MATHEMATICS OF OPERATIONS RESEARCH
Vol. 29, No. 4, November 2004, pp. 891–922
ISSN 0364-765X | EISSN 1526-5471 | 04 | 2904 | 0891

DOI 10.1287/moor.1040.0102

Stabilizing Queueing Networks with Setups

J. G. Dai
School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, Georgia 30332-0205, dai@isye.gatech.edu

O. B. Jennings
Fuqua School of Business, Duke University, Durham, North Carolina 27708-0120, otisj@duke.edu

For multiclass queueing networks, dispatch policies govern the assignment of servers to the jobs they process. Production policies perform the analogous task for queueing networks whose servers are subject to switch-over delays or setups, a model we refer to as *setup networks*. It is well known that a poorly chosen dispatch policy may lead to instability of a multiclass queueing network, even when the traffic intensity at each station is less than one and the policy is nonidling. Not surprisingly, setup networks and production policies inherit these instability concerns. With this in mind, we define a family of "sensible" production policies that are adaptations of dispatch policies and restrict the frequency of setup performance.

We provide a framework for proving the stability of a setup network operating under a sensible production policy. Central to this framework is the artificial fluid model of a setup network. The artificial fluid models presented are generalizations of standard fluid models of multiclass queueing networks; see, for example, Dai (1995). Unlike their standard fluid model counterparts, artificial fluid models do not arise directly from a limiting procedure on some discrete network process; hence the "artificial" qualifier. Nevertheless, stability of the artificial fluid model implies stability of the associated setup network, a connection paralleling the main result of Dai (1995).

As an exercise in using the artificial fluid model framework for proving stability of setup networks, we investigate several production policies adapted from dispatch policies. One production policy of particular interest involves a modification of the first-in-first-out dispatch policy.

Key words: setup delay; multiclass queueing networks; fluid models; stability; throughput; Kelly networks
MSC2000 subject classification: Primary: 68M20, 90B15, 60K25
OR/MS subject classification: Primary: Queues: networks
History: Received June 30, 2002; revised December 24, 2003.

1. Introduction. Multiclass queueing networks are effective tools for modelling many industrial settings. One setting for which the model is particularly attractive is the production flow within semiconductor manufacturing facilities. Still, there are some significant aspects of industrial settings that escape the modelling scope of multiclass queueing networks, which henceforth we refer to as *standard networks*. One such feature is the effect of server switch-over delays or *setups*. For some workstations, the processing resource (server) must first incur a delay before switching its processing efforts from one class of job to another. Hence, there is a natural deterrent to switching, measured in lost potential server effort, that the standard network model cannot capture. In this paper, we extend the standard network model by including setups at each station. The resulting model, queueing networks with setups, or *setup networks* for short, is formally presented in §2. Other works that consider setup networks are Bertsimas and Nino-Mora (1999), Warren (1997), and Jennings (2000).

Beyond the presentation of our setup network model, the primary concern of this paper is the stability of such networks. Our notion of stability, formalized in §2.4, is analogous to rate stability of standard networks. Rate stability has been advanced by Stidham and coauthors under the name pathwise stability (see El-Taha and Stidham 1999 and references therein). In short, a network is rate stable if the long-run input rate of the system is matched by the long-run output rate.

The study of single server systems subject to setups has an extensive history; see, for example, Takagi (1986, 1990) for surveys of early work related to telecommunications and Lan and Olsen (2004) for a more contemporary review. One of the more frequently investigated server scheduling policies is exhaustive service, also referred to as setup avoidance. As the name suggests, exhaustive service involves processing all jobs of a given class until jobs of that class are no longer available. Only then is the server allowed to switch to

another class. An immediate benefit is that, among all policies that do not idle when work is present, exhaustive service generates the largest stability region. That is, if the system can be stable, then it will be stable under exhaustive service.

