Closed queueing networks in heavy traffic: Fluid limits and efficiency

Sunil Kumar and P.R. Kumar

ABSTRACT We address the behavior of stochastic Markovian closed queueing networks in heavy traffic, i.e., when the population trapped in the network increases to infinity. All service time distributions are assumed to be exponential. We show that the fluid limits of the network can be used to study the asymptotic throughput in the infinite population limit. As applications of this technique, we show the efficiency of all policies in the class of Fluctuation Smoothing Policies for Mean Cycle Time (FSMCT), including in particular the Last Buffer First Serve (LBFS) policy for all reentrant lines, and the Harrison–Wein balanced policy for two station reentrant lines. By "efficiency" we mean that they attain bottleneck throughput in the infinite population limit.

1 Introduction

Consider a closed queueing network with a population size N. Typically, the objective is to schedule such networks to maximize the throughput. In this paper we are interested in the heavy traffic behavior, i.e., as $N \to \infty$, of the throughput. We assume that all service times are exponentially distributed.

Let $\lambda^u(x_n)$ denote the throughput when the closed network is started with the initial condition x_n, and scheduling policy u is employed. The population size is $|x_n|$. Let λ^* denote the maximum throughput sustainable by the network. We say that the scheduling policy u is *efficient* if $\lim_{|x_n|\to\infty} \lambda^u(x_n) = \lambda^*$. Our goal in this paper is to address the efficiency of scheduling policies.

Our approach is via the "fluid limits" of the closed network. These fluid limits (see Dai [4] and Chen and Mandelbaum [2]) are obtained by suitably scaling the queue lengths in the network and applying a functional strong law of large numbers to this scaled network to obtain a limit process. This deterministic limit process, called the fluid limit, is not unique. However, every such limit must necessarily obey a set of integral equations.

It is known that by studying such fluid limits one can study the stability, i.e., positive Harris recurrence, of *open* queueing networks, see Dai [4]. It has been shown by Lu and Kumar [13] that the First Buffer First Serve (FBFS) policy, the Last Buffer First Serve (LBFS) policy and the Least Slack (LS) scheduling policy are all stable for open reentrant lines, under a bursty deterministic model. Using the fluid limit approach, Dai [4] has established that the FBFS policy is stable for stochastic reentrant lines, while Dai and Weiss [3] and Kumar and Kumar [11] have established the stability of the LBFS policy. Finally, the stability of all Fluctuation Smoothing Policies for Mean Cycle Time (FSMCT), a special subset of LS policies has been established in Kumar and Kumar [12]. Note that the LBFS policy is a special case of such an FSMCT policy. Such policies have been shown to perform well in the simulation studies of [14] for closed reentrant lines.

In this paper we employ the fluid limit approach for *closed* queueing networks. By studying the integral equations, we can determine a superset of (i.e., a larger set containing) the fluid limits. We relate the throughputs of the fluid limits to the asymptotic throughput of the reentrant line. We thereby obtain sufficient conditions for the reentrant line to be efficient, i.e., to achieve the maximal achievable throughput in the infinite population limit.

We utilize this approach to establish the efficiency of certain specific scheduling policies for reentrant lines. Even though there is no notion of "first" or "last buffer" in a closed reentrant line, let us designate some arbitrary buffer as the "last buffer." Consider any resulting FSMCT policy. We establish that all such FSMCT policies are efficient. Lest it be regarded that all policies which are stable for open networks are also efficient for closed networks, we note that the FBFS policy is inefficient for a closed network, as shown in Harrison and Nguyen [7], even though it is stable for open networks.

For *two-station* closed networks, by studying the reflected Brownian motion approximation of the network, Harrison and Wein [8] have devised a buffer priority policy which they conjecture is asymptotically optimal in heavy traffic. We establish the efficiency of this Harrison–Wein policy for two-station closed reentrant lines. Such a result for general two-station closed networks has been shown earlier by Jin, Ou and Kumar [9] using the very different method of functional bounding by linear programs.

In the next section we describe the model of the closed queueing network, and also describe the special case of a closed reentrant line. In Section 3 we discuss the existence of fluid limits, and specify the set of integral equations that these limits satisfy. In Section 4 we define the notion of efficiency, and obtain conditions on the fluid limits which guarantee it. In Section 5 we refine the limit for buffer priority policies. In Section 6 we establish the efficiency of all FSMCT policies, regardless of network topology, for closed reentrant lines. In Section 7 we establish the efficiency of the Harrison–Wein policy for two-station reentrant lines. Finally, we provide some examples of networks with inefficient fluid flows in Section 8.

2 The stochastic closed network model

The network we consider consists of S servers labeled $\{1, 2, \ldots, S\}$. There are L buffers labelled $b_1, b_2, \ldots, b_L$. Customers at buffer b_i require exponentially distributed service time with mean $m_i = \frac{1}{\mu_i}$ from server $\sigma(i) \in \{1, 2, \ldots, S\}$. On completing service at buffer b_i, customers move to buffer b_j with probability p_{ij}. The routing matrix $P = [p_{ij}]$ is stochastic and irreducible. Hence if the initial population size is $|x_n| = N$, then these N customers are trapped in the system. Let $\pi = \pi P$ be the unique invariant probability measure associated with P.

A special case of the above model is the *closed reentrant line*, which may be described as follows. Customers begin processing at buffer b_1, located at server $\sigma(1) \in \{1, \ldots, S\}$. Upon completing service, they proceed to buffer b_2 located at server $\sigma(2) \in \{1, \ldots, S\}$. Let b_L at server $\sigma(L)$ be the last buffer visited. The sequence $\{\sigma(1), \ldots, \sigma(L)\}$ is the *route* of the customer. At the same time that a customer completes service at buffer b_L, a new customer is released into buffer b_1. This is what is termed as a "closed loop" release policy in manufacturing [18, 16], and a "window" based admission policy in communication networks [17, 1]. The end result is a closed network with routing matrix given by $P = [p_{ij}]$, where $p_{ij} = 1$ if $j = (i + 1) \bmod L$ and $= 0$ otherwise. Note that in this case, $\pi_i = 1/L$, for all i.

In a closed network, one cannot really talk of an exit or entry point, but, for convenience, we will continue to call b_L as the "last buffer." The goal is to maximize the "throughput" of the system. For reentrant lines it is immediate that the throughput can be taken to be the rate of departures from buffer b_L. In the case of the more general network, following [9]we define a "normalized"throughput as follows. Consider a scheduling policy u. Under u, let

$$\begin{aligned} w_i(t) &:= 1 \text{ if } \sigma(i) \text{ is working on customer in } b_i \text{ at } t, \\ &:= 0 \text{ otherwise.} \end{aligned}$$

Let $\beta_i := E[w_i(t)]$. Now

$$\sum_j \mu_i \beta_i p_{ij} = \sum_j \mu_j \beta_j p_{ji}.$$

But π is the unique invariant measure of P. Therefore, there exists an $\alpha > 0$ such that

$$\alpha = \frac{\beta_i \mu_i}{\pi_i}.$$

We call α the *normalized throughput*, and define the (unnormalized) *throughput* as $\lambda^u(x_n) := \pi_L \alpha = \mu_L \beta_L$.

For each server $\sigma \in \{1, 2, \ldots, S\}$, define

$$\rho_\sigma := \left(\sum_{\{1 \le k \le L,\ \sigma(k)=\sigma\}} \pi_k m_k \right), \tag{1.1}$$

and

$$\alpha^* := \min_{\sigma \in \{1,2,\ldots,S\}} \left(\sum_{\{1 \le k \le L,\ \sigma(k)=\sigma\}} \pi_k m_k \right)^{-1}. \tag{1.2}$$

We call ρ_σ the *nominal normalized load* on server σ, and α^* the *normalized bottleneck throughput.* We define the (unnormalized) bottleneck throughput of the network as

$$\lambda^* := \pi_L * \alpha^*. \tag{1.3}$$

We note that the long term rate of departures from b_L can never exceed λ^*. Any server σ achieving the "min" on the RHS of (1.2) is thus a bottleneck.

We assume that the scheduling policy employed is *non–idling*; that is, a server cannot be idle when any of its buffers is non–empty, and *stationary*; that is, the policy depends only on the current buffer lengths. In the following, we will mainly restrict attention to buffer priority policies which are nonidling and stationary. If a priority policy is used, we assume that it is preemptive resume. By this we mean that the service of a customer is interrupted to serve a higher priority customer whenever any such customer is present at the station, and is resumed to work on the remaining portion of the service time whenever there is no such higher priority customer.

3 The fluid limits

In this section, we define the fluid limits and describe the integral equations which they must satisfy. With the sole exception that the network is closed rather than open, all the results in this section are the same as in [4]. First we introduce the notation and some preliminaries. Let $Q_k(t)$ denote the queue length, and $v_k(t)$ the residual service time at buffer b_k at time t. Let $D_k(t)$ denote the number of departures from buffer b_k in $[0, t]$. Let $\Phi_l^k(D_k(t))$ denote the numbers of these departures in $[0, t]$ which were routed from b_k to b_l. Then $A_k(t)$, the total number of arrivals to buffer b_k in $[0, t]$, including the initial condition, is given by:

$$A_k(t) = Q_k(0) + \sum_{l=1}^{L} \Phi_k^l(D_l(t)),\ k = 1, \ldots, L.$$

Also, we have

$$Q_k(t) = A_k(t) - D_k(t).$$

Let $T_k(t)$ be the amount of time in $[0,t]$ that server $\sigma(k)$ spends working on b_k, and let $B_\sigma(t) := \sum_{\{j:1\le j\le L,\text{ and } \sigma(j)=\sigma\}} T_j(t)$ be the amount of time server σ was busy in $[0,t]$. Let $I_\sigma(t) := t - B_\sigma(t)$ denote the idle time of σ. Because of the non–idling assumption, we have

$$\int_0^\infty \left[\left(\sum_{\{j:\,1\le j\le L \text{ and } \sigma(j)=\sigma\}} Q_j(t) \right) \wedge 1 \right] dI_\sigma(t) = 0. \tag{1.4}$$

The state of the system is given by

$$X(t) := (Q_1(t),\dots,Q_L(t), v_1(t),\dots,v_L(t)).$$

For $x = (q_1,...,q_L,v_1,...,v_L)$ we define $|x| := \sum_{k=1}^L (q_k + v_k)$. In the sequel, we shall denote explicit dependence on the initial condition x of $X(t)$ by adding a superscript x to the variable of interest. For any function f, let

$$\bar{f}^x(t) := \frac{1}{|x|} f^x(|x|t), \text{ for } t \ge 0 \text{ and } x > 0,$$

denote its scaled version. This is the so–called "fluid scaling."

For simplicity, we make the additional assumption that all the service time distributions are exponential. This allows us to ignore the effects of v_k. The state $x(t)$ is then redefined to consists only of the queue lengths $(Q_1(t),\dots,Q_L(t))$, and $|Q(t)| := \sum_{i=1}^L Q_i(t) \equiv N$.

For each initial condition x_n, we consider the resulting vector process $(\bar{D}^{x_n}, \bar{A}^{x_n}, \bar{Q}^{x_n}, \bar{T}^{x_n}, \bar{I}^{x_n})$ as an element of $\mathcal{D}^d_{\mathcal{R}}[0,\infty)$, the space of $\mathcal{R}^d$-valued right continuous paths with left limits, endowed with the Skorohod topology. Thus we can define the countable collection of $\mathcal{D}^d_{\mathcal{R}}$-valued variables (one for each x_n) on a common probability space. This is the probability space with respect to which all the results in the sequel are stated.

First, we state a result which is about the same as that of Dai [4]. It is the closed network version of Theorem 4.4 given in [4] for open networks.

Theorem 3.1 Almost surely*, the following holds. For every sequence of initial conditions with* $|x_n| \to \infty$*, there exists a further subsequence* x_{n_l} *such that along this subsequence, as* $l \to \infty$*,*

$$(\bar{D}^{x_{n_l}}, \bar{A}^{x_{n_l}}, \bar{Q}^{x_{n_l}}, \bar{T}^{x_{n_l}}, \bar{I}^{x_{n_l}}) \Rightarrow (\bar{D}, \bar{A}, \bar{Q}, \bar{T}, \bar{I}), \tag{1.5}$$

where "$\Rightarrow$" denotes uniform convergence on compacts. Furthermore, the limit processes, called the fluid limits*, satisfy:*

$$\bar{D}_k(t) = \mu_k \bar{T}(t), \tag{1.6}$$

$$\bar{A}_k(t) = q_k + \sum_{l=1}^L p_{lk} \bar{D}_l(t), \tag{1.7}$$

$$\bar{Q}_k(t) = \bar{A}_k(t) - \bar{D}_k(t), \tag{1.8}$$

$$\bar{I}_\sigma(t) = t - \sum_{\{j:1\le j\le L \text{ and } \sigma(j)=\sigma\}} \bar{T}_j(t), \tag{1.9}$$

$$0 \le \bar{T}_k(t_1) \le \bar{T}_k(t_2) \text{ and } 0 \le \bar{I}_\sigma(t_1) \le \bar{I}_\sigma(t_2) \text{ if } t_1 \le t_2, \tag{1.10}$$

$$|\bar{I}_\sigma(t) - \bar{I}_\sigma(s)| \le |t-s|,\ |\bar{T}_k(t) - \bar{T}_k(s)| \le |t-s|, \text{ and } \bar{Q}_k(t) \text{ is also Lipschitz}, \tag{1.11}$$

$$\int_0^\infty \left(\sum_{\{j:1\le j\le L \text{ and } \sigma(j)=\sigma\}} \bar{Q}_j(t) \wedge 1 \right) d\bar{I}_\sigma(t) = 0, \tag{1.12}$$

$$\text{and } \sum_{k=1}^{L} q_k = 1. \tag{1.13}$$

Proof. The equality (1.6) follows from the renewal theorem while (1.7) follows from the classical functional strong law of large numbers, see also Lemma 4.2 of [4]. We use Lemma 4.1 of [4] to conclude uniform convergence on compacts. Since

$$|\bar{T}_k^{x_n}(t) - \bar{T}_k^{x_n}(s)| \le |t-s|, \text{ for all } x_n,$$

$\{\bar{T}_k^{x_n}\}$ is relatively compact in $\mathcal{D}_{\mathcal{R}}[0,\infty)$; see Ethier and Kurtz [6], Theorem 6.3, p.123. Therefore there is a convergent subsequence in the Skorohod topology. Also, the Lipschitz property guarantees the continuity of the limit. Since the limit is continuous, it follows that the convergence is uniform on compacts, see [6]. The other quantities, namely $\bar{A}_k^{x_n}(t)$, $\bar{Q}_k^{x_n}(t)$ and $\bar{I}_\sigma^{x_n}(t)$, depend on $\bar{D}_k^{x_n}(t)$ and $\bar{T}_k^{x_n}(t)$ in an affine fashion, and hence contain convergent subsequences by the continuous mapping theorem.

The equations (1.8–1.10) follow from the fact that they hold for every sample path of the scaled processes. The result (1.11) is also immediate. The result (1.13) follows from the scaling by $|x_n|$. Only (1.12) needs justification. For this, note that $\left(\sum_{\{j:1\le j\le L \text{ and } \sigma(j)=\sigma\}} \bar{Q}_j^{x_n}(t) \wedge 1\right)$ is continuous in $\bar{Q}_j^{x_n}(t)$, and $\bar{I}_\sigma^{x_n}(t)$ is continuous and nondecreasing. So by Lemma 2.4 of Dai and Williams [5],

$$\int_0^t \left(\sum_{\{j:1\le j\le L \text{ and } \sigma(j)=\sigma\}} \bar{Q}_j^{x_{n_l}}(s) \wedge 1 \right) d\bar{I}_\sigma^{x_{n_l}}(s) \to \int_0^t \left(\sum_{\{j:1\le j\le L \text{ and } \sigma(j)=\sigma\}} \bar{Q}_j(s) \wedge 1 \right) d\bar{I}_\sigma(s),$$

for all t. This and (1.4) yield (1.12). $\square$.

It should be noted that the derivatives of the processes $\bar{A}_i$, $\bar{Q}_i$, $\bar{D}_i$ and

$\bar{I}_\sigma$ exist at a.e. t. Such times of differentiablity are called "regular times," as in Dai and Weiss [3].

4 Fluid limit sufficient condition for efficiency

In this section we define the notion of "efficiency" and provide sufficient conditions for it in terms of the throughput of the fluid limit.

Given a stationary scheduling policy u and an initial state x, we note that the throughput $\lambda^u(x)$ can be expressed as

$$\lambda^u(x) := \lim_{T\to\infty} \frac{E\left[D_L^x(T)\right]}{T}.$$

Definition The stationary scheduling policy u[1] is said to be *efficient* if for every sequence of initial conditions $\{x_n\}$ with $|x_n| \uparrow \infty$, we have

$$\lim_n \lambda^u(x_n) = \lambda^* \tag{1.14}$$

where λ^* is defined in (1.3).

The next lemma provides a sufficient condition for efficiency based on calculating the limits in the opposite order to (1.14).

Lemma 4.1 *If for every sequence x_n with $|x_n| \uparrow \infty$,*

$$\limsup_{T\to\infty} \limsup_n \frac{D_L^{x_n}(|x_n|T)}{|x_n|T} \geq \lambda^* \text{ a.s.}, \tag{1.15}$$

then (1.14) also holds for every such sequence.

Proof. Consider a sequence x_n with $|x_n| \uparrow \infty$. For every x_n, let x'_n be a state with the same population size, and lying in a closed communicating class with the least throughput, i.e.,

$$|x_n| = |x'_n| \text{ and } \lambda^u(x'_n) = \min_{\{y_n:\, |y_n|=|x_n|\}} \lambda^u(y_n). \tag{1.16}$$

Since x'_n is in a closed communicating class,

$$\lim_{T\to\infty} \frac{D_L^{x'_n}(T)}{T} = \lambda^u(x'_n) \text{ a.s.} \tag{1.17}$$

Fix ω a sample point in our probability space, outside of a set of zero measure excluded in (1.15) above. Let $D_L^{x'_n}$ be the sequence of $\mathcal{D}_{\mathcal{R}}$-valued

[1] More precisely, there is a sequence of stationary, non-idling scheduling policies, one for each $|x_n| = N$.

variables corresponding to ω. Then, from (1.15) above applied to the sequence x'_n, given any x'_n, T and $\epsilon > 0$, we can find x'_{n_l} with $|x'_{n_l}| \geq |x'_n|$ and $T' \geq T$ such that,

$$\frac{D_L^{x'_{n_l}}(|x'_{n_l}|T')}{|x'_{n_l}|T'} \geq \lambda^* - \epsilon.$$

So we have

$$\limsup_{x'_n} \limsup_{T\to\infty} \frac{D_L^{x'_n}(T)}{T} \geq \lambda^* \text{ a.s.}$$

¿From (1.17), we thus obtain

$$\limsup_{x'_n} \lambda^u(x'_n) \geq \lambda^* \text{ a.s.}$$

Then from (1.16) we have, for *every* sequence x_n with $|x_n| \uparrow \infty$,

$$\limsup_{x_n} \lambda^u(x_n) \geq \lambda^*. \tag{1.18}$$

Hence

$$\liminf_{x_n} \lambda^u(x_n) \geq \lambda^*, \tag{1.19}$$

and the result (1.14) follows since $\lambda^u(x) \leq \lambda^*$ for every x. □

We are now ready to prove the main sufficient condition for efficiency, which is that all the fluid limits have maximal throughput.

Theorem 4.2 *Suppose that for every fluid limit* $\bar{D}_L(t)$

$$\limsup_{t\to\infty} \frac{\bar{D}_L(t)}{t} \geq \lambda^* \textit{ a.s.} \tag{1.20}$$

Then (1.15) holds, and hence the stationary non-idling scheduling policy u *is efficient.*

Proof. Fix ω a sample point in our probability space, outside of the set of zero measure excluded in Lemma 4.1. Let $D_L^{x_n}$, $|x_n| \uparrow \infty$, be the sequence of $\mathcal{D}_{\mathcal{R}}$-valued variables corresponding to ω. For brevity we will omit the explicit dependence on ω. Let $\bar{D}_L^{x_{n_k}}$ be any convergent subsequence, converging to $\bar{D}_L$. Fix $\epsilon > 0$ arbitrary. Then from (1.20) for every T, $\exists\, T < t_1 < \infty$ such that,

$$\frac{\bar{D}_L(t_1)}{t_1} \geq \lambda^* - \epsilon.$$

Moreover, from Theorem 3.1 for every t_1, $\exists K(t_1, \epsilon)$ such that for all $k > K$,

$$\left| \frac{D_L^{x_{n_k}}(|x_{n_k}|t_1)}{|x_{n_k}|} - \bar{D}_L(t_1) \right| < \epsilon.$$

Hence,

$$\frac{D_L^{x_{n_k}}(|x_{n_k}|t_1)}{|x_{n_k}|t_1} \geq \lambda^* - \epsilon(1+\frac{1}{t_1}) \text{ for all } k > K(t_1,\epsilon). \tag{1.21}$$

So,

$$\liminf_{x_{n_k}} \frac{D_L^{x_{n_k}}(|x_{n_k}|t_1)}{|x_{n_k}|t_1} \geq \lambda^* - \epsilon(1+\frac{1}{t_1}).$$

Since for every T, $\exists\, t_1 > T$ for which this holds, we have

$$\limsup_{t\to\infty} \liminf_{x_{n_k}} \frac{D_L^{x_{n_k}}(|x_{n_k}|t)}{|x_{n_k}|t} \geq \lambda^* - \epsilon.$$

Since ϵ was arbitrary, we have

$$\limsup_{t\to\infty} \liminf_{x_{n_k}} \frac{D_L^{x_{n_k}}(|x_{n_k}|t)}{|x_{n_k}|t} \geq \lambda^*.$$

The above holds for almost all ω, and hence the result follows. □

5 Buffer priority policies

Consider a buffer priority policy u. Note that it is stationary and non-idling, and also, as shown in Jin, Ou and Kumar [9], there is a single communicating class.

We use the notation "$b_k \succ b_l$" if $\sigma(k) = \sigma(l)$, $k \neq l$, and b_k has higher priority than b_l. Let $H_k := \{1 \leq j \leq L \mid b_j \succ b_k\} \cup \{k\}$ denote the set of buffers having at least as high a priority as b_k, and let

$$U_k(t) := t - \sum_{j\in H_k} T_j(t),$$

denote the time *not* spent working on them respectively. The following lemma identifies additional constraints satisfied by the fluid limits of the buffer priority policy u.

Lemma 5.1 *For every sequence of initial conditions x_n with $|x_n| \uparrow \infty$, there exists a subsequence x_{n_l} such that $\bar{U}_k^{x_{n_l}}(t) \to \bar{U}_k(t)$ in the sense of Theorem 3.1, i.e., almost sure convergence uniformly on compacts. Furthermore, the fluid limits satisfy the additional constraints:*

$$\bar{U}_k(t) = t - \sum_{j\in H_k} \bar{T}_j(t), \tag{1.22}$$

$$0 \leq \bar{T}_k(t) \leq \bar{U}_j(t) \leq t \text{ for all } j \in H_k,\ j \neq k \tag{1.23}$$

$$0 \leq \bar{U}_k(t_1) \leq \bar{U}_k(t_2) \text{ if } t_1 \leq t_2, \tag{1.24}$$

$$\int_0^\infty \left(\sum_{j\in H_k} \bar{Q}_j(t) \wedge 1 \right) d\bar{U}_k(t) = 0, \tag{1.25}$$

Proof. Identical to that of Theorem 3.1. □

Now we can provide a sufficient condition based on Theorem 4.2 which depends only on the solutions to the constraints (1.6–1.13) and (1.22–1.25).

Theorem 5.2 *(i) Suppose every solution $\bar{D}_L(t)$ of (1.6–1.13, 1.22–1.25) satisfies*

$$\limsup_{t\to\infty} \frac{\bar{D}_L(t)}{t} \geq \lambda^*. \tag{1.26}$$

Then the buffer priority scheduling policy u is efficient.

(ii) If for every solution $\bar{D}_L(t)$ of (1.6–1.13, 1.22–1.25), there exists a time $T < \infty$ such that

$$\frac{d}{dt}\bar{D}_L(t) \geq \lambda^* \text{ for almost every } t > T, \tag{1.27}$$

then the buffer priority scheduling policy u is efficient.

Proof. The result (i) is immediate from Theorem 4.2 using the fact that the fluid limits almost surely satisfy (1.6–1.13, 1.22–1.25). The result (ii) follows by noting that (1.27) implies (1.26). □

We now provide a sufficient condition for (1.26), in terms of the idleness process at the bottleneck machine.

Lemma 5.3 *If*

$$\bar{I}_\sigma(t) = 0 \text{ for all } t, \text{ for some } \sigma \in \{1, 2, ...S\}, \tag{1.28}$$

for a solution of (1.6–1.13, 1.22–1.25), i.e., if one of the servers never idles in the fluid limit, then the solution satisfies (1.26).

Proof. Suppose $\bar{I}_1(t) = 0$ for all t. Then

$$\sum_{\{k:\sigma(k)=1\}} m_k \bar{D}_k(T) = T \text{ for all } T \geq 0. \tag{1.29}$$

Now assume (1.26) does not hold, that is,

$$\liminf_{t\to\infty} \frac{\bar{D}_L(t)}{t} < \lambda^*.$$

Suppose that $\{t_l\}$ is a subsequence of times at which

$$\lim_{l\to\infty} \frac{\bar{D}_L(t_l)}{t_l} = \liminf_{t\to\infty} \frac{\bar{D}_L(t)}{t}.$$

We can assume (by taking further subsequences if necessary) that all the limits $\lim_{l\to\infty} \frac{\bar{D}_k(t_l)}{t_l}$ for $k = 1, 2, ..., L$ exist along the subsequence $\{t_l\}$.

Note that $\sum_{j=1}^{L} \bar{Q}_j(t) \le 1$ for all $t \ge 0$. From (1.7) we have

$$\left| \sum_{k=1}^{L} \bar{D}_k(t_l) p_{kj} - \sum_{k=1}^{L} \bar{D}_j(t_l) p_{jk} \right| \le 1.$$

So we have

$$\sum_{k=1}^{L} p_{kj} \lim_{l\to\infty} \frac{\bar{D}_k(t_l)}{t_l} = \lim_{l\to\infty} \frac{\bar{D}_j(t_l)}{t_l} \sum_{k=1}^{L} p_{jk}.$$

But π is the unique invariant probability measure associated with P. So we must have

$$\lim_{l\to\infty} \frac{\bar{D}_k(t_l)}{t_l} = \pi_k \alpha \text{ for some constant } \alpha > 0, \text{ for all } k. \tag{1.30}$$

Since $\pi_L \alpha < \lambda^*$, we have $\alpha < \alpha^*$. ¿From (1.30), for a fixed $\delta > 0$ sufficiently small, for any T, we can find a $t_l \ge T$ such that

$$\bar{D}_k(t_l) < (\pi_k \alpha^* - \delta) t_l, \text{ for all } k = 1, 2, ..., L.$$

Thus, from (1.29),

$$t_l = \sum_{\{k:\sigma(k)=1\}} m_k \bar{D}_k(t_l) \le \left(\sum_{\{k:\sigma(k)=1\}} m_k \pi_k \right) [(\alpha^* - L\delta) t_l].$$

But $\sum_{\{k:\sigma(k)=1\}} m_k \pi_k \alpha^* \le 1$, leading to a contradiction. Thus (1.26) must hold. □

6 Efficiency of all FSMCT policies

In this section, we utilize Theorem 5.2(ii) to establish that all policies in the class of Fluctuation Smoothing Policies for Mean Cycle Time (FSMCT) are efficient for all closed reentrant lines, regardless of network topology. Since the Last Buffer First Serve (LBFS) policy is a member of this class of FSMCT policies, its efficiency is also thus established.

First, as described in Section 2, we designate some buffer b_L as the last buffer. The FSMCT policies can be described as follows, see [15]. At any given instant t, machine σ should work on the first part in that non–empty buffer b_k for which

$$k = \arg\min_{j} \sum_{i=j+1}^{L} (Q_i(t) - \xi_i). \tag{1.31}$$

Here, $\xi_i \in \mathcal{R}$, for $i = 1, 2, ..., L$, is intended to be an estimate of the mean number of parts in b_i, usually chosen by some empirical method. This policy attempts to regulate the downstream shortfall and by doing so, attempts to reduce fluctuations. It has been shown to perform well in simulation studies of reentrant lines[14].

However, we shall let $\{\xi_i\}$ be arbitrary real numbers, subject only to the assumption that the partials sums $\sum_{i=j+1}^{L} \xi_i$ are unique with unique fractional parts for each j. This can always be done by changing the fractional parts of the ξ_i's by arbitrarily small amounts. This also means that the minimizer in (1.31) above is unique. One should note that if a different buffer had been chosen as the "last" buffer, then one obtains a different policy. The following result applies to *all* the resulting FSMCT policies from any choice of the last buffer b_L, and any choice of ξ_i's.

The LBFS scheduling policy functions as follows. If buffers b_j and b_k share the same server, i.e., $\sigma(j) = \sigma(k)$, and $j < k$, then priority is given to b_k. Therefore if $k < L$, b_k can be worked on by $\sigma(k)$ only if $b_j = 0$ for all $j = k+1, \ldots, L$, with $\sigma(j) = \sigma(k)$. Note that b_L is never preempted. Note that this corresponds to the case when all the ξ_i above have been chosen negative in the FSMCT policy. Hence the LBFS policy is a special case of FSMCT policies.

Theorem 6.1 *Every FSMCT policy is efficient.*

Proof. We show in Lemma 6.2 below that the integral equations describing the fluid limits of the queueing network under any FSMCT policy are identical to those describing the LBFS policy, along the lines of Kumar and Kumar [12]. Then we establish in Lemma 6.3 that (1.27) holds for LBFS policies. We thus conclude the validity of the result using Theorem 5.2(ii). □

Lemma 6.2 *The fluid limits of any FSMCT policy obey the integral equations (1.6-1.13) and (1.22-1.25) with the H_k denoting the set of buffers which are downstream of b_k, i.e.,*

$$H_k := \{b_j \mid k \leq j \leq L\}.$$

This means that the integral equations describing the fluid limits of the queueing network under any FSCMCT policy are identical to those describing the LBFS policy.

Proof. The proof is based on Theorem 2 of Kumar and Kumar [12]. Note that we need only prove (1.25). This is the nontrivial assertion which proves that the fluid limits of all FSMCT policies are identical those of LBFS. Define $Q(t) := (Q_1(t), \ldots, Q_L(t))$, $\xi := (\xi_1, \ldots, \xi_L)$, and

$$f_k(Q(t), \xi) := \min(1, g_k(Q(t), \xi))$$

where

$$g_k(Q(t), \xi) \quad := \quad \sum_{\{j:\sigma(j)=\sigma(k)\}} Q_j(t) \text{ if } k = \min\{j : \sigma(j) = \sigma(k)\}$$

$$:= \sum_{\substack{\{j\geq k \ \& \\ \sigma(j)=\sigma(k)\}}} Q_j(t) \prod_{\substack{\{n<j \ \& \\ \sigma(n)=\sigma(k)\}}} \left(\sum_{i=n+1}^{j} Q_i(t) - \xi_i \right)^+ \quad \text{otherwise.}$$

Now it follows from (1.31) that $f_k(Q(t), \xi) > 0$ implies that the highest priority non-empty buffer at $\sigma(k)$ at time t, say b_j, is such that $j \geq k$. Thus the policy and the non–idling assumption require that machine $\sigma(k)$ work on some buffer in the set $\{b_j : j \geq k \text{ and } \sigma(j) = \sigma(k)\}$. So if we define $I_k(t) := t - \sum_{j\geq k} T_j(t)$, the amount of time in $[0, t]$ not spent on buffers $\{b_j, \ j \geq k\}$, we must have

$$\int_0^\infty f_k(Q(s), \xi)\, dI_k(s) = 0 \text{ for } k = 1, ..., L. \tag{1.32}$$

Note that from (1.32), we have for trajectories starting from initial conditions x_{n_l},

$$\int_0^\infty f_k(\bar{Q}^{x_{n_l}}(s), \bar{\xi}^{x_{n_l}})\, d\bar{I}_k^{x_{n_l}}(s) = 0, \text{ for all } x_{n_l}, \tag{1.33}$$

where, by $\bar{\xi}^x$, we mean $\frac{\xi}{|x|}$. Now, f_k is a continuous bounded function on $\mathcal{R}^{2L}$. Also, $(\bar{Q}^{x_{n_l}}, \bar{\xi}^{x_{n_l}}) \Rightarrow (\bar{Q}, 0)$ in $D_{\mathcal{R}^{2L}}[0, \infty)$, $\bar{I}_k^{x_{n_l}} \Rightarrow \bar{I}_k$ in $C_{\mathcal{R}}[0, \infty)$, and $\bar{I}_k^{x_{n_l}}$ is non–decreasing for each l. We can use Lemma 2.4 of Dai and Williams [5] to conclude that uniformly for all t in any compact subset of $\mathcal{R}$,

$$\int_0^t f_k(\bar{Q}^{x_{n_l}}(s), \bar{\xi}^{x_{n_l}})\, d\bar{I}_k^{x_{n_l}}(s) \rightarrow \int_0^t f_k(\bar{Q}(s), 0)\, d\bar{I}_k(s).$$

¿From the above and (1.33), we obtain

$$\int_0^\infty f_k(\bar{Q}(s), 0)\, d\bar{I}_k(s) = 0. \tag{1.34}$$

Note that (1.34) does not involve ξ's. So it must be the same as the integral equation obtained under the LBFS policy. Alternately, we can see that

$$f_k(\bar{Q}(t), 0) > 0 \Leftrightarrow \sum_{\{k\leq j\leq L,\, \sigma(j)=\sigma(k)\}} \bar{Q}_j(t) > 0.$$

So (1.34) becomes

$$\int_0^\infty \sum_{\{k\leq j\leq L,\, \sigma(j)=\sigma(k)\}} \bar{Q}_j(s)\, d\bar{I}_k(s) = 0, \tag{1.35}$$

thus yielding (1.25) and completing the proof. □

Lemma 6.3 *(1.27) holds for any LBFS policy.*

Proof. Let σ^* be a bottleneck server, i.e.,

$$\sum_{\{j:\, 1 \le j \le L \text{ and } \sigma(j)=\sigma^*\}} m_j = \frac{1}{\lambda^*}.$$

First, let us assume that σ^* is unique. We will relax this assumption later. Let b_{k^*} be the lowest indexed buffer at σ^*, i.e.,

$$b_{k^*} = \min\{j :\ 1 \le j \le L \text{ and } \sigma(j) = \sigma^*\}.$$

Then we claim that $\exists T < \infty$ such that for all $t > T$, $\bar{Q}_k(t) = 0$ for all $k \neq k^*$, and $\bar{Q}_{k^*}(t) = 1$. By Dai and Weiss [3], Prop. 4.2[2], this in turn implies that for all $k = 1, 2, ..., L$, and almost all $t > T$,

$$\frac{d}{dt}\bar{D}_k(t) = \lambda^*,$$

thus establishing (1.27).

We now prove the claim that there exists $T < \infty$ such that $\bar{Q}_k(t) = 0$ for all $k \neq k^*$ and $\bar{Q}_{k^*}(t) = 1$, for all $t > T$. The arguments in the proof will be very similar to those in Kumar and Kumar [11] and Dai and Weiss [3].

Suppose $k^* = L$. This implies in particular that b_L is the only buffer served by $\sigma(L)$. Consider $W(t) := \sum_{k=1}^{L-1} \bar{Q}_k(t)$ at a regular time t. Note that b_{L-1} has highest priority at $\sigma(L-1)$ since $\sigma(L-1) \neq \sigma(L)$ by assumption. Then, since $\sigma(L)$ is the bottleneck and the only buffer it serves is b_L, for almost all t such that $\bar{Q}_{L-1}(t) > 0$, we have

$$\frac{d}{dt}W(t) \le \mu_L - \mu_{L-1} < 0.$$

So by Lemma 2.2 of [3], $\exists t_{L-1} \le \frac{1}{\mu_{L-1} - \mu_L}$ such that $Q_{L-1}(t_{L-1}) = 0$. Arguing as in Lemmas 2 and 3 of [11], define

$$k := \max\{i < L-1 \mid \sum_{\{j:\, i \le j \le L \text{ and } \sigma(j)=\sigma(i)\}} m_j \ge m_{L-1}\}$$

if the set on the right hand side is non–empty, and $k := 0$ otherwise. As shown in [11], it follows that

$$\bar{Q}_{k+1}(t_{L-1} + \delta) = \cdots = \bar{Q}_{L-1}(t_{L-1} + \delta) = 0, \text{ for all } \delta \ge 0.$$

This argument can be iterated backwards from b_k. First we observe that there exists a time t_k such that $\bar{Q}_k(t_k) = 0$, noting that the input rate to

[2] Though [3] deals with fluid limits of open networks, their proposition can be adapted to closed networks with the modification that $d_0(t) = d_L(t)$.

the section $\{b_1, ..., b_k\}$ is no larger than λ^* while the output rate from b_k is $\frac{1}{\sum_{\{i:k\leq i\leq L \text{ and } \sigma(i)=\sigma(k)\}} m_i} > \lambda^*$. Then at time t_k we identify a k' such that

$$k' = \max\{j < k | \sum_{\{i:j\leq i\leq L \text{ and } \sigma(i)=\sigma(j)\}} m_i \geq \sum_{\{n:k\leq n\leq L \text{ and } \sigma(n)=\sigma(k)\}} m_n\}.$$

We argue that $b_{k'+1}, ..., b_k$ are empty at t_k and remain empty thereafter. The arguments are identical to those in [11] and are omitted. Iterating backwards from $b_{k'}$ completes the proof, when we reach b_1, leading to

$$\bar{Q}_1(t_1+\delta) = \cdots = \bar{Q}_{L-1}(t_1+\delta) = 0 \text{ for all } \delta \geq 0. \tag{1.36}$$

Suppose $k^* < L$. Then it is enough to show that there is a $t_{k^*+1} < \infty$ such that

$$\bar{Q}_{k^*+1}(t) = \cdots = \bar{Q}_L(t) = 0 \text{ for all } t \geq t_{k^*+1}. \tag{1.37}$$

All we would then have left to do for $t \geq t_{k^*+1}$ is to apply the case when $k^* = L$ to the section $b_1, b_2, ..., b_{k^*}$ with μ_j for $j = 1, 2, ..., k^*$ being replaced by

$$\tilde{\mu}_j := \mu_j \left(1 - \sum_{\{i:\, i>j \text{ and } \sigma(i)=\sigma(j)\}} m_i\lambda^*\right).$$

See (16) of [11] for the idea behind this argument. Now we prove (1.37). The key idea is the following. Suppose at a regular time τ, for some $j > k^*$, $\bar{Q}_j(\tau) > 0$ and $\bar{Q}_{j+1}(\tau) = \bar{Q}_{j+2}(\tau) = \cdots = \bar{Q}_L(\tau) = 0$. Then under LBFS,

$$\frac{d}{dt}\bar{D}_L(\tau) \geq \frac{1}{\sum_{\{i:j\leq i\leq L \text{ and } \sigma(i)=\sigma(j)\}} m_i} > \lambda^*.$$

Since this exit rate cannot be sustained for almost all $\tau \geq t$, it follows that at some t_{k^*+1},

$$\bar{Q}_{k^*+1}(t_{k^*+1}) = \cdots = \bar{Q}_L(t_{k^*+1}) = 0.$$

Moreover as shown in Lemma 3 of [11], this imples (1.37).

Let us now relax the assumption that the bottleneck server is unique. Suppose that there are exactly two bottleneck servers. Suppose k_1^* and k_2^* are the lowest priority buffers at bottleneck servers σ_1^* and σ_2^* respectively, and $k_1^* > k_2^*$. Then we proceed as follows, imitating the arguments above. For brevity, we will not repeat the arguments and we will confine ourselves to indicating the sections which empty. The exact same argument as leading to (1.37) shows that there exists a $t_{k_1^*+1} < \infty$ such that

$$\bar{Q}_{k_1^*+1}(t_{k_1^*+1}+\delta) = \cdots = \bar{Q}_L(t_{k_1^*+1}\delta) = 0 \text{ for all } \delta \geq 0.$$

Now consider the section $b_{k_2^*+1}, ..., b_{k_1^*-1}$. Using the same argument as that leading to (1.36), we see that there exists a $T_1 < \infty$ such that

$$\bar{Q}_{k_2^*+1}(T_1 + \delta) = \cdots = \bar{Q}_{k_1^*-1}(T_1 + \delta) = 0 \text{ for all } \delta \geq 0.$$

Finally, we consider the section $b_1, ..., b_{k_2^*-1}$. The input rate to this section cannot exceed λ^* after $t_{k_1^*+1}$. Also note that there is no buffer of higher priority than buffers $b_1, ..., , b_{k_2^*-1}$ which is nonempty after T_1. Once again, using the arguments leading to (1.36), we see that there exists a $T_2 < \infty$ such that

$$\bar{Q}_1(T_2 + \delta) = \cdots = \bar{Q}_{k_2^*-1}(T_2 + \delta) = 0 \text{ for all } \delta \geq 0.$$

Thus we have shown that there exists a $T' < \infty$ such that

$$\sum_{\{k: k \neq k_1^* \text{ and } k \neq k_2^*\}} \bar{Q}_k(T' + \delta) = 0 \text{ for all } \delta \geq 0,$$

and so for all $\delta \geq 0$,

$$\bar{Q}_{k_1^*}(T' + \delta) + \bar{Q}_{k_2^*}(T' + \delta) = 1.$$

By applying Dai and Weiss [3], Prop. 4.2, we obtain the result. The extension to any number of bottleneck servers is immediate. □

7 Efficiency of the Harrison–Wein policy

For closed queueing networks with *two* servers, Harrison and Wein [8] have examined the reflected Brownian motion approximation associated with the heavy traffic scenario, and conjectured that a particular buffer priority policy provides maximal throughput in the infinite population limit. In this section we prove this conjecture for two-station closed reentrant lines.

To describe their policy, it is convenient to imagine that when a customer leaves b_L, it exits from the system only to be replaced by a new customer in b_1. (As mentioned in Section 2, closed queueing networks arise from such window based admission control strategies). Let P_L be the same matrix as P except that all the elements of the L-th row are set to zero. Then $V := (I - P_L)^{-1} = I + P_L + P_L^2 + \cdots$ exists, and its ij-th element v_{ij} = expected number of visits to buffer b_j, before exit from a customer starting in b_i. Then

$$M_{\sigma,i} := \sum_{\{j: i \leq j \leq L \text{ and } \sigma(j) = \sigma\}} m_j v_{ij}$$

is the mean amount of work on a customer in b_i still remaining to be done by server σ prior to the customer's exit from b_L. Also let ρ_σ be the relative utilization of server σ as in (1.1). At the first server, $\sigma = 1$, rank the buffers to give higher priority to buffers with *smaller* values of the index

$$\eta_j := \rho_2 M_{1,j} - \rho_1 M_{2,j}.$$

At the second server ($\sigma = 2$) rank the buffers to give higher priority to buffers with *larger* values of the index η_j. The resulting buffer priority policy is enforced in a preemptive resume fashion and will be called the *Harrison–Wein policy* hereafter. The following result has been established in Jin, Ou and Kumar [9] using a different approach based on functional bounding by linear programs.

Theorem 7.1 *The Harrison–Wein policy for two-station closed reentrant lines is efficient.*

Proof. This proof uses Theorem 5.2(i). In the sequel we will use the notation

$$d_k(t) := \frac{d}{dt}\bar{D}_k(t),$$

to denote the derivative at the regular time t when the derivative exists.

Note that for reentrant lines

$$M_{\sigma,j} = \sum_{\{j \le k \le L \text{ and } \sigma(k)=\sigma\}} m_k,$$

and we can take ρ_σ to be $M_{\sigma,1}$. The next result identifies some properties of the priority ordering used in the Harrison–Wein policy.

Lemma 7.2 *Under the Harrison–Wein policy, the following are true.*

(i) If $\sigma(k) = \sigma((k+1) \bmod L)$ *then* $b_{(k+1)\bmod L} \succ b_k$.

(ii) If $j \le i-1$, $\sigma(j) = 1$ *and* $b_j \succ b_i$, *then*

$$\sum_{\{k: j \le k \le i-1; \sigma(k)=2\}} m_k \ge \frac{\rho_2}{\rho_1} \sum_{\{k: j \le k \le i-1; \sigma(k)=1\}} m_k.$$

(iii) If $j \ge i+1$, $\sigma(j) = 1$ *and* $b_j \succ b_i$, *then*

$$\sum_{\{k: j \le k \le L; \sigma(k)=2\}} m_k + \sum_{\{k: 1 \le k \le i-1; \sigma(k)=2\}} m_k \ge \frac{\rho_2}{\rho_1}\left[\sum_{\{k: j \le k \le L; \sigma(k)=1\}} m_k + \sum_{\{k: 1 \le k \le i-1; \sigma(k)=1\}} m_k\right].$$

Above, empty sums are taken to be zero.

Proof. (i) If $\sigma(k) = \sigma(k+1) = 1$ for some $k \in \{1, 2, ..., L-1\}$, then $\eta_k = \eta_{k+1} + \rho_2 m_k > \eta_{k+1}$ and so b_{k+1} always has higher priority than b_k. Similarly, if $\sigma(k) = \sigma(k+1) = 2$, for some $k \in \{1, 2, ..., L-1\}$, then $\eta_k = \eta_{k+1} - \rho_1 m_k < \eta_{k+1}$ and so b_{k+1} again has higher priority than b_k. If $\sigma(1) = \sigma(L) = 1$, then $0 = \eta_1 < \eta_L = \rho_2 m_L$ and so b_1 has higher priority than b_L. The result for the case when $\sigma(1) = \sigma(L) = 2$ is obtained similarly.

(ii) $\sigma(j) = \sigma(i) = 1$ and $b_j \succ b_i$ imply $\eta_j \leq \eta_i$, and so we have

$$\rho_2 \sum_{\{k:j \leq k \leq L; \sigma(k)=1\}} m_k - \rho_1 \sum_{\{k:j \leq k \leq L; \sigma(k)=2\}} m_k$$
$$\leq \rho_2 \sum_{\{k:i \leq k \leq L; \sigma(k)=1\}} m_k - \rho_1 \sum_{\{k:i \leq k \leq L; \sigma(k)=2\}} m_k,$$

from which we get

$$\rho_2 \sum_{\{k:j \leq k \leq i-1; \sigma(k)=1\}} m_k \leq \rho_1 \sum_{\{k:j \leq k \leq i-1; \sigma(k)=2\}} m_k.$$

Then (ii) is immediate.

(iii) As before, $\sigma(j) = \sigma(i) = 1$ and $b_j \succ b_i$ imply $\eta_i \geq \eta_j$, and so

$$\rho_2 \sum_{\{k:i \leq k \leq j-1; \sigma(k)=1\}} m_k \geq \rho_1 \sum_{\{k:i \leq k \leq j-1; \sigma(k)=2\}} m_k.$$

Thus,

$$\rho_1\rho_2 - \rho_2 \sum_{\{k:i \leq k \leq j-1; \sigma(k)=1\}} m_k \leq \rho_1\rho_2 - \rho_1 \sum_{\{k:i \leq k \leq j-1; \sigma(k)=2\}} m_k,$$

from which (iii) follows. □

Now we are ready to prove (1.28). Let us assume without loss of generality that $\rho_1 \leq \rho_2$, i.e, server 1 is faster. Then we claim that $\bar{I}_2(t) = 0$ for all t, for all solutions to (1.6–1.13, 1.22–1.25).

Suppose not. Then for some regular t, we must have

$$\frac{d}{dt}\bar{I}_2(t) > 0.$$

For this to happen, from (1.22–1.25) and Prop. 4.2 of Dai and Weiss [3], we must have $\sum_{\{k:1 \leq k \leq L; \sigma(k)=2\}} \bar{Q}_k(t) = 0$ and $d_k(t) = d_{(k-1) \bmod L}(t)$ for all $1 \leq k \leq L$ such that $\sigma(k) = 2$. Necessarily, it must also be true that

$$\sum_{\{k:1 \leq k \leq L; \sigma(k)=2\}} m_k d_k(t) < 1.$$

Now, $\sum_{\{k:1 \leq k \leq L; \sigma(k)=1\}} \bar{Q}_k(t) = 1$, and so

$$\sum_{\{k:1 \leq k \leq L; \sigma(k)=1\}} m_k d_k(t) = 1.$$

Thus,

$$\sum_{\{k:1\le k\le L;\sigma(k)=2\}} m_k d_k(t) - \sum_{\{k:1\le k\le L;\sigma(k)=1\}} m_k d_k(t) < 0. \tag{1.38}$$

At the regular time t, let b_{k^*} be the highest priority nonempty buffer at server 1.

If b_{k^*} is the lowest priority buffer at server 1, then we must have $d_j(t) = d_{(j-1)\mathrm{mod}L}(t)$ for all $j \neq k^*$, since every other buffer in the network must be empty (recall that server 2 is empty). Thus, $d_j(t) = d_{k^*}(t)$ for all j. Hence (1.38) yields

$$\left(\sum_{\{k:1\le k\le L;\sigma(k)=2\}} m_j - \sum_{\{k:1\le k\le L;\sigma(k)=1\}} m_j\right) d_{k^*}(t) = (\rho_2 - \rho_1) d_{k^*}(t) < 0,$$

which however cannot hold because $\rho_1 \le \rho_2$.

Thus b_{k^*} is *not* the lowest priority buffer at server 1. Let b_l be the first buffer at server 1 which is of lower priority than b_{k^*}, encountered as we traverse the network in the order b_{k^*}, b_{k^*+1}, ... wrapping around at b_L if need be. Now there are only two possibilities because of Lemma 7.2(i), either $l > k^*+1$ or $l \le k^*-1$. In the first case, $l > k^*+1$, we must have $d_{k^*}(t) = d_{k^*+1}(t) = \cdots = d_{l-1}(t)$; $d_l(t) = d_{l+1}(t) = \cdots = d_L(t) = 0$ and $d_1(t) = d_2(t) = \cdots = d_{k^*-1}(t) = 0$. Thus (1.38) becomes

$$\left(\sum_{\{j:k^*\le j\le l-1;\sigma(j)=2\}} m_j - \sum_{\{j:k^*\le j\le l-1;\sigma(j)=1\}} m_j\right) d_{k^*}(t) < 0.$$

But from Lemma 7.2(ii), since $l > k^*+1$,

$$\sum_{\{j:k^*\le j\le l-1;\sigma(j)=2\}} m_j \ge \frac{\rho_2}{\rho_1} \sum_{\{j:k^*\le j\le l-1;\sigma(j)=1\}} m_j \ge \sum_{\{j:k^*\le j\le l-1;\sigma(j)=1\}} m_j,$$

and therefore (1.38) cannot hold.

Last, we consider the case when $l \le k^*-1$. Then $d_{k^*}(t) = d_{k^*+1}(t) = \cdots = d_L(t)$; $d_1(t) = d_2(t) = \cdots = d_{l-1}(t) = d_{k^*}(t)$ (this second set of equations is, of course, irrelevant when $l = 1$), and $d_l(t) = d_{l+1}(t) = \cdots = d_{k^*-1}(t) = 0$. Thus (1.38) becomes

$$\begin{aligned}&\left(\sum_{\{j:k^*\le j\le L;\sigma(j)=2\}} m_j + \sum_{\{j:1\le j\le l-1;\sigma(j)=2\}} m_j\right) d_{k^*}(t)\\ &- \left(\sum_{\{j:k^*\le j\le L;\sigma(j)=1\}} m_j + \sum_{\{j:1\le j\le l-1;\sigma(j)=1\}} m_j\right) d_{k^*}(t) < 0\end{aligned} \tag{1.39}$$

But from Lemma 7.2(iii) we have

$$\begin{aligned} &\sum_{\{j:k^*\le j\le L;\sigma(j)=2\}} m_j + \sum_{\{j:1\le j\le l-1;\sigma(j)=2\}} m_j \\ &\ge \frac{\rho_2}{\rho_1}\left[\sum_{\{j:k^*\le j\le L;\sigma(j)=1\}} m_j + \sum_{\{j:1\le j\le l-1;\sigma(j)=1\}} m_j\right] \\ &\ge \sum_{\{j:k^*\le j\le L;\sigma(j)=1\}} m_j + \sum_{\{j:1\le j\le l-1;\sigma(j)=1\}} m_j, \end{aligned}$$

and so (1.39) cannot hold.

Thus, in all cases (1.38) cannot hold, and so

$$\frac{d}{dt}\bar{I}_2(t) = 0 \text{ for all regular } t,$$

thus completing the proof by Lemma 5.3. □

8 Inefficient reentrant lines

In this section, we give two examples of reentrant lines for which the integral equations (1.6–1.13) and (1.22–1.25) admit solutions which do not satisfy (1.26). This does not rule out the possibility that the actual fluid limits still satisfy (1.26).

The first example is similar to that of Harrison and Nguyen [7], which is the closed version of the examples in Kumar and Seidman [10] and Lu and Kumar [13], and is a counterexample to the conjecture that if all fluid limits of an open reentrant line under a buffer priority policy with Poisson arrivals empty in finite time, i.e, are stable, then all fluid limits of the corresponding closed reentrant line operated under the same buffer priority policy satisfy (1.26). The second example shows that the topology constraint that there are no self-loops, i.e., $\sigma(k) \neq \sigma(k+1)$, is not enough to guarantee (1.26) in two-server lines. This demonstrates that even in the two-server case when self loops are not the problem, alternate blocking and starvation leading to inefficient utilization could well be a problem. The second example also shows that even if there is only one bottleneck in a system, the fluid flows still need not satisfy (1.26).

Example 8.1 *The FBFS buffer priority policy can admit an inefficient solution to the fluid limit integral equations.*

Consider the closed reentrant line shown in Figure 1. Here $m_1 = m_3 = 1$ and $m_2 = m_4 = 0.2$. So $\lambda^* = 1/1.2$. The scheduling policy used is the FBFS policy which was shown to be stable in the open stochastic case by Dai [4].

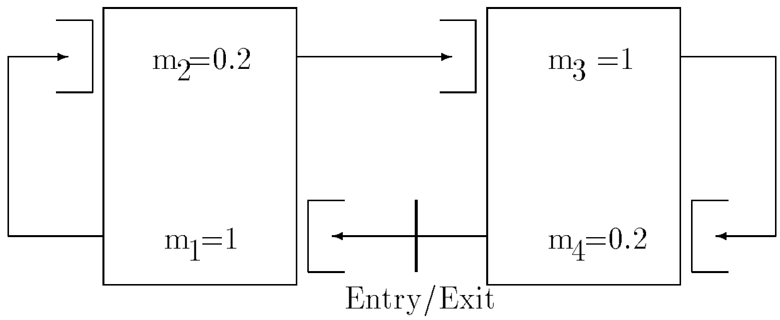

FIGURE 1. System of Example 1 under FBFS

Time	State
0	$(0, 1, 0, 0)$
0.2	$(0, 0, 0.8, 0.2)$
1.0	$(0, 0, 0, 1)$
1.2	$(0.8, 0.2, 0, 0)$
2.0	$(0, 1, 0, 0)$

TABLE 1.1. Inefficient trajectory for Example 1

So $b_1 \succ b_2$ and $b_3 \succ b_4$. Consider the initial condition $q = (0, 1, 0, 0)$. Then we can construct a piecewise linear trajectory which evolves as shown in Table 1.

Thus we see that $\frac{\bar{D}_L(t)}{t} \to 1/2 < \lambda^* = 1/1.2$, proving that (1.26) does not hold for this trajectory. Thus, in the closed case, with suitable choice of the "first" buffer, the FBFS policy can lead to inefficient fluid flows.

Example 8.2 *Two-server reentrant lines without self loops can admit an inefficient solution to the fluid limit integral equations.*

Consider the system shown in Figure 2, with the buffer priority policy $b_1 \succ b_5 \succ b_3$ and $b_4 \succ b_2 \succ b_6$. The processing times are $m_1 = 1$, $m_2 = 0.02$, $m_3 = 0.01$, $m_4 = 0.5$, $m_5 = 0.1$, and $m_6 = 0.02$. So $\lambda^* = 1/1.11$. An

Time	State
0	$(0, 0, 1, 0, 0, 0)$
0.0125	$(0, 0, 0, 0.975, 0, 0.025)$
0.5	$(0, 0, 0, 0, 0, 1)$
$0.5 + 1/49$	$(1 - 1/49, 0, 1/49, 0, 0, 0)$
1.5	$(0, 0, 1, 0, 0, 0)$

TABLE 1.2. Inefficient trajectory for Example 2

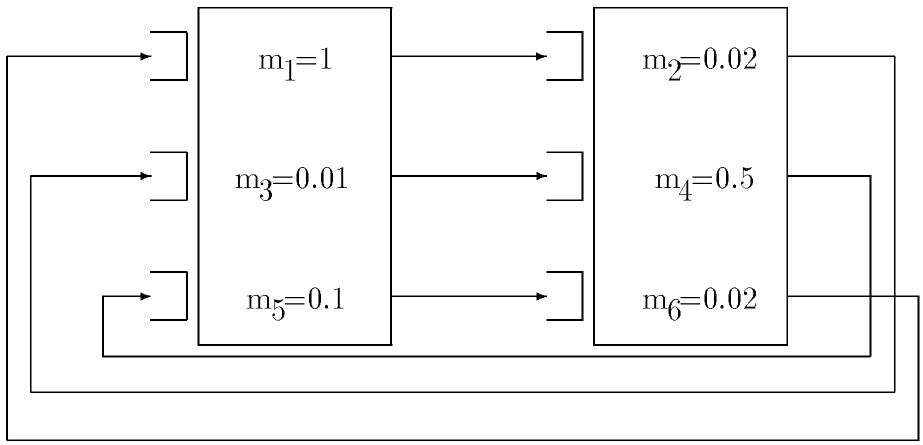

FIGURE 2. System of Example 2 without self-loops

inefficient piecewise linear trajectory which satisfies the integral equations is shown in Table 2 (with its breakpoints).

Thus we see that $\frac{\bar{D}_L(t)}{t} \to 1/1.5 < 1/1.11$ and hence (1.26) cannot be satisfied.

9 Concluding remarks

We have investigated the fluid limit approach for analyzing the heavy traffic behavior of closed queueing networks. We have shown that the efficiency of such networks can be established by studying the throughput of the fluid limits. Using this approach, we have proved the efficiency of the class of all FSMCT policies, including the Last Buffer First Serve policy, for general closed reentrant lines, and the Harrison–Wein buffer priority policy for two-station closed reentrant lines. We have also presented examples which illustrate how inefficiency can arise in closed reentrant lines and shown that buffer priority policies which lead to stable fluid flows in an open network may still inefficiently utilize servers in the corresponding closed fluid flow network.

Acknowledgments: The authors are grateful to an anonymous reviewer for suggesting a slight extension to the original Lemma 4.1 and Theorem 4.2 which featured “lim infs” rather than “lim sups” in the statements of the results. The research reported here has been supported in part by the U. S. Army Research Office under Grant No. DAAH 04-95-1-0090, the National Science Foundation under Grant No. ECS-94-05371, and the Joint Services

Electronics Program under Contract No. N00014-90-J-1270.

10 References

[1] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] H. Chen and A. Mandelbaum. Discrete flow networks: Bottleneck analysis and fluid approximations. *Math. Oper. Res.*, 16:408–446, 1991.

[3] J. Dai and G. Weiss. Stability and instability of fluid models for certain re-entrant lines. Preprint, February 1994. To appear in Mathematics of Operations Research.

[4] J. G. Dai. Stability of open multiclass queueing networks via fluid models. Technical report, Georgia Institute of Technology, 1994.

[5] J. G. Dai and R. J. Williams. Existence and uniqueness of semimartingale reflecting Brownian motions in convex polyhedrons. Submitted to *Theory of Probability and its Applications*, 1994.

[6] S. N. Ethier and T. G. Kurtz. *Markov Processes: Characterization and Convergence.* Wiley, New York, NY, 1985.

[7] J. M. Harrison and V. Nguyen. Some badly behaved closed queueing networks. Technical Report 3691-94-MSA, Stanford and MIT, May 1994.

[8] J. M. Harrison and L. M. Wein. Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Operations Research*, 38(6):1052–1064, 1990.

[9] H. Jin, J. Ou, and P. R. Kumar. The throughput of closed queueing networks–functional bounds, asymptotic loss, efficiency, and the Harrison-Wein conjectures. Submitted to *Mathematics of Operations Research*, October 1994.

[10] P. R. Kumar and T. I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control*, AC-35(3):289–298, March 1990.

[11] S. Kumar and P. R. Kumar. The last buffer first policy is stable for stochastic re–entrant lines. Technical report, Coordinated Science Laboratory, University of Illinois, Urbana, IL, 1994.

[12] S. Kumar and P. R. Kumar. Fluctuation smoothing policies are stable for stochastic re-entrant lines. To appear in *Discrete Event Dynamical Systems*, 1996.

[13] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, AC-36(12):1406–1416, December 1991.

[14] Steve C. H. Lu, Deepa Ramaswamy, and P. R. Kumar. Efficient scheduling policies to reduce mean and variance of cycle–time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing*, 7(3):374–385, August 1994.

[15] Steve H. Lu and P. R. Kumar. Fluctuation smoothing scheduling policies for queueing systems. In *Proceedings of the Silver Jubilee Workshop on Computing and Intelligent Systems*, Indian Institute of Science, Bangalore, December 1993.

[16] D. J. Miller. Simulation of a semiconductor manufacturing line. *Communications of the ACM*, 33(10):98–108, 1990.

[17] J. Walrand. *Communication Networks–A First Course*. Aksen Associates, Boston, 1991.

[18] L. M. Wein. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 1(3):115–130, August 1988.