| \sum_{j\in\mathcal{U}} P_{j,i}(t) \left[\int_0^t g_j(x) \left(u_{1,j}(t-x) - u_{2,j}(t-x)\right) dx \right] \right| \\
&\le \sum_{i=1}^{m} \sup_{0\le t\le T} \sum_{j\in\mathcal{U}} \|u_{1,j} - u_{2,j}\|_T P_{j,i}(t) G_j(t) \\
&\le m \max_{1\le j\le m} G_j(T) \cdot \sup_{0\le t\le T} \sum_{j\in\mathcal{U}} \|u_{1,j} - u_{2,j}\|_T \le \tilde{C}(T) \|u_1 - u_2\|_T,
\end{aligned}$$

where $\tilde{C}(T) \equiv m \max_{1\le j\le m} G_j(T)$. This provides what we need, because we can make $\tilde{C}(T) < 1$ for sufficiently small $T > 0$, because $G_i(t) \to 0$ as $t \to 0$ for all $1 \le i \le m$ by our assumption on the existence of the service densities. ∎

5.2. The Overall FPE-Based Algorithm with GI Service

Algorithm Alg(FPE,GI) has two parts: (i) regime switching and (ii) the new FPE within each fixed network regime. The regime switching can be managed just as for the FASQ and Alg(ODE). As before, we work with a regime switching step size ΔT. Given a time t, we apply the new FPE in §5.1 to find a new TAR vector over the interval $[t, t + \Delta T]$. However, after doing that calculation, we must check to see if there is a regime switch at any queue in the network. If such a regime switch occurs at time $s \in [t, t + \Delta T]$, then we replace t with s and repeat. In this way, we move forward in time until we compute the TAR vector for all of $[0, T]$.

Within each interval with fixed network regime, we calculate the TAR using FPE (5.5). Given that TAR within each interval with fixed network regime, we apply the single-queue algorithm from Liu and Whitt (2011a) to calculate the queue performance at each queue. That is more complicated than the FASQ in §2, because it is necessary to solve the FPE (5.3) at each queue that is OL in that particular network regime.

For this last algorithm, the computational complexity is more difficult to determine from the algorithm structure, because the algorithm is more complicated. Just as for Alg(ODE), there are $O(m\mathcal{S})$ network regimes, so that regime switching should have complexity of order $O(m\mathcal{S})$. The new FPE is more complicated, requiring an FPE within the overall FPE at each queue. Since the first-step FPE (5.3) is done at each queue throughout $[0, T]$, we can estimate its complexity as $O(mT)$. The second-step FPE (5.5) also may have complexity of

order $O(mT)$. In addition, these FPE's depend on the ETPs ϵ. Since both operators are contraction, the rate of convergence is geometric. Hence the computational complexity of both iterations as functions of ϵ are $O(\log(1/\epsilon))$. Thus, we estimate that the computational complexity should be

$$\mathcal{C}_{FPE,GI}(m,T,\mathcal{S},\epsilon) = O\left((mT+mT)\, m\mathcal{S}\log\left(\frac{1}{\epsilon}\right)\right) = O(m^2\mathcal{S}T\log(1/\epsilon)). \tag{5.8}$$

6. Examples

In this section we report the results of implementing the algorithms in §§3-5 and applying them to three examples: (i) a Markovian $(M_t/M/s+M)^2/M$ two-queue FQNet, (ii) a Markovian $(M_t/M/s+M)^m/M$ FQNet with m queues, $2 \le m \le 160$, and (iii) a non-Markovian $(G_t/LN/s+E_2)^2/M$ model. For simplicity, in these example we make only the arrival rate time-varying. The extension to time-varying staffing is of course very important and is not difficult to do as well, as we illustrate with an example in the appendix. Adding time-varying functions to the service, abandonment and routing is less important, so we do not directly illustrate those extensions. The third algorithm applies to all three examples, but the first two algorithms only apply to the first two examples. In §6.1 we first provide details about our implementation.

6.1. Implementation Details

Before discussing the examples, we briefly explain how we implemented the numerical algorithms and conducted the simulation experiments. For both, we used Matlab on a personal computer. To numerically solve ODEs, both one-dimensional for $w(t)$ at each queue, as in (2.26), and multi-dimensional for the TAR, as in (4.1), we used the Matlab solvers "ode23" and "ode45", which employ automatic step-size Runge-Kutta-Fehlberg integration methods. The first one, ode23, uses a pair of simple second-order and third-order formulas. The second, ode45, uses a pair of fourth-order and fifth-order formulas. See Thomas (1995) for details on finite-difference methods for numerically solving differential ODE's. As a base case for the examples, we considered a system starting empty over the time interval $[0,T]$ with $T = 20$. In that framework, we divided continuous time interval $[0,T]$ into discrete intervals with length 0.002.

Care is needed in estimating the various time-dependent performance functions in the simulation experiments. For the mean head-of-line waiting time, $E[W(t)]$, the mean queue

length $E[Q(t)]$ and the mean number of busy servers $E[B(t)]$, we divide the interval $[0,T]$ into subintervals or bins. For $E[W(t)]$, we kept track of all customer arrivals in each sample path. For a customer n, we keep track of the arrival time A_n, and the time that the customer enters service E_n. Therefore, one value for this sample path is $(t, \hat{W}(t)) = (E_n, E_n - A_n)$. Of course, this customer may have already abandoned by time E_n. Since we are interested in the potential waiting time, assuming infinite patience, we keep track of the time that the customer would enter service even after it abandons; i.e., our procedure includes the behavior of virtual customers. The bin size for $E[W(t)]$ is 0.1, while the bin size for $E[Q(t)]$ and $E[B(t)]$ is 0.05. Thus, we sampled the queue length once every 0.05 units of time.

6.2. A Two-Queue FQNet Example

We first consider the two-queue $(M_t/M/s+M)^2/M$ FQNet discussed in §1. It has sinusoidal external arrival rates

$$\lambda_i^{(0)}(t) = a_i + b_i \sin(c_i\, t + \phi_i), \quad i = 1, 2, \tag{6.1}$$

exponential service and patience distributions: $\bar{G}_i(x) = e^{-\mu_i\, x}$, $\bar{F}_i(x) = e^{-\theta_i\, x}$, $i = 1, 2$, constant staffing functions s_i, $i = 1, 2$, and a constant 2×2 Markov transition probability matrix P with elements $P_{1,2} = P_{2,1} = 0.2$ and $P_{i,i} = 0.3$, so that $P_{i,0} = 0.5$, $i = 1, 2$. Let $a_1 = a_2 = 0.5$, $b_1 = 0.25$, $b_2 = 0.35$, $c_1 = c_2 = 1$, $\phi_1 = 0$, $\phi_2 = -3$, $\mu_1 = 1$, $\mu_2 = 0.5$, $\theta_1 = 0.5$, $\theta_2 = 0.3$, $s_1 = 1$, and $s_2 = 2$. We let the network be initially empty.

We first show how the FPE-based algorithm Alg(FPE) from §3 works. It is based on a FPE for the total arrival rates (TARs) $\lambda_1(t)$ and $\lambda_2(t)$ for $0 \le t \le T$, Figure 6 in Appendix G.1 displays the arrival rates in successive iterations, dramatically showing both the monotone convergence and the geometric rate of convergence of the operator Ψ in §3.1. Alg(FPE) terminates after iteration $\mathcal{I}(\epsilon)$, where $\epsilon > 0$ is the pre-specified error tolerance parameter (ETP) and

$$\mathcal{I}(\epsilon) \equiv \inf\left\{ n \ge 0 : \mathcal{E}_T(n) \equiv \max_{j=1,2} \|\lambda_j^{(n)} - \lambda_j^{(n-1)}\|_T \le \epsilon \right\},$$

yielding final TARs $\lambda_i \equiv \lambda_i^{(\mathcal{I}(\epsilon))}$, $i = 1, 2$ For this example, we show how the number of iterations $\mathcal{I}(\epsilon)$, the total run time $\mathcal{T}(\epsilon)$ and the terminating error $\mathcal{E}_T(\mathcal{I}(\epsilon))$ depend on the EPT ϵ in Table 1.

Figure 7 in Appendix G.1 shows plots of all the standard performance functions in the fluid network using Alg(FPE), including λ_i, Q_i, w_i, B_i, X_i, and $b_i(\cdot, 0)$, $i = 1, 2$. Figure

$\log_{10}(\epsilon)$	-1	-2	-3	-4	-5	-6	-7	-8	-9
$\mathcal{I}(\epsilon)$	3	6	8	11	13	15	16	17	19
$\mathcal{T}(\epsilon)$	1.03	1.82	2.41	2.90	3.12	3.67	3.94	4.28	4.73
$\mathcal{E}_T(\mathcal{I}(\epsilon))$	0.081	0.007	9.2E-4	4.8E-5	4.9E-6	2.8E-7	5.2E-8	8.3E-9	1.4E-10

Table 1: The number of iterations $\mathcal{I}(\epsilon)$, computation time $\mathcal{T}(\epsilon)$ and terminating error $\mathcal{E}_T(\mathcal{I}(\epsilon))$ for algorithm Alg(FPE) as a function of the ETP $\epsilon \equiv 10^{-n}$, $n \geq 1$, for the two-queue FQNet example using $T = 20$ and $\Delta T = 2$.

1 compares the fluid approximations with results from a simulation experiment for a very large-scale queueing system. The queueing model has nonhomogeneous Poisson external arrival processes with sinusoidal rate functions $\lambda_{n,i}^{(0)}(t) = n\lambda_i^{(0)}(t)$, $i = 1, 2$, with $n = 4000$. We compare the fluid model predictions to a single sample path of the queueing system (one simulation run). In Figure 1 the solid lines are the simulation estimations of single sample paths applied with fluid scaling, and the dashed lines are the fluid approximations.

When the scale of the queueing model is not large, i.e., when n is smaller, single sample paths of the queueing functions typically do not agree closely with the fluid functions because of stochastic fluctuations. However, the mean functions of these processes can be well approximated, as shown in Appendix G.1 in Figure 8 for the case $n = 50$. In this example, the two queues do not become OL (UL) at the same time because of the phase difference of the external arrival rates (i.e., $\phi_1 = 0$, $\phi_2 = -3$). We also consider different phases ϕ_i in another example in Appendix G.1.

All three algorithms were run on this example; the resulting identical performance functions confirm all the algorithms. For this small FQNet example, the most important characteristic is ease of implementation, for which the Alg(ODE) from §4 tends to be easiest, while Alg(FPE,GI) from §5 is hardest. For all examples, Alg(FPE,GI) tends to have the longest run time, as expected because it involves an FPE for each queue, as well as an FPE for the TARs. For two-queue examples like the one just considered, the running time of Alg(FPE,GI) tends to be twice as long as for Alg(ODE).

6.3. A Network with Many Queues

We next evaluate the performance of algorithms Alg(FPE) and Alg(ODE) as a function of the number of queues, m. To do so, we consider a simple idealized network with m queues. Each queue i has a time-varying arrival rate as in (6.1), exponential service and patience

times with rate μ_i and θ_i, constant staffing level s_i, and constant routing probabilities $P_{i,j}$, where

$$a_i = 0.5, \quad b_i = ia_i/m, \quad \phi_i = \pi(1.5 - i/m), \quad \theta_i = 0.5,$$
$$c_i = s_i = \mu_i = 1, \quad P_{i,j} = 1/2m, \quad 1 \le i \le m, 1 \le j \le m.$$

Figure 13 in Appendix G.2 shows plots of the performance functions for $m = 10$.

m	2	4	6	8	10	12	14	16	18	20
$\mathcal{I}(m)$	12	12	12	13	12	12	12	12	12	12
$\mathcal{T}(m)$	2.86	4.68	6.43	8.75	11.02	11.96	13.96	15.63	17.39	19.21
m	30	40	50	60	70	80	100	120	140	160
$\mathcal{I}(m)$	12	12	12	12	12	12	12	12	12	12
$\mathcal{T}(m)$	29.76	37.37	48.67	58.42	68.15	73.63	96.77	115.0	134.84	147.7

Table 2: The number of iterations $\mathcal{I}(m)$ and computation time $\mathcal{T}(m)$ (seconds) as a function of m, the number of queues, using ALG(FPE) with fixed EPT $\epsilon = 10^{-5}$.

m	2	4	6	8	10	12	14	16	18	20
$\mathcal{T}(m)$	2.77	3.67	6.16	8.92	12.03	15.46	20.35	25.95	31.30	37.37
m	30	40	50	60	70	80	100	120	140	160
$\mathcal{T}(m)$	64.72	107.05	132.65	178.7	227.64	312.61	411.09	567.15	765.55	1013.1

Table 3: The computation time $\mathcal{T}(m)$ (seconds) as a function of the number of queues m using ALG(ODE).

Table 2 shows the number of iterations $\mathcal{I}(m)$ and computation time $\mathcal{T}(m)$ in seconds as a function of the number of queues, m, $2 \le m \le 160$, using algorithm Alg(FPE) with fixed EPT $\epsilon = 10^{-5}$. In this example we observe that (i) the number of iterations $\mathcal{I}(m)$ does not grow with the number of queues, m, and (ii) the computation time $\mathcal{T}(m)$ grows linearly in m.

We also analyzed the performance of this same model using Alg(ODE). Table 3 shows the computation times $\mathcal{T}(m)$ as a function of m. Since we used the ODE solvers ode23 and ode45, which are $O(m)$ algorithms, the running time for Algorithm 3 becomes $O(m^2\mathcal{S})$. Figure 2 dramatically shows the difference in the algorithm performance.

We conclude this section with some general observations comparing the performance of the two algorithms Alg(FPE) and Alg(ODE). For small m (e.g., $2 \le m \le 8$) and small ϵ

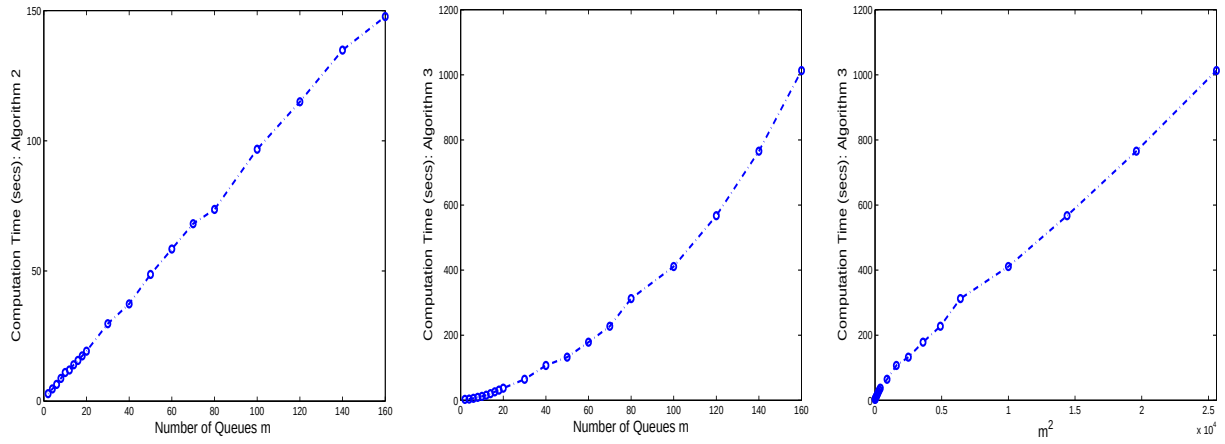

Figure 2: Computing times of algorithms Alg(FPE) and Alg(ODE) for the m-queue FQNet as a function of m, $2 \leq m \leq 160$.

(e.g., $\epsilon < 10^{-5}$), ALG(ODE) runs faster than Alg(FPE); for big m and medium ϵ, Alg(FPE) runs faster than Alg(ODE). Of course, the complexity of Alg(ODE) depends on the choice of the multi-dimensional ODE solver. The polynomial growth in m as shown in Table 3 is attributed to the specific numerical scheme (such as Runge-Kutta-Fehlberg) of the ODE solver.

6.4. A $(G_t/LN/s + E_2)^2/M$ non-Markovian Example

We now consider an example with a non-exponential service-time distribution, for which only the final algorithm Alg(FPE,GI) introduced in §5 applies. To illustrate this example, we consider the $(G_t/LN/s + E_2)^2/M$ model with Lognormal service distributions at each queue (the LN) and Erlang-2 patience distributions at each queue (the E_2). Specifically, we let the service time at station i be $S_i \equiv e^{Z_i}$, where Z_i is a normal random variable with mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$, i.e., $Z_i \sim N(\hat{\mu}_i, \hat{\sigma}_i^2)$, $i = 1, 2$. The service pdf is

$$g_i(x) = \frac{1}{x\hat{\sigma}_i\sqrt{2\pi}}\, e^{-\frac{(\log x - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2}}, \quad x \geq 0, \quad i = 1, 2.$$

For $i = 1, 2$, the mean service times and the variances are

$$\mu_i^{-1} \equiv E[S_i] \;=\; e^{\hat{\mu}_i + \frac{1}{2}\hat{\sigma}_i^2} \quad \text{and} \quad \sigma_i^2 \equiv Var(S_i) = (e^{\hat{\sigma}_i^2} - 1)\, e^{2\hat{\mu}_i + \hat{\sigma}_i^2}.$$

The LN assumption is representative because Brown et al. (2005) showed that service times in call centers follow LN distributions.

We let the patience distribution be Erlang-2 (E_2) with pdf

$$f_i(x) = 4\theta_i^2 x\, e^{-2\theta_i x}, \quad x \geq 0.$$

Letting A_i be a generic patience time of a customer at queue i, we have $E[A_i] = 1/\theta_i$, $i = 1, 2$. The E_2 distribution has a squared coefficient of variation $c^2 \equiv Var(X)/E[X]^2 = 1/2$. We choose $\hat{\mu}_1 = -0.549$, $\hat{\sigma}_1 = 1.048$, $\hat{\mu}_2 = 0.144$, $\hat{\sigma}_2 = 1.048$ such that $\mu_1 = 1$, $\mu_2 = 0.5$, $\sigma_1^2 = 2$, $\sigma_2^2 = 8$. Thus, we have $c^2 = 2$ for the service distributions. We let $\theta_1 = 0.5$, $\theta_2 = 0.3$. In this way both the service rates (μ_1 and μ_2) and the patience rates (θ_1 and θ_2) remain the same as in the example in §6.2. For comparison purpose, we let the external arrival rate $\lambda^{(0)}$ be sinusoidal as in (6.1) and the Markovian routing matrix $\mathbf{P}$ be constant with the same parameters as in §6.2. We also let the system be initially empty.

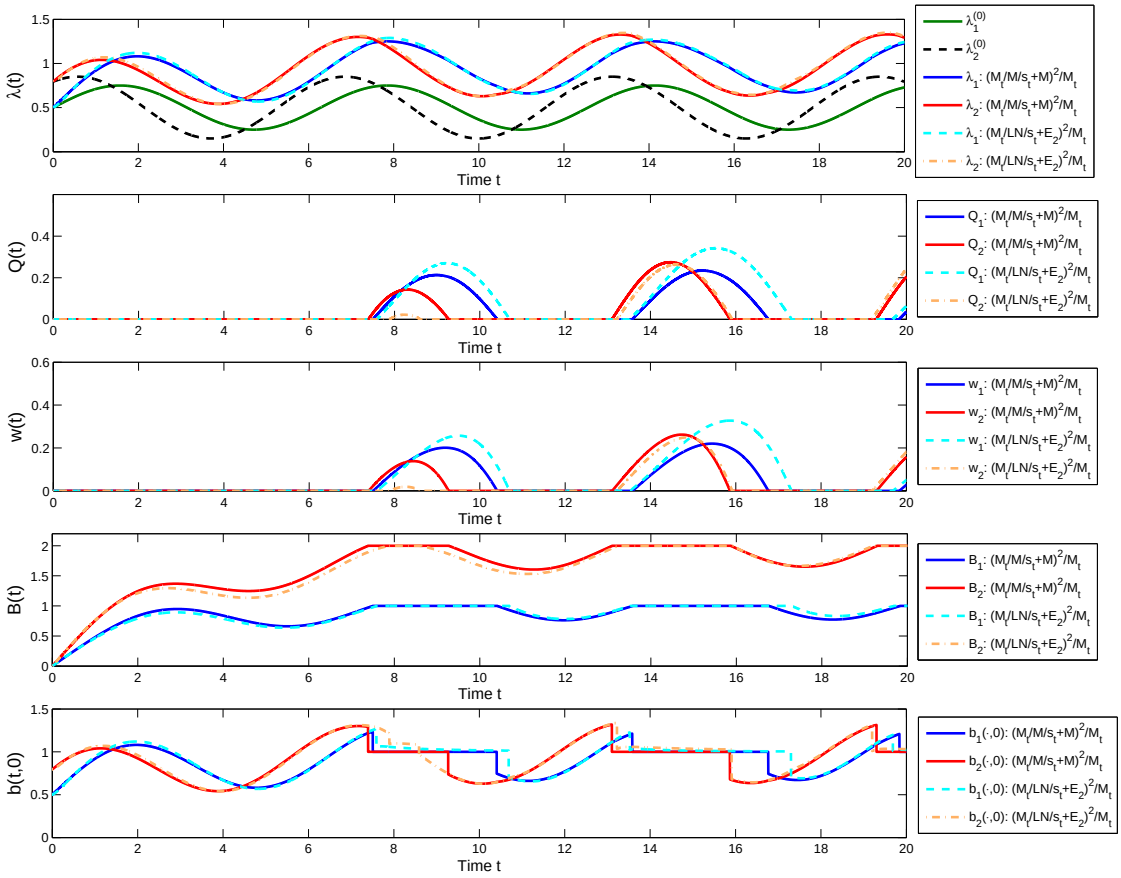

Figure 3: Computing the fluid performance functions for the $(M_t/LN/s_t + E_2)^2/M_t$ network fluid model.

Figures 3 above and 14 in Appendix G.3 show plots of the standard performance functions and compares them to simulation experiments in the two cases $n = 4000$ and $n = 50$. These

two figures are analogs of Figures 7 and 8. As before, for $n = 4000$ the fluid performance agrees with individual sample paths of the SQNet; for $n = 50$ the fluid performance agrees with the mean values of the time-varying stochastic SQNet performance. In Figure 3, we compare the fluid functions of the two-queue Markovian model (the solid lines: blue for Queue 1 and red for Queue 2) and those of the non-Markovian $(M_t/LN/s + E_2)^2/M$ model (the dashed lines: light blue for Queue 1 and light brown for Queue2). As indicated above, these two models have the same model parameters, including the service and patience rates μ and θ, except for the service and patience distributions.

In addition to showing that the new algorithm Alg(FPE,GI) is effective, Figure 3 shows that the service and patience distributions beyond their means play an important role in the time-dependent performance of the fluid network with time-varying model parameters. For the stationary $G/GI/s+GI$ fluid queue, Whitt (2006) showed that the patience distribution beyond its mean plays an important role, while the service-time distribution does not. In Liu and Whitt (2011a) we showed that the service-time distribution beyond its mean is also important in the time-dependent behavior.

Acknowledgments

This research was supported by NSF grant CMMI 1066372.

References

Brown, L., N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao. 2005. Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Statist. Assoc.* **100** 36–50.

Green, L. V., P. J. Kolesar, W. Whitt. 2007. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management* **16** 13–39.

Kang, W., G. Pang. 2011. Computation and properties of fluid models for time-varying many-server queues with abandonment. Working paper, University of Maryland at Baltimore County and the Pennsylvania State University. Available at: Available at: http://www2.ie.psu.edu/pang

Liu, Y., W. Whitt. 2011a. A fluid approximation for the $G_t/GI/s_t + GI$ queue. *Queueing Systems*, to appear. Available at: http://www.columbia.edu/∼ww2040/allpapers.html

Liu, Y., W. Whitt. 2011b. A network of time-varying many-server fluid queues with customer abandonment. *Operations Research.* **59** 835-846.

Liu, Y., W. Whitt. 2011c. Large-Time Asymptotics for the $G_t/M_t/s_t + GI_t$ Many-Server Fluid Queue with Abandonment. *Queueing Systems.* **67** 145-182.

Liu, Y., W. Whitt. 2011d. Many-Server Heavy-Traffic Limits for Queues with Time-Varying Parameters. Working paper. Available at: http://www.columbia.edu/∼ww2040/allpapers.html

Liu, Y., W. Whitt. 2011e. A Many-Server Fluid Limit for the $G_t/GI/s_t+GI$ Queueing Model experiencing Periods of Overloading. Working paper. Available at: http://www.columbia.edu/∼ww204

Mandelbaum, A., W. A. Massey, M. I. Reiman. 1998. Strong approximations for Markovian service networks. *Queueing Systems.* **30** 149-201.

Mandelbaum, A., W. A. Massey, M. I. Reiman, B. Rider. 1999a. Time varying multiserver queues with abandonments and retrials. *Proceedings of the 16th International Teletraffic Congress*, P. Key and D. Smith (eds.).

Mandelbaum, A., W. A. Massey, M. I. Reiman, A. Stolyar. 1999b. Waiting time asymptotics for time varying multiserver queues with abandonment and retrials. *Proceedings of the Thirty-Seventh Annual Allerton Conference on Communication, Control and Computing*, Allerton, IL, 1095-1104.

Massey, W. A., W. Whitt. 1993. Networks of infinite-server queues with nonstationary Poisson input. *Queueing Systems.* **13** 183-250.

Nelson, B.L., M. R. Taaffe. 2004a. The $Ph_t/Ph_t/\infty$ Queueing System: Part I - The Single Node. *INFORMS Journal on Computing.* **16**(3) 266-274.

Nelson, B.L., M. R. Taaffe. 2004b. The $[Ph_t/Ph_t/\infty]^k$ Queueing System: Part II - The Multiclass Network. *INFORMS Journal on Computing.* **16**(3) 275-283.

Thomas, J. W. 1995. *Numerical Partial Differential Equations: Finite Difference Methods*, Springer, New York.

Whitt, W. 2006. Fluid models for multiserver queues with abandonments. *Operations Research.* **54** 37-54.

ONLINE SUPPLEMENT

Algorithms for Time-Varying Networks of Many-Server Fluid Queues

by Yunan Liu and Ward Whitt

A. Overview

A.1. The Main Paper

Following Liu and Whitt (2011a-b), in the main paper we have developed, implemented and tested three algorithms to compute the transient performance functions in fluid queue networks (FQNets). The first two algorithms apply to $G_t/M_t/s_t + GI_t)^m/M_t$ FQNets with m queues and proportional routing. The first algorithm Alg(FPE) in §3 iteratively solves a single fixed point equation (FPE) for the vector of total arrival rate (TAR) functions over the entire time interval $[0, T]$. Since the operator in the FPE is a contraction, the iteration for calculating the TAR converges geometrically fast. Given the TAR at each queue, we can apply the fluid algorithm for single queues (FASQ) with M_t service reviewed in §2 to each queue separately.

The second algorithm Alg(ODE) in §4 finds the TAR vector in any interval for which no queue changes its regime by solving an m-dimensional ordinary differential equation (ODE). However, just as for the FASQ, it is necessary to control the switching from one network regime to another. Both Alg(FPE) and Alg(ODE) have been shown to be effective. Alg(ODE) is appealing because it is easier to implement and is faster for small networks, but Alg(FPE) has been found to be more efficient for large networks, having run time of order $O(m)$ as compared to $O(m^2)$ for Alg(ODE).

The third algorithm Alg(FPE,GI) in §5 is a new algorithm to analyze FQNets with non-exponential service at each queue. This algorithm is less efficient computationally, but is appealing because non-exponential service times arise in many applications. Like algorithms FASQ and Alg(ODE), Alg(FPE,GI) requires working over intervals with a fixed network regime, so that the algorithm exploits a regime switching step size. For the individual queues with GI service, we apply the algorithm for the single fluid queue with GI service developed in Liu and Whitt (2011a), which is more complicated than the FASQ in §2 because

it involves another FPE to calculate the flow rate into service. Fortunately, the operators in both these FPE's are contractions, so that iterations converge geometrically fast.

In §6 we described results of implementing and testing these algorithms for FQNets. The examples showed that all algorithms are effective and can provide useful approximations for corresponding stochastic queueing networks (SQNets) of many-server queues, experiencing periods of overloading.

A.2. The Contents of this Supplement

This is a supplement to the main paper, with material expanding on the paper presented in the order of the sections in the main paper. At the outset in §1, we assumed that our staffing functions are feasible, never forcing fluid out of service. We start in §B by discussing how to detect the first violation of feasibility in the FQNets, if any, and how to find the minimum feasible staffing function greater than or equal to an initial one if that one is infeasible. Next in §C we present additional material on the algorithm for single fluid queues in §2. We give a formal statement of the algorithm and we examine the running time as a function of the interval length, T, the number of switches, $\mathcal{S}$, and the switching step size, ΔT.

We give formal statements of the FPE-based and ODE-based algorithms for FQNets in §§D and E. We give explicit formulas for the arrival rates using the ODE-based algorithm for the case $m = 2$ in §E.1. Finally, we provide more results for the examples in §G. In §G.1.2 we give an example with a time-varying staffing function, illustrating that it can be effectively treated as well.

B. Detecting Staffing Function Feasibility

We now discuss how to detect the first violation of feasibility of a staffing function and how to find the minimum feasible staffing function greater than or equal to the original staffing function if that one is infeasible.

In general, there is no guarantee that a staffing function s is feasible, i.e., having the property that the staffing function is set exogenously and adhered to, without forcing any fluid that has entered service to leave without completing service, because we allow s to decrease. (The fluid is assumed to be incompressible.) At any time $t > 0$ and $i \in \mathcal{O}(t)$, we require

$$b_i(t, 0) = s_i'(t) + \sigma_i(t) \geq 0. \tag{B.1}$$

We note that the above criterion becomes $s_i'(t)+\mu_i(t)\,s_i(t) \geq 0$ for all $i \in \mathcal{O}(t)$ for M_t service. Hence, we immediately have a sufficient condition for M_t service:

$$s_i'(t) + \mu_i\, s_i \geq 0, \quad \text{for all } 1 \leq i \leq m.$$

However, since we allow s_i to decrease, the feasibility condition in (B.1) might be violated in general when the given staffing functions decrease too quickly, i.e., when $-s_i' > 0$ is too large. The detection of infeasibility and construction of feasible staffing for a single fluid queue has already been discussed in §9 of Liu and Whitt (2011a). Also see Appendix G.2 there for an example. We next explain how to generalize our approach to the algorithms for the FQNets.

The idea is in the same spirit as in Liu and Whitt (2011a). Suppose in the kth iteration, the algorithm considers the time interval $\mathcal{I}_k \equiv [t^*, t^* + \Delta T]$. If there exists a time $t' \in \mathcal{I}_k$ such that the condition in (B.1) is violated for some $i \in \mathcal{O}(t')$, then we set $b_i(t, 0) = 0$ (shut the flow from the queue into the service facility) starting from time t'. Of course, the resulting staffing function $s_i^*(t) = B(t)$ will be strictly greater than $s_i(t)$. We continue with this strategy until our revised staffing function s_i^* coincides with the original s_i (if ever before T). See §9 and Appendix G.2 of Liu and Whitt (2011a) for more discussion.

For the FPE algorithm, we follow this strategy in each FPE iteration when we call FASQ with a temporary TAR $\lambda^{(k)}$, $k \geq 1$. For the ODE and FPE-GI algorithms, in each iteration interval I_j, we check the feasibility condition (B.1) for every $i \in \mathcal{O}(t)$, $i \in I_j$, and construct revised staffing functions if needed.

C. More on the Single-Queue Algorithm

We first give a formal statement of the single fluid-queue algorithm (FASQ) from Liu and Whitt (2011a-b) that was reviewed in §2.4. For each new regime switch time t, we use k to set up all computations needed from time t until the end of the interval at time T, but restart by selecting a new larger starting time whenever a regime switch is detected.

C.1. Sensitivity to the Switching Step Size

We now see how the FASQ run time depends on the switching step size ΔT. For that purpose, consider an $M_t/M/s_t + M$ queue over the time interval $[0, T]$ for $T = 20$ with a sinusoidal arrival rate $\lambda(t) = a+b\sin(c\cdot t)$, exponential service and abandonment distributions

Algorithm 1 : A Fluid Algorithm for Single Queues (FASQ) for the $G_t/M_t/s_t + GI_t$ fluid model, with model data $\mathcal{D} \equiv \left(\lambda, s, G, F, \hat{\mathcal{P}}(0)\right)$ and switching step size ΔT

Initialization: Set $\mathcal{R}(0)$ and let $t := 0$
repeat
  for $k = 0, 1, \ldots, \lceil (T-t)/\Delta T \rceil$ **do**
    Given $\mathcal{R}(t) = UL$, compute (b, B) in interval $[t + (k-1)\Delta T, t + k\Delta T]$ using (2.23);
    Given $\mathcal{R}(t) = OL$, compute $\mathcal{P}$ in interval $[t+(k-1)\Delta T, t+k\Delta T]$ using (2.24)-(2.27), (2.5)-(2.7);
    if $T_{\mathcal{R}}(t) < t + k\Delta T$, **then**
      $t := T_{\mathcal{R}}(t)$
      $\mathcal{R} := \{\text{OL}, \text{UL}\} \backslash \mathcal{R}$
      BREAK for-loop
    end if
  end for
until $t \geq T$.

$G(x) = 1 - e^{-\mu x}$ and $F(x) = 1 - e^{-\theta x}$, constant staffing $s(t) = s$, with $a = c = \mu = s = 1$, $b = 0.6$ and $\theta = 0.5$. We manually made the system switches for about 100 times between UL and OL intervals (i.e., $\mathcal{S} = O(100)$) by setting $c = 30$ so that the arrival-rate period is $\tau = 2\pi/c = 0.21$. The total number of periods in $[0, 20]$ is $N(\tau) = 20/\tau_1 = 95.2$.

In Table 4, we provide the computation times $\mathcal{T}(\Delta T)$ as a function of ΔT for $0.02 \leq \Delta T \leq T = 20$.

ΔT	20	10	6.67	5	4	2	1.33	1
$\mathcal{T}(\Delta T)$	15.36	12.89	10.25	7.99	6.62	3.57	3.54	1.93
ΔT	0.67	0.5	0.33	0.25	0.2	0.1	0.04	0.02
$\mathcal{T}(\Delta T)$	1.93	1.93	1.93	1.93	1.93	1.93	1.94	1.94

Table 4: The computation time $\mathcal{T}(\Delta T)$ (seconds) as a function of ΔT in the interval $[0, T]$, $T = 20$.

We observe that $\mathcal{T}(\Delta T)$ is almost insensitive to the choices of ΔT as long as ΔT is not too big. First, clearly it is not efficient to have a big ΔT when $\mathcal{S}$ is large. Suppose the current interval is $[t, t + \Delta T]$. If the system changes regime right after t, say at $t + h$, then all computation for the performance functions in $[t + h, t + T]$ is wasted and has to be redone later (possibly many times). Second, if ΔT is small, the computation obviously becomes efficient for large $\mathcal{S}$. Third, if we choose a small ΔT when $\mathcal{S}$ is small, for instance, the system stays OL (or UL) for the entirely interval $[0, T]$, the algorithm could experience

a large number of iterations $N(\Delta T) \equiv T/\Delta T$ before reaching time T. However, since the computation complexity in each iteration is linear in ΔT, the total complexity will be $(T/\Delta) \cdot O(\Delta T) = O(T)$, which is independent with ΔT. Hence, consistent with Table 4, we conclude that it should always be good to have a small ΔT regardless of the number of regime changes $\mathcal{S}$.

C.2. Sensitivity to $\mathcal{S}$ and T

We now see how the FASQ run time depends on the number of switches, $\mathcal{S}$, and the length of the time interval T. We consider the dependence upon $\mathcal{S}$ for fixed T and the dependence upon T for fixed $\mathcal{S}$. To do so, we use the same example considered in §C.1.

Table 5 gives the computation times $\mathcal{T}(\mathcal{S})$ for $1 \leq \mathcal{S} \leq 96$ in the interval $[0, T]$ with $T = 20$ by varying c. Here $\Delta T = 0.5$.

$\mathcal{S}$	1	2	4	7	13	20
$\mathcal{T}(\mathcal{S})$	0.98	0.75	0.76	1.13	1.76	2.50
$\mathcal{S}$	26	32	48	64	80	96
$\mathcal{T}(\mathcal{S})$	3.25	4.09	6.32	8.91	11.51	15.32

Table 5: The computation time $\mathcal{T}(\mathcal{S})$ (seconds) as a function of $\mathcal{S}$, the number of regimes switches (between UL and OL) using FASQ in the interval $[0, T]$, $T = 20$.

Next, Tables 6 and 7 give the computation times $\mathcal{T}(T)$ for $1 \leq T \leq 300$ in two cases: (i) with $c = 10\pi/T$ and (ii) with $c = 1$. We choose $\Delta T = T/150$. In case (i), we fixed the number of switches $\mathcal{S} \approx 5$ by decreasing c; in case (ii), the number of switches $\mathcal{S}$ grows linearly with T since c is a constant.

$\mathcal{S}$	5	10	15	20	25	30	40	50
$\mathcal{T}(T)$	0.267	0.32	0.71	0.97	1.25	1.53	2.12	2.72
$\mathcal{S}$	60	80	100	120	150	200	250	300
$\mathcal{T}(T)$	3.33	4.56	5.78	6.99	8.82	11.85	14.89	17.85

Table 6: The FASQ computation time $\mathcal{T}(T)$ (seconds) as a function of the length of time interval, T, for fixed total number of switches.

We plot $\mathcal{T}(\mathcal{S})$ as a function of $\mathcal{S}$ for fixed T in Figure 4(a) and $\mathcal{T}(T)$ as a function of T for fixed $\mathcal{S}$ in Figure 4(b) for case (i) and in Figure 5 for case (ii). Figure 4 shows

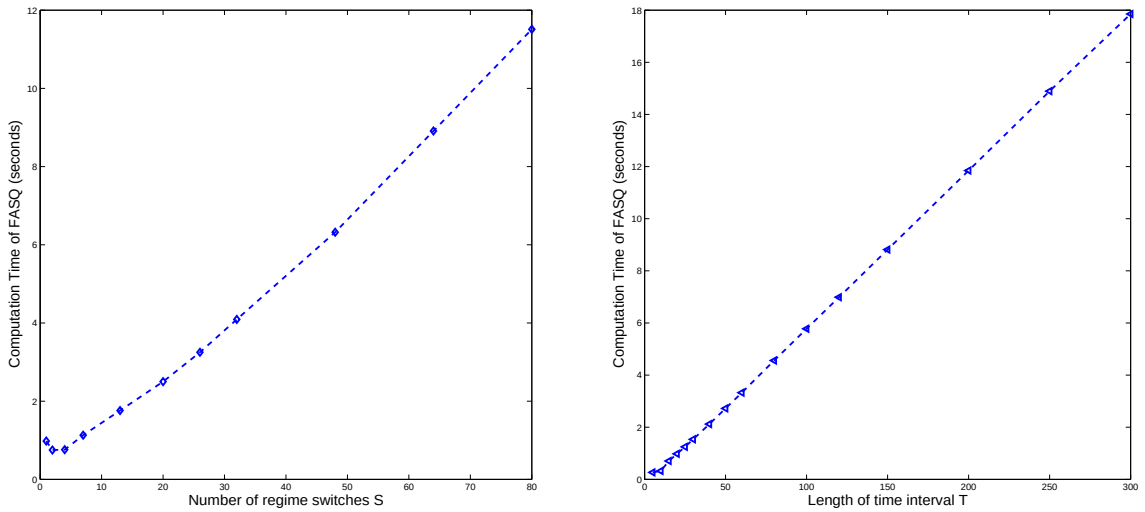

Figure 4: Computation times of FASQ in case (i) where $\mathcal{S}$ is independent of T, as functions of (a) the number of switches $\mathcal{S}$ for fixed T and (b) the length of time interval T for fixed $\mathcal{S}$.

that the computation time is linear in $\mathcal{S}$ (when T is fixed) and linear in T (when $\mathcal{S}$ is fixed). Figure 5 shows that the computation time is quadratic in T when $\mathcal{S} = O(T)$, because $\mathcal{T}(\mathcal{S}\,T) = O(\mathcal{S}\,T) = O(T^2)$. These experiment support our observations in §2.4.

$\mathcal{S}$	5	10	15	20	25	30	40	50
$\mathcal{T}(T)$	0.26	0.30	0.48	0.76	1.10	1.50	2.42	3.61
$\mathcal{S}$	60	80	100	120	150	200	250	300
$\mathcal{T}(T)$	4.98	8.40	12.71	17.94	27.34	47.54	73.33	104.31

Table 7: The computation time $\mathcal{T}(T)$ (seconds) as a function of the length of time interval, T, when $\mathcal{S}$, the number of switches, is proportional to T.

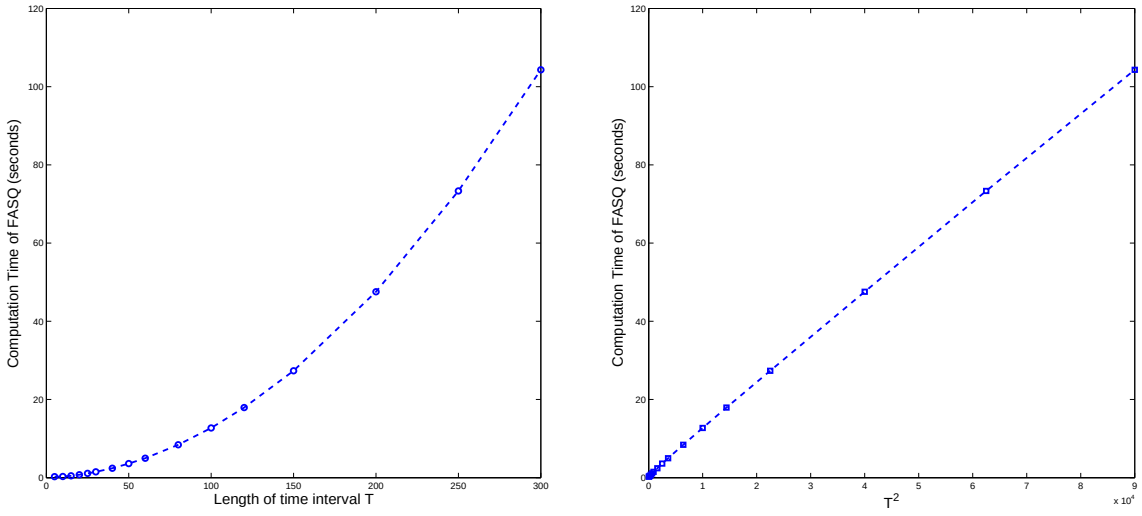

Figure 5: Computation times of FASQ in case (ii) where $\mathcal{S}$ is proportional to T, as functions of (a) T, the length of time interval, and (b) T^2.

D. The FPE-Based Algorithm

We now give a formal statement of the FPE-based algorithm in §3.

Algorithm 2 : An FPE based algorithm for the $(G_t/M_t/s_t + GI_t)^m/M_t$ Fluid Network

Initialization: $\lambda^{(1)} := \lambda^{(0)}$, $0 \leq i \leq m$,
for $k = 1, 2, \ldots$ **do**
  for $i = 1, 2, \ldots, m$ **do**
    Compute σ_i in $[0, T]$ using FASQ (Algorithm 1) with data $\left(\lambda_i^{(k)}, s_i, G_i, F_i, \hat{\mathcal{P}}_i(0)\right)$ and switching step size ΔT
  end for
  Let $\lambda^{(k+1)} := \lambda^{(0)} + P^T \cdot \sigma$ in $[0, T]$
  if $\|\lambda^{(k+1)} - \lambda^{(k)}\|_T < \epsilon$ **then**
    $\lambda := \lambda^{(k+1)}$
    Break
  end if
end for
Compute $\mathcal{P}_i$ for $1 \leq i \leq m$ using FASQ (Algorithm 1) with data $\left(\lambda_i, s_i, G_i, F_i, \hat{\mathcal{P}}_i(0)\right)$ and switching step sizeΔT.

E. The ODE-Based Algorithm

The ODE-based algorithm in §4 obtains the TAR vector over any interval over which there are no regime switches at any queue by solving an ODE. Thus, a key step is identifying successive intervals during which all queues remain in the same regime. Paralleling the FASQ, that is done with a network switching step size ΔT.

We now summarize the algorithm. It requires that we specify the desired time interval $[0, T]$, the vector of model data defined in (3.5), and a positive network switching step size ΔT (which should typically be shorter than for a single queue).

E.1. Explicit Formulas with the ODE-Based Approach for $m = 2$

The ODE-based approach in §4 yields analytic solutions when $m = 2$. Consider the following four system regimes:

(i) When Queue 1 is OL and Queue 2 is UL (i.e., $B_1(t) = s_1(t)$, $Q_1(t) \geq 0$, $B_2(t) < s_2(t)$),

$$\begin{aligned} B_1(t) &= s_1(t), \\ B_2'(t) &= \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)s_1(t) + (P_{2,2}(t) - 1)\mu_2(t)B_2(t), \end{aligned}$$

Algorithm 3 : An ODE based algorithm for the $(G_t/M_t/s_t + GI_t)^m/M_t$ Fluid Network

```
Initialization: t := 0
repeat
   for k = 0, 1, ..., ⌈(T − t)/ΔT⌉ do
      Compute λ(s) and B(s) for s ∈ [t + (k − 1)ΔT, t + k ΔT], using (4.1)-(4.2)
      Compute P(s) for s ∈ [t + (k − 1)ΔT, t + k ΔT] using (2.23)-(2.27), (2.5)-(2.7)
      if T_R(t) < t + k ΔT for T_R(t) in (4.3) then
         t := T_R(t)
         Update U(t) and O(t) by (??)-(??)
         BREAK for-loop
      end if
   end for
until t ≥ T
```

which has a unique solution

$$B_2(t) = e^{\int_0^t (P_{2,2}(u)-1)\mu_2(u)du}\left[\int_0^t e^{\int_0^u (P_{2,2}(v)-1)\mu_2(v)dv}\left(\lambda_2^{(0)}(u) + P_{1,2}(u)\mu_1(u)s_1(u)\right)du + B_2(0)\right].$$

(ii) When Queue 1 is UL and Queue 2 is OL (i.e., $B_1(t) < s_1(t)$, $B_2(t) = s_2(t)$, $Q_2(t) \geq 0$),

$$\begin{aligned} B_1'(t) &= \lambda_1^{(0)}(t) + (P_{1,1}(t) - 1)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)s_2(t), \\ B_2(t) &= s_2(t). \end{aligned}$$

which has a unique solution

$$B_1(t) = e^{\int_0^t (P_{1,1}(u)-1)\mu_1(u)du}\left[\int_0^t e^{\int_0^u (P_{2,1}(v)-1)\mu_1(v)dv}\left(\lambda_1^{(0)}(u) + P_{2,1}(u)\mu_2(u)s_2(u)\right)du + B_1(0)\right].$$

(iii) When both queues are OL,

$$B_1(t) = s_1(t), \quad B_2(t) = s_2(t).$$

(iv) When both queues are UL,

$$\begin{aligned} B_1'(t) &= \lambda_1^{(0)}(t) + (P_{1,1}(t) - 1)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)B_2(t), \\ B_2'(t) &= \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)B_1(t) + (P_{2,2}(t) - 1)\mu_2(t)B_2(t), \end{aligned}$$

or

$$\mathbf{B}'(t) = \lambda^{(0)}(t) + \mathbf{C}(t) \cdot \mathbf{B}(t), \tag{E.1}$$

where

$$\mathbf{C}(t) \equiv \left(\mathbf{P}^T(t) - \mathbf{I}\right)\mathbf{\Gamma}(t) \quad \text{and} \quad \mathbf{\Gamma}(t) \equiv \begin{bmatrix} \mu_1(t) & 0 \\ 0 & \mu_2(t) \end{bmatrix}.$$

After $\mathbf{B}(t)$ is obtained, the aggregated arrival rate

$$\begin{aligned}\lambda_1(t) &= \lambda_1^{(0)}(t) + P_{1,1}(t)\mu_1(t)B_1(t) + P_{2,1}(t)\mu_2(t)B_2(t),\\ \lambda_2(t) &= \lambda_2^{(0)}(t) + P_{1,2}(t)\mu_1(t)B_1(t) + P_{2,2}(t)\mu_2(t)B_2(t).\end{aligned}$$

F. The FPE Algorithm for GI Service

We now present more about the algorithm for GI service distributions. Given a desired duration T of an interval $[0, T]$, the vector of the model data defined as (3.5), a step size $0 < \Delta T \leq T$, and an error tolerance parameter (ETP) $\epsilon > 0$, we summarize the algorithm formally as the following.

Algorithm 4 : An FPE based algorithm for the $(G_t/GI/s_t + GI_t)^m/M_t$ Fluid Network

Initialization: $t := 0$
repeat
  for $k = 0, 1, \ldots, \lceil (T - t)/\Delta T \rceil$ **do**
    for all $i \in \mathcal{O}(t)$ **do**
      - Compute $b_i(s, 0)$ solving FPE (5.3) with ETP ϵ, $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$
      - Let $\sigma_i(s) := b_i(s, 0) - s_i'(s)$
    end for
    Compute $\lambda(s)$ using FPE (5.5) with ETP ϵ, $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$
    Compute $\mathcal{P}(s)$ for $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$ using (2.23)-(2.27), (2.5)-(2.7) or the algorithm from Liu and Whitt (2011a) if the service is GI
    if $T_{\mathcal{R}}(t) < t + k\,\Delta T$ for $T_{\mathcal{R}}(t)$ in (4.3) **then**
      $t := T_{\mathcal{R}}$
      Update $\mathcal{U}(t)$ and $\mathcal{O}(t)$ by (**??**)-(**??**)
      BREAK for-loop
    end if
  end for
until $t \geq T$

G. More for the Examples

G.1. More on the Two-Queue FQNet

We demonstrate how the FPE-based algorithm works. Since it is key to obtain the total arrival rates $\lambda_1(t)$ and $\lambda_2(t)$ for $0 \leq t \leq T$, we first demonstrate how fast the fixed-point algorithm converges. We initially let $\lambda_i^{(1)}$ be $\lambda_i^{(0)}$, $i = 1, 2$. In Figure 6, we plot the total arrival rates in every iteration. The two functions at the bottom are $\lambda_1^{(0)}(t)$ and $\lambda_2^{(0)}(t)$; the

functions at the top are the $\lambda_1(t)$ and $\lambda_2(t)$, computed using the ODE based algorithm; the other functions are the intermediate values computed using the FPE based algorithm. The monotone convergence and geometric rate of convergence are evident from Figure 6.

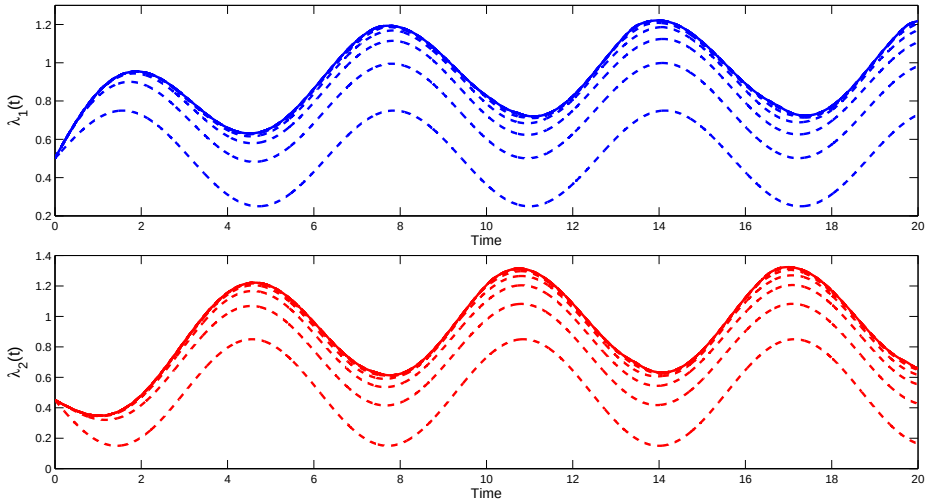

Figure 6: The convergence to the fixed point of the total arrival rate vector: the increasing computed values at each queue in successive iterations.

In Figure 7, we plot all standard performance measures of the fluid network using the FPE based algorithm, including λ_i, Q_i, w_i, B_i, X_i, and $b_i(\cdot, 0)$, $i = 1, 2$.

Figure 8 illustrate how these approximations perform for the same example with $n = 50$. Now the solid lines are simulation estimates of the mean of these scaled stochastic processes, obtained by averaging multiple independent sample paths. For this small value of n, the stochastic variability cannot be simply ignored. But the means can be well approximated by the fluid functions. Figure 1 and 8 show that the fluid approximation is effective in describing the performance of the stochastic system.

G.1.1. Different Phases

Complementing the two-queue example in §6.2, we now consider the same model with different phases in the sinusoidal arrival-rate functions. In particular, let $\phi_1 = 0$ and $\phi_2 = 1$ and let all other model parameters remain the same. As the analogs of Figure 7 and 8 (with $\phi_2 = -3$), we plot the fluid function and perform simulation comparison in Figure 9 and 10 with $\phi_2 = 1$. In this case the two queues become OL and UL almost at the same time.

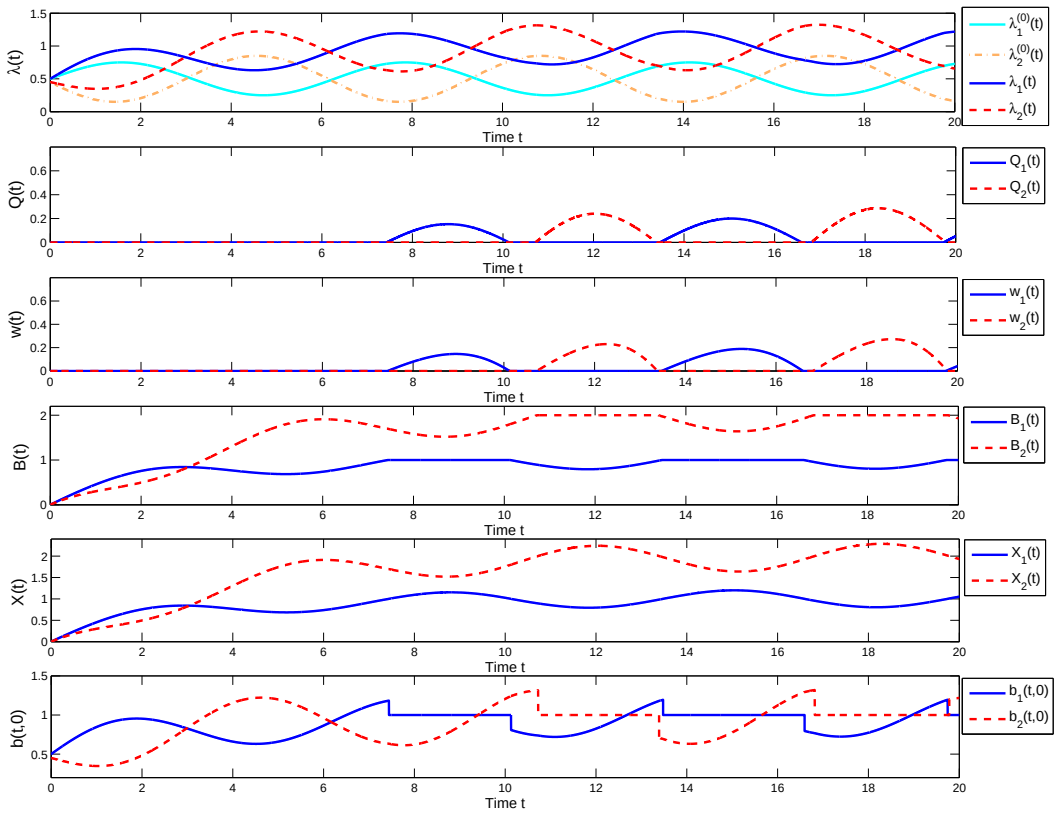

Figure 7: Computing the fluid performance functions for the $(M_t/M/s_t + M)^2/M_t$ network fluid model.

G.1.2. Time-Varying Staffing

The two-queue example in §6.2 had constant staffing functions. To show that the algorithms can also handle the important feature of time-varying staffing, we now consider sinusoidal staffing, using the staffing functions

$$s_i(t) = \alpha_i + \beta_i \cdot \sin(\gamma_i\, t + \psi_i), \quad i = 1, 2, \tag{G.1}$$

with $\alpha_1 = 1$, $\alpha_2 = 2$, $\beta_1 = 0.6\,\alpha_1$, $\beta_2 = 0.5\,\alpha_2$, $\gamma_1 = 1$, $\gamma_2 = 0.5$, $\psi_1 = 3$, $\psi_2 = 0$. While other model parameters remain the same. With these particular choices on the model parameters, it is not hard to see that for $i = 1, 2$,

$$s_i'(t) + \mu_i\, s_i(t) = \beta_i \gamma_i \cos(\gamma_i\, t + \psi_i) + \mu_i\, (\alpha_i + \beta_i \sin(\gamma_i\, t + \psi_i)) \geq 0,$$

or equivalently,

$$\sin\left(\gamma_i\, t + \psi_i + \arctan\left(\frac{\gamma_i}{\mu_i}\right)\right) \geq -\frac{\mu_i \alpha_i}{\sqrt{\mu_i^2 + \gamma_i^2}}, \quad \text{for all} \quad t \geq 0,$$

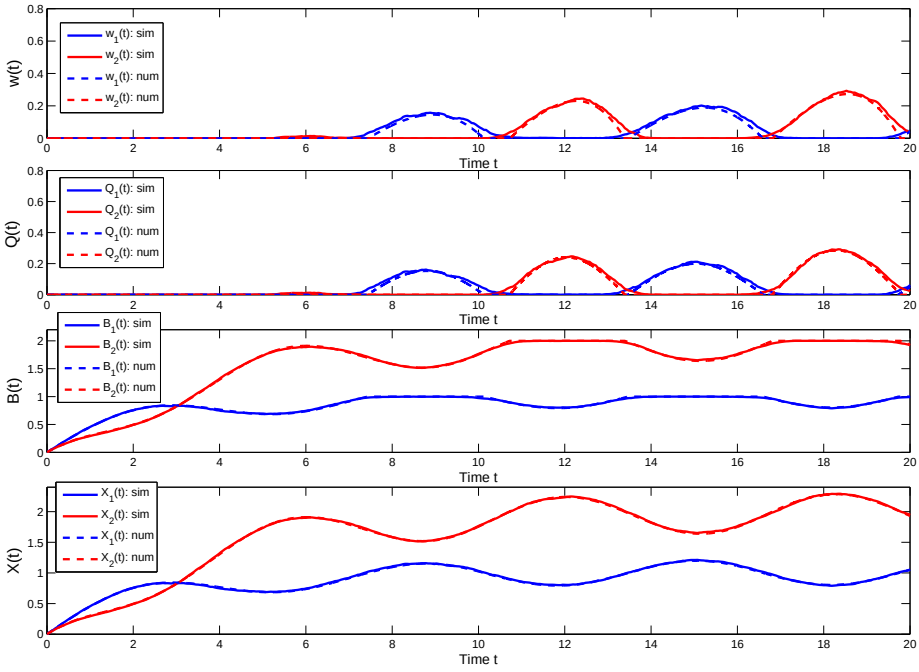

Figure 8: A comparison of performance functions in the $(M_t/M/s_t + M)^2/M_t$ FQNet with simulation estimates of time-varying mean values, obtained by averaging $n = 50$ independent sample paths from the corresponding QNet.

which guarantees the feasibility for both staffing functions s_1 and s_2. See the Appendix I.2.1 of Liu and Whitt (2011a) for more discussion.

As the analogs of Figure 9, we plot the fluid function in Figure 11.

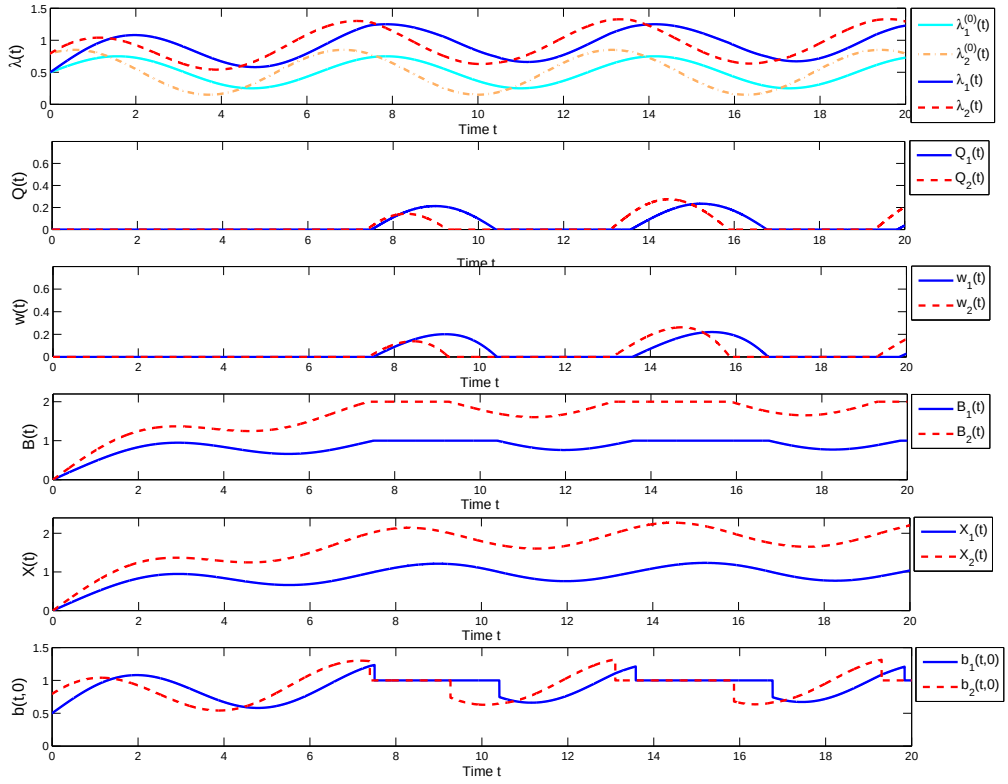

Figure 9: Computing the fluid performance functions for the $(M_t/M/s_t + M)^2/M_t$ network fluid model.

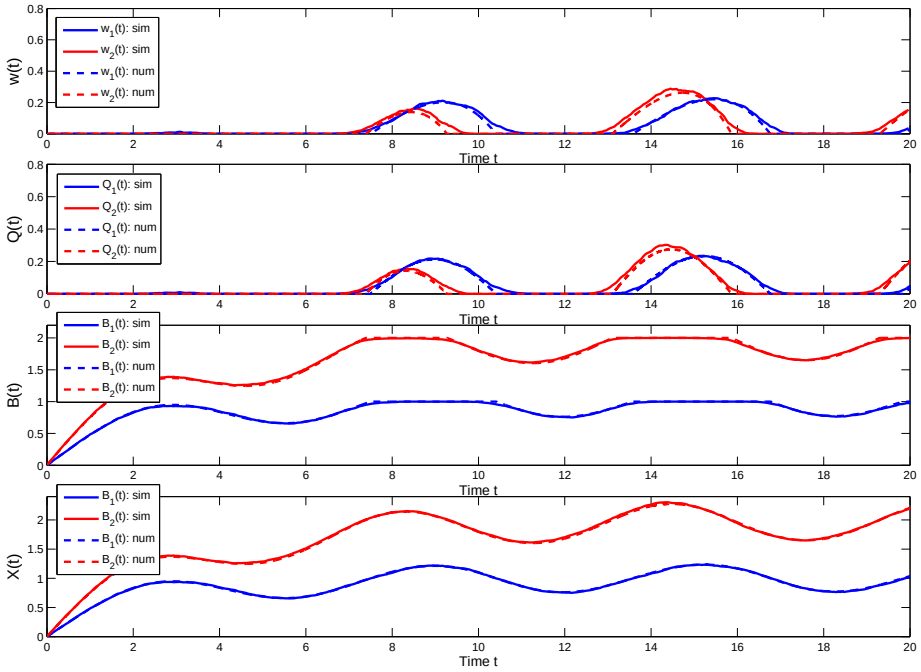

Figure 10: A comparison of the $(M_t/M/s_t + M)^2/M_t$ network fluid model with a simulation run averaging 50 independent sample paths, $n = 100$.

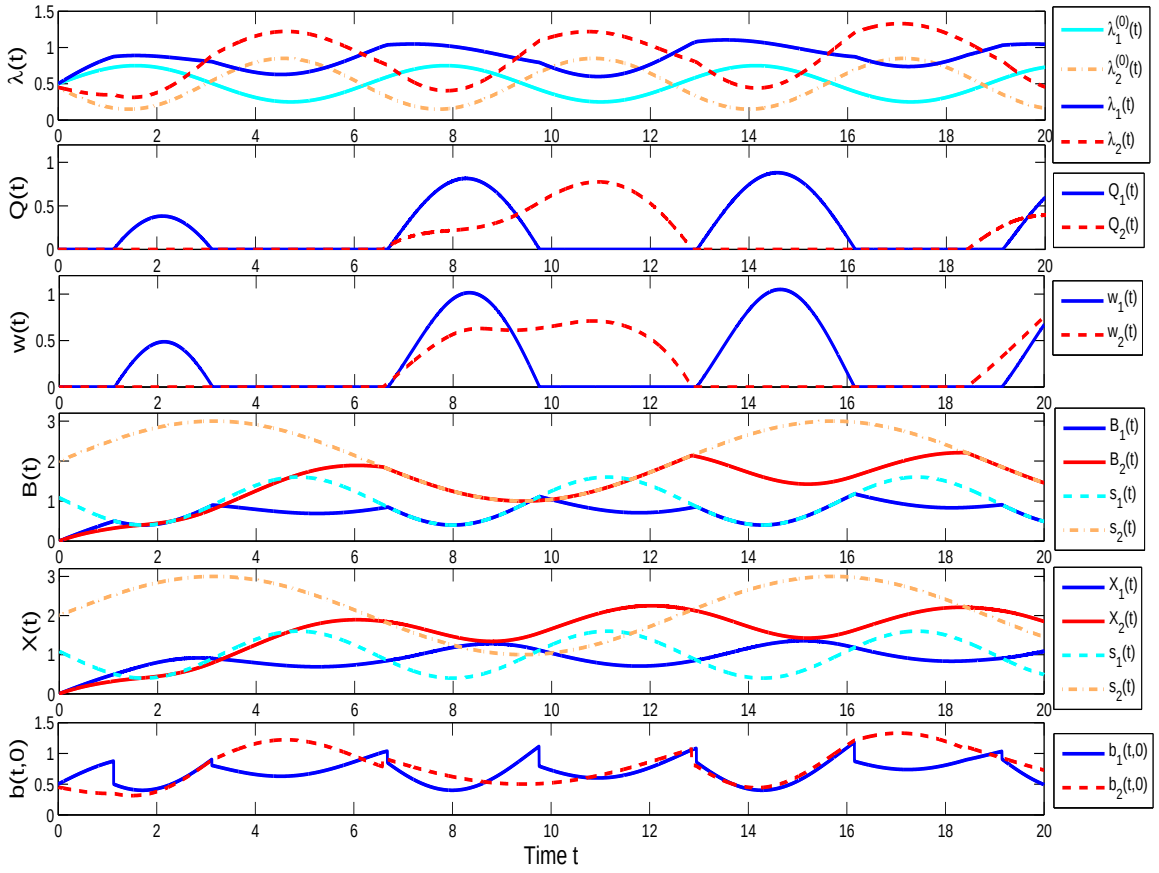

Figure 11: Computing the fluid performance functions for the $(M_t/M/s_t + M)^2/M_t$ network fluid model with sinusoidal staffing functions.

G.2. More on the Many-Queue Example

We now provide more material related to the many-queue example in §6.3.

Complementing the example in §6.3, we let $m = 10$, $\phi_i = 0$, $i = 1, \ldots, 10$. We repeat the experiment and plot the fluid functions for all 10 queues in Figure 12 (which is an analog of Figure 13). Figure 13 shows plots of the performance functions for $m = 10$. The red dashed lines are λ_i, i.e., the total arrival rates which are the solutions to the FPE; the blue solid lines are $\lambda_i^{(0)}$, i.e., the external arrival rates.

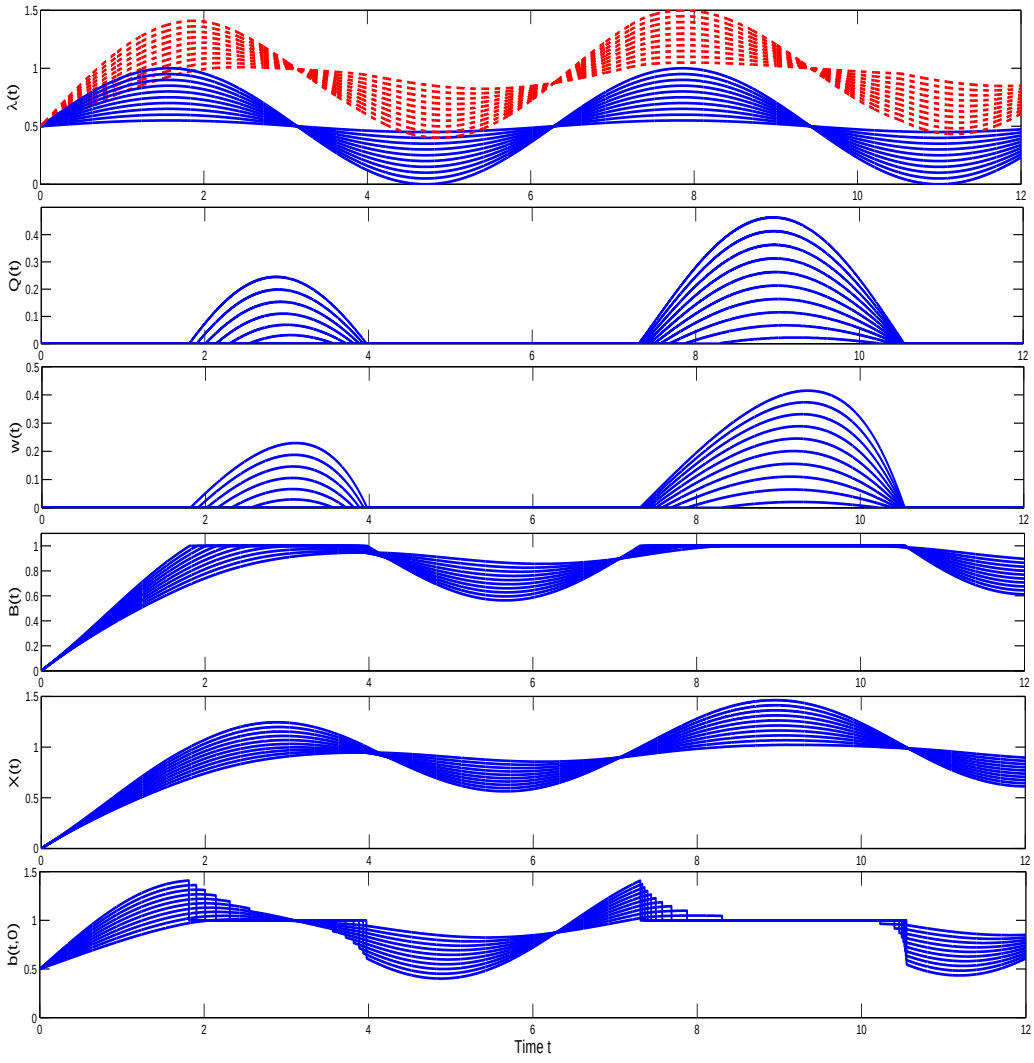

Figure 12: Computing the fluid performance functions for the $(M_t/M/s_t + M)^{10}/M_t$ network fluid model.

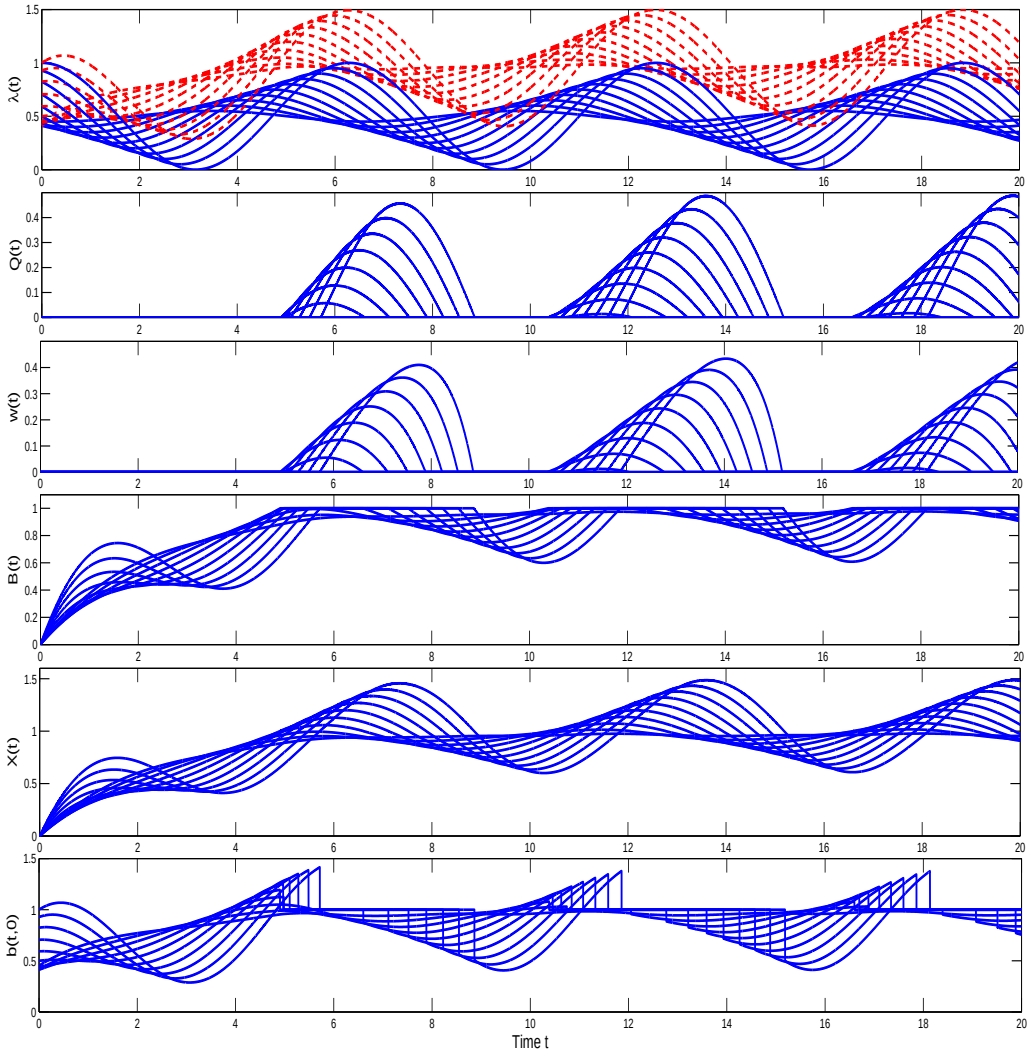

Figure 13: Computing the fluid performance functions for the $(M_t/M/s_t+M)^{10}/M_t$ network fluid model ($\phi_i = \pi(1.5 - i/m)$).

G.3. More on the GI Service Example

Figure 14 compares the results of Alg(FPE,GI) applied to the $(M_t/LN/s + E_2)^2/M$ FQNet with simulation estimates of mean values for the corresponding stochastic processes in the corresponding SQNet with $n = 50$, obtained by averaging the sample paths from 200 independent replications.

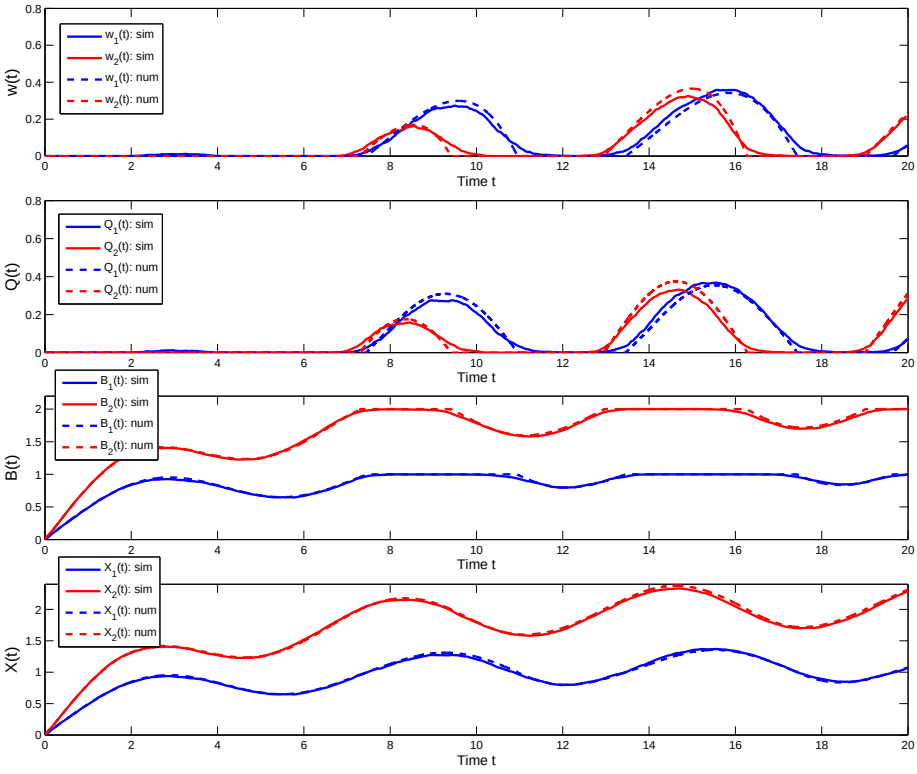

Figure 14: A comparison of the $(M_t/LN/s_t + E_2)^2/M_t$ network fluid model with a simulation run averaging 50 independent sample paths, $n = 100$.