Ideally, the benefits of exhaustive service would extend to the network setting. Gershwin (1995) warns against employing exhaustive service or other policies that spend long stretches of time on a single job class in a network setting, claiming such practices result in large inventories and delays. Moreover, Kumar and Seidman (1990) show that exhaustive service can lead to instability in a standard network, a more immediate concern. Jennings (2000) provides a simulation example, illustrating how exhaustive service may lead to instability in a setup network.

Ensuring stability is a concern for both setup and standard networks. Still, directing the processing efforts of servers that incur setups is fundamentally different from doing so for setup-exempt servers. Hence, we refer to the server scheduling rule in standard networks as the *dispatch policy* and the analogous rule in the setup network setting as the *production policy*.

In this paper, we define a family of "sensible" production policies that are adaptations of dispatch policies and restrict the frequency of setup performance. Then, we provide a framework, using artificial fluid models of setup networks, to prove the stability of a setup network operating under a specific sensible production policy. Finally, we illustrate the use of the framework by proving the stability of setup networks operating under three sensible production policies.

The *artificial fluid model* presented here is a generalization of fluid models of standard networks. Having achieved widespread acceptance in the literature, fluid models of standard networks will be referred to as *standard fluid models*. Unlike their standard fluid model counterparts, artificial fluid models do not arise directly from a limiting procedure of some discrete network process, hence, the "artificial" qualifier. Nevertheless, stability of the artificial fluid model implies stability of the setup network (see Theorem 3.3), a connection paralleling the relation between standard fluid models and standard networks; see Dai (1995).

The sensible production policies that we use as test cases are modifications of three dispatch policies: first-in-first-out (FIFO), early-steps-first (ESF), and generalized round robin (GRR). We highlight the first of these, FIFO, which requires special network structure. It was shown in Bramson (1996) that a Kelly-type standard network operating under the FIFO dispatch policy is stable. (All jobs processed at a given station in a Kelly-type network have the same mean processing time, regardless of the class in which they reside.) In this paper, we show that a class of sensible FIFO production policies stabilizes a setup network if its corresponding standard network is Kelly-type; see Theorem 5.2.

The study of the stability of networks with setups is limited. Perkins and Kumar (1989) consider a deterministic system and provide stabilizing production policies. Jennings (2000) provides a framework for demonstrating that a setup network is positive Harris recurrent, a stronger notion of stability than the one considered here. Warren (1997) uses an approximation method to predict the stability of setup networks.

There are recent works in the literature that extend the standard network model in other directions. For example, Dai and Li (2003) treat the batch processing operations issue. A similar service was performed by Andradóttir et al. (2003) for networks of flexible servers subject to setups. Dai and Jennings (2003) present a model including setups, batch processing, and multiserver workstations.

The remainder of this paper will progress as follows. Immediately following, we present the setup network model and formalize our definition of rate stability. In §3, we present a collection of equations governing the dynamics of the setup network and the continuous-flow analog of these equations, the artificial fluid model. Next, we draw the connection between artificial fluid models and standard fluid models in §4. In particular, we discuss

how the Lyapunov functions used to demonstrate stability of the latter may be as effective for the former. Three examples of sensible production policies are examined in §5. Section 6 gives concluding remarks. For ease of exposition, some technical proofs are delayed until the appendix.

2. Open queueing networks with setups. In this section, we present the family of networks under study throughout this paper. The model is an extension of open multiclass queueing networks, as presented by Harrison (1988). For the purpose of this paper, we will refer to that model as the standard network. Our model, which adds setup times to the standard network, will be referred to as the *open queueing network with setups*, or setup network, for short. After drawing connections between scheduling the resources of both standard and setup networks, we conclude this section with a formal definition of rate stability.

2.1. Network description. Consider a network of stations, labelled $j = 1, \ldots, J$. The stations are populated by classes, labelled $k, k' = 1, \ldots, K$, where class k is associated with a unique station $\sigma(k)$. Whenever k and j appear together, it is implied that $j = \sigma(k)$. The collection $\mathscr{C}(j)$ of all classes associated with station j is referred to as the station's constituents.

Jobs, the basic unit of flow, enter the network exogenously and change classes as they move through the network. While awaiting processing, class k jobs are said to reside in buffer k. There is a one-to-one relationship between classes and buffers. "Buffer" is used to connote physical location, as in a storage place for jobs awaiting processing. The "class" label has a more metaphysical interpretation. For example, one removes a job from a buffer for processing, but the job retains its class designation until processing is complete.

Each station is manned by a single server responsible for processing the resident jobs. A server may process at most one job at a time. Once the processing of a job begins it cannot be interrupted; in other words, there is no preemption of service. (There may be occasions where it makes sense to allow preemption of service. We exclude it for modelling ease.) We will occasionally say that a server is processing a class or buffer, meaning, processing the jobs from that class.

Suppose a server last processed class k and is about to process a job from a different class $k' \neq k$. Before the actual processing can begin, the server must perform a *setup*. That is, a delay is incurred whenever a server switches its processing efforts between classes. Setups are denoted by the pair (k, k'), where k signifies the class just processed and k' is the subsequent class. Generally, the duration of a type (k, k') setup, or *setup time*, depends on both k and k', as well as their order. In this sense, setups are *sequence dependent*. With the inclusion of setups in our model, servers are always in one of three states: *idle*, *in-service*, or *in-setup*. In addition to not being interrupted while processing jobs, servers are never preempted while performing a setup.

For each class k, we have the cumulative processes $E_k = \{E_k(t), t \geq 0\}$, $V_k = \{V_k(n)\colon n = 1, 2, \ldots\}$, and $\Phi^k = \{\Phi^k(n)\colon n = 1, 2, \ldots\}$. For each time $t \geq 0$, $E_k(t)$ counts the number of external arrivals to class k in $[0, t]$. For each positive integer n, $V_k(n)$ records the total service time requirement for the first n class k jobs. For each positive integer n, $\Phi^k(n)$ is a K-dimensional vector with each component being a nonnegative integer. For each class k', $\Phi^k_{k'}(n)$ records the number of the first n *processed* class k jobs that are routed to buffer k' upon completion of service. When $\Phi^k(n-1) = \Phi^k(n)$, the nth processed class k job immediately leaves the system. By convention, we assume

$$E_k(0) = 0, \qquad V_k(0) = 0, \qquad \text{and} \qquad \Phi^k(0) = 0.$$

For each $t \geq 0$, we extend the definitions of $V_k(t)$ and $\Phi^k(t)$ as follows:

$$V_k(t) = V_k(\lfloor t \rfloor) \qquad \text{and} \qquad \Phi^k(t) = \Phi^k(\lfloor t \rfloor),$$

where $\lfloor t \rfloor$ denotes the largest integer less than or equal to t.

In addition, we define the cumulative process $F_{kk'} = \{F_{kk'}(n)\colon n = 1, 2, \dots\}$, associated with type (k, k') setups. When k and k' appear together in the same subscript of a setup quantity, it is implied that the classes are distinct, $k \neq k'$, and that they reside at the same station, $\sigma(k) = \sigma(k')$. For each positive integer n, $F_{kk'}(n)$ records the total time required for the first n setups from class k to k'. Again, we assume $F_{kk'}(0) = 0$. Furthermore, for each $t \geq 0$, we extend the definition so that $F_{kk'}(t) = F_{kk'}(\lfloor t \rfloor)$.

We call (E, V, Φ, F) the set of primitive processes, where $E = \{E(t), t \geq 0\}$, $V = \{V(t), t \geq 0\}$, $\Phi = \{\Phi(t), t \geq 0\}$, and $F = \{F(t), t \geq 0\}$, with $E(t) = (E_1(t), E_2(t), \dots, E_K(t'))$, $V(t) = (V_1(t), V_2(t), \dots, V_K(t'))$, $\Phi(t) = (\Phi^1(t)), \Phi^2(t), \dots, \Phi^K(t))$, and $F(t) = \{F_{kk'}(t),\ k, k' \in \mathscr{C}(j),\ j = 1, \dots, J\}$. (All vectors are envisioned as column vectors unless otherwise stated. Prime on a vector or a matrix denotes transpose.)

We assume that the strong law of large numbers holds for the primitive processes; namely, with probability one,

$$\lim_{t\to\infty} \frac{E_k(t)}{t} = \alpha_k, \qquad \lim_{t\to\infty} \frac{V_k(t)}{t} = m_k, \tag{1}$$

$$\lim_{t\to\infty} \frac{\Phi^k_{k'}(t)}{t} = P_{kk'}, \qquad \text{and} \qquad \lim_{t\to\infty} \frac{F_{kk'}(t)}{t} = s_{kk'}.$$

The parameter set (α, m, P, s) with $\alpha = (\alpha_1, \dots, \alpha_K)'$, $m = (m_1, \dots, m_K)'$, $P = (P_{kk'})$, and $s = \{s_{kk'}, k, k' \in \mathscr{C}(j),\ j = 1, \dots, J\}$ has the following interpretation: For each k, α_k is the external job arrival rate to buffer k and m_k is the mean service time for class k jobs. For classes k and k', the quantity $P_{kk'}$, referred to as the routing probability from k to k', captures the long-run fraction of class k jobs that become class k' jobs immediately after being processed. The $K \times K$ matrix P is called the routing matrix. For each pair (k, k'), $s_{kk'}$ is the mean setup time when a server switches from class k to class k'. Let $s_k = \max_{k' \in \mathscr{C}(j)} s_{k'k}$ be the maximum possible mean setup time *to* class k.

We introduce the counting processes $\Psi = \{\Psi(t)\colon t \geq 0\}$ and $\Upsilon = \{\Upsilon(t)\colon t \geq 0\}$, associated with the primitive service process V and the primitive setup process F, respectively. For each time $t \geq 0$, $\Psi(t) = (\Psi_1(t), \dots, \Psi_K(t))'$, with

$$\Psi_k(t) = \max\{n\colon V_k(n) \leq t\}, \quad k = 1, \dots, K,$$

and $\Upsilon(t) = \{\Upsilon_{kk'}(t),\ k, k' \in \mathscr{C}(j),\ j = 1, \dots, J\}$, with

$$\Upsilon_{kk'}(t) = \max\{n\colon F_{kk'}(n) \leq t\}, \quad k, k' \in \mathscr{C}(j),\ \ j = 1, \dots, J.$$

The process $\Upsilon_{kk'}$ is only defined when $s_{kk'} > 0$. It follows from the strong law of large numbers (1) that, with probability one,

$$\lim_{t\to\infty} \frac{\Psi_k(t)}{t} = \mu_k, \quad k = 1, \dots, K, \tag{2}$$

where $\mu_k = 1/m_k$, and that, for each setup pair (k, k') such that $s_{kk'} > 0$,

$$\lim_{t\to\infty} \frac{\Upsilon_{kk'}(t)}{t} = 1/s_{kk'}. \tag{3}$$

We assume that the network is open, i.e., the matrix

$$Q = I + P' + (P')^2 + \cdots$$

is finite, which is equivalent to the fact that $(I - P')$ is invertible such that $Q = (I - P')^{-1}$. Let $\lambda = (\lambda_1, \dots, \lambda_K)'$ be the vector of nominal total arrival rates. It is defined by the following system of equations:

$$\lambda_k = \alpha_k + \sum_{k'=1}^{K} \lambda_{k'} P_{k'k} \quad \text{for each } k = 1, \dots, K. \tag{4}$$

In vector form, $\lambda = \alpha + P'\lambda$. Because $(I - P')$ is invertible, the unique solution to (4) is given by $\lambda = Q\alpha$. We define the traffic intensity ρ_j for station j as

$$\rho_j = \sum_{k \in \mathscr{C}(j)} \lambda_k m_k, \quad j = 1, \ldots, J, \tag{5}$$

with $\rho = (\rho_1, \rho_2, \ldots, \rho_J)'$ being the corresponding vector. When, for each station $j = 1, \ldots, J$, we have

$$\rho_j < 1, \tag{6}$$

we say that the usual traffic condition is satisfied for the setup network.

2.2. The standard network and dispatch policies. We now define the corresponding *standard network* of a setup network. The corresponding standard network is identical to the setup network except that the setup times are zero such that the server exists in either the idle state or the in-service state. Moreover, the traffic intensity for station j in the standard network is the same as in the setup network, i.e., $\tilde{\rho}_j = \rho_j$. (Our notational convention is to use a tilde when referring to a quantity associated with the standard network.) It goes without saying that the usual traffic condition for the standard network holds if and only if the traffic condition for the setup network (6) holds. For a setup network driven by the primitive processes (E, V, Φ, F) the corresponding standard network is driven by the primitive processes (E, V, Φ).

Whenever multiple jobs reside at a station, there is discretion in the processing order of those jobs. For standard networks, the *dispatch policy* $\tilde{\pi}$ is the sole mechanism by which servers are assigned to classes. That is, when a server becomes available for processing, the dispatch policy selects the class from which the next job will be processed. Given the class assignment, the oldest job, based on arrival to the associated buffer, is processed. In this sense, jobs within a single buffer are processed in a FIFO fashion. A dispatch policy is said to be *nonidling* if a server is never in the idle state when jobs are present at the station.

Not surprisingly, scheduling of servers in a standard network is less complex than in a setup network. To stress this point, an alternative term is used to distinguish between the two scheduling tasks. A *production policy* is to the setup network what a dispatch policy is to a standard network. One of the main themes of this paper is that dispatch policies that work well for standard networks are useful in crafting effective production policies for setup networks.

2.3. Production policies and sensible policies. The decision process governing the processing of jobs in a setup network is embodied in the production policy. Because of the complex nature of each station, we envision most "useful" production policies having the following three-tiered approach. When a server requires an assignment to a class for processing, one first filters the set of constituent classes into a subset of *eligible* classes. Second, the server is *dispatched* to one of the eligible classes. The final decision involves setting the termination time of the assignment.

Throughout this paper, we assume production policies have the form $\pi = (\theta, \tilde{\pi}, l)$. The K-dimensional vector $\theta = (\theta_1, \theta_2, \ldots, \theta_K)$ of positive integers enforces the filtering function. When a server at station j requires an assignment, the constants determine which constituent classes are eligible. Class k is eligible if the number of class k jobs is greater than or equal to θ_k. (Exceptions that relax the eligibility requirement can be found in §5.3.) It is possible that the collection of eligible classes at station j is empty. In cases where no class passes the first eligibility test, the criterion is relaxed. Under the relaxed test, any class with a nonempty buffer is eligible. From the standpoint of the first eligibility test,

if the number of class k jobs is less than θ_k, that class is effectively empty. In this sense, the components of θ can be thought of as *thresholds*.

The second component of the production policy π is the dispatch policy $\tilde{\pi}$. This terminology is borrowed from literature on standard networks. The idea here is the same. Given the current state of the system, dispatch policies perform the actual assignment of servers to classes. With the standard network, the assignment lasts at most through the processing of a single job. For setup networks, however, the assignment lasts (potentially) for the processing of several jobs. Moreover, the assignment is restricted to the set of eligible classes.

The stretch of time during which the dispatched assignment holds is referred to as the *production run*. The length of the production run or, equivalently, the number of jobs processed before seeking a new assignment, is the third and final decision to be made. The vector l determines the length of the production run. If the assignment is for a class k that passed the first eligibility test, then we process l_k jobs before seeking a new assignment, if possible. If the class only passed the relaxed eligibility test, we process one job and terminate the production run immediately, allowing the opportunity to make another production run decision based on updated system information.

The nonidling property for setup networks is not as straightforward as for the standard network. As the name suggests, nonidling now refers to both the in-service and in-setup states. A server in the setup network is said to be *busy* if it is either in-service or in-setup. A production policy is said to be *nonidling* if each server is allowed to idle only when there are no jobs at the corresponding station. (This is not to say that nonidling policies are ideal. Indeed, in some instances it may beneficial to delay commitment to the processing of a given buffer until some buffer accumulates a critical number of jobs; see, e.g., Cooper et al. (1998) and references therein.)

Assuming that a server is assigned to a class k that passed the first eligibility test, we are assured that at least θ_k class k jobs are available at the beginning of the production run to be processed. If we assume the thresholds are set such that they exceed the maximum number of jobs processed in a production run, i.e.,

$$\theta_k \geq l_k, \tag{7}$$

then l_k class k jobs will be processed during the production run. One can amortize the average setup time incurred from switching to class k over all of the jobs processed during the subsequent production run. The results are *setup-adjusted mean processing times* and *rates*, which are defined, respectively, to be

$$\breve{m}_k \equiv m_k + s_k/l_k \quad \text{and} \quad \breve{\mu}_k \equiv 1/\breve{m}_k, \qquad k = 1, \ldots, K. \tag{8}$$

With the adjusted mean processing times, one can compute a *setup-adjusted traffic intensity*

$$\breve{\rho}_j = \sum_{k \in \mathscr{C}(j)} \lambda_k \breve{m}_k, \quad j = 1, \ldots, J. \tag{9}$$

Throughout this paper, the convention $\breve{x}$ signifies an inflation of the quantity x, due to the amortization of setup time, or simply an adjustment to the quantity x related to the influence of setups.

We conclude with a further restriction of the type of production policies under consideration. A production policy is said to be *sensible* if it is nonidling, condition (7) holds, and $\breve{\rho}$ obeys the usual traffic condition; i.e.,

$$\breve{\rho}_j < 1 \quad \text{for each } j = 1, \ldots, J. \tag{10}$$

The reasoning behind the restriction to sensible policies is straightforward. By ensuring that the servers avoid spending an inordinate amount of time performing setups, sensible

policies eliminate setups as sources of instability. Furthermore, implicit in the sensible policy condition is the ability to make trade-offs of production run lengths. For example, it is simple to compensate for setting a relatively short production run for one class with a long production run for another class at the same station. One might use such trade-offs as a mechanism for accommodating high priority "hot lots," for which production run considerations are secondary at best. Condition (10) ensures that no job can be ignored indefinitely under any nonidling policy.

2.4. Rate stability. We now define rate stability for setup networks. Let $D_k(t)$ denote the number of jobs in the setup network that have departed class k during the interval $[0, t]$. In the following definition, the term *state* is used. The precise definition of a state depends on the particular production policy used. The system state typically includes, but is not limited to, the number of jobs in each class, the status and assignment of each server, the remaining processing times of the jobs being processed, the remaining interarrival times for jobs arriving from outside the network, the lengths so far of the current production runs, and the remaining setup time for each server. We do not attempt a precise definition of state here. Roughly speaking, a state is a snapshot of the network at any given time. It should contain enough information such that once the current state of the network is given, the future evolution of the network is completely determined in distribution. Readers are referred to Dai (1995) and Bramson (1998) for examples and additional discussions of states in standard networks under various dispatch policies.

DEFINITION 2.1. A setup network is *rate stable* if, for each fixed initial state, with probability one,

$$\lim_{t\to\infty} \frac{D_k(t)}{t} = \lambda_k \quad \text{for each } k = 1, \ldots, K. \tag{11}$$

The setup network is rate stable if the throughput rate or departure rate from each class is equal to the nominal total arrival rate to that class. Rate stability has been advanced by Stidham and co-authors under the title pathwise stability; see El-Taha and Stidham (1999) and references therein. (There are other definitions of stability, such as positive Harris recurrence; see Dai 1995. The results in this paper can be extended to those settings as well; see, e.g., Jennings 2000.) This notion of stability, for the standard network setting, is discussed in Chen (1995). As in a standard network, $\rho_j \le 1$ for each station j is a necessary condition for rate stability of a setup network; see Dai (1999). The case when $\rho_j = 1$ for some stations j is more subtle, and is not considered in this paper. As stated earlier, even though we specify that the setup-adjusted traffic intensities for sensible policies must obey the usual traffic condition, this is not necessary for stability of the setup network.

3. Network and fluid model equations. In this section, we define fluid models of both setup networks and standard networks. Fluid models are continuous, deterministic analogs of discrete networks and are defined through a set of equations. To describe the fluid models, we start with equations governing the dynamics of the discrete networks. Unless explicitly stated otherwise, we assume that the setup network is operated under a sensible production policy π and the standard network is operated under a nonidling dispatch policy $\tilde{\pi}$.

3.1. Network dynamics. The dynamics of the setup network are captured by the process $\mathbb{X} = (A, D, S, T, U, Y, Z)$. The components $A = \{A(t), t \ge 0\}$, $D = \{D(t), t \ge 0\}$, $T = \{T(t), t \ge 0\}$, $S = \{S(t), t \ge 0\}$, and $Z = \{Z(t), t \ge 0\}$ are K-dimensional. For each class k, $A_k(t)$ denotes the number of jobs that have arrived to class k (from external and internal sources) in $[0, t]$, $D_k(t)$ denotes the number of jobs that have departed from class k in $[0, t]$, $S_k(t)$ denotes the amount of time the server at station $j = \sigma(k)$ has spent setting up for class k during the interval $[0, t]$, $T_k(t)$ denotes the amount of time the server at

station $j = \sigma(k)$ has spent processing class k jobs during interval $[0, t]$, and $Z_k(t)$ denotes the total number of class k jobs that are buffered or being served at station j at time t. The processes A, D, S, T, and Z are called the arrival, departure, setup allocation, service allocation, and *job-count* processes, respectively. The components $U = \{U(t), t \geq 0\}$ and $Y = \{Y(t), t \geq 0\}$ are J-dimensional. For each station j, $U_j(t)$ denotes the total number of jobs at station j that are buffered or being served at time t, and $Y_j(t)$ denotes the amount of time the server at station j has been idle in the time interval $[0, t]$. The process Y is called the cumulative idle time process. The process $\mathbb{X} = (A, D, S, T, U, Y, Z)$ satisfies the following set of equations:

$$A(t) = E(t) + \textstyle\sum_k \Phi^k(D_k(t)), \quad t \geq 0, \tag{12}$$

$$Z(t) = Z(0) + A(t) - D(t), \quad t \geq 0, \tag{13}$$

$$Z(t) \geq 0, \quad t \geq 0, \tag{14}$$

$$U(t) = CZ(t), \quad t \geq 0, \tag{15}$$

$$C(S(t) + T(t)) + Y(t) = et, \quad t \geq 0, \tag{16}$$

$$Y_j(t) \text{ can increase only if } U_j(t) = 0, \quad j = 1, \ldots, J, \tag{17}$$

$$\Psi_k(T_k(t)) = D_k(t), \quad t \geq 0, \ \ k = 1, \ldots, K, \tag{18}$$

(19) additional equations associated with the particular production policy π.

Here, C is the constituency matrix defined as

$$C_{jk} = \begin{cases} 1 & \text{if } k \in \mathscr{C}(j), \\ 0 & \text{otherwise,} \end{cases}$$

and e is the J-dimensional vector of 1s.

We provide a brief interpretation of Equations (12)–(19); where convenient, the interpretation is componentwise. Equation (12) implies the cumulative arrivals to buffer k consists of those jobs arriving to k from the outside ($E_k(t)$), and those jobs routed to class k after being processed in some other class. As for (13), the class k job-count process at time t, $Z_k(t)$, is equal to the number of class k jobs present initially, plus all jobs that have arrived to buffer k thus far, net those class k jobs that have been processed. Expression (14), referred to as the nonnegativity constraint, is self-explanatory. For each j, the stationwide job-count quantities are computed in (15). Equation (16) tracks, for each station j, how the total server time has been distributed, up until time t, i.e., between performing setups, processing jobs, and idling. Equation (17) reflects the nonidling condition. We capture the departures as a function of dedicated server effort in (18). The equation is well known for standard networks operating under a head-of-line dispatch policy. Finally, the production policy π, used to govern the scheduling of servers, will have a major effect on system dynamics, hence, the provision in (19).

When enough jobs are present at the station, a sensible production policy restricts the frequency of class k setups to one for every l_k jobs. To facilitate capturing the restriction mathematically, we expand the definition of $S_k(t)$ to $S_{k'k}(t)$, the cumulative server time spent performing setups from k' to k. Clearly, $S_k(t) = \sum_{k'} S_{k'k}(t)$. Suppose (7) holds. Then, for every $0 < t_1 < t_2$, if, for every $s \in [t_1, t_2]$, $Z_k(s) \geq \theta_k$ for some $k \in \mathscr{C}(j)$, then

$$D_k(t_2) - D_k(t_1) \geq l_k \left(\sum_{k' \in \mathscr{C}(j)} \left[\Upsilon_{k'k}(S_{k'k}(t_2)) - \Upsilon_{k'k}(S_{k'k}(t_1)) \right] - 2 \right) \tag{20}$$

for each $k \in \mathscr{C}(j)$ and $j = 1, \ldots, J$. The left-hand side of (20) is the number of processed jobs within the interval $[t_1, t_2]$. Because class k is eligible throughout this interval, when both a setup for class k and the subsequent production run lie entirely within the interval, the production run consists of exactly l_k jobs. Hence, the l_k term on the right-hand side. The -2 term accounts for the setups and production runs at the beginning and end of the interval. The setup at the beginning of the time horizon could have been initiated when no class passed the first (stricter) eligibility test. Hence, the setup is not necessarily followed by the processing of l_k jobs. Likewise, the production run succeeding the last setup within the interval may extend beyond time t_2, again preempting the service of l_k jobs. Finally, if two class k production runs occur in succession, no setup is incurred between the runs and the inequality in (20) obtains more slack. We call Equations (12)–(20) the *setup network equations*. Note that S, T, and Y are continuous, and that A, D, and Z are right continuous with left limits. All variables are nonnegative in each component, with A, D, S, T, and Y being nondecreasing. By assumption,

$$A(0) = D(0) = S(0) = T(0) = Y(0) = 0.$$

For each setup network driven by (E, V, Φ, F), the corresponding standard network driven by (E, V, Φ) has similar processes. In contrast with the setup network process $\mathbb{X} = (A, D, S, T, U, Y, Z)$, the standard network process is denoted $\widetilde{\mathbb{X}} = (\tilde{A}, \tilde{D}, \tilde{T}, \tilde{U}, \tilde{Y}, \tilde{Z})$. (Again, we attach tildes to terms specific to standard networks.) Note that $\tilde{S}$ is missing from $\widetilde{\mathbb{X}}$ because there are no setups in the standard network. The equations governing the standard network process are almost identical to those of the setup network. The exceptions are that Equation (16) lacks the $S(t)$ term and Equation (19) is replaced by

(21) additional equations associated with the particular dispatch policy $\tilde{\pi}$.

3.2. Fluid models. Let $\widehat{\mathbb{X}} = (\hat{A}, \hat{D}, \hat{T}, \hat{U}, \hat{Y}, \hat{Z})$ be the formal deterministic analog of the standard network process $\widetilde{\mathbb{X}} = (\tilde{A}, \tilde{D}, \tilde{T}, \tilde{U}, \tilde{Y}, \tilde{Z})$. Consider the following collection of equations:

$$\hat{A}(t) = \alpha t + P' \hat{D}(t), \quad t \geq 0, \tag{22}$$

$$\hat{Z}(t) = \hat{Z}(0) + \hat{A}(t) - \hat{D}(t), \quad t \geq 0, \tag{23}$$

$$\hat{Z}(t) \geq 0, \quad t \geq 0, \tag{24}$$

$$\hat{U}(t) = C\hat{Z}(t), \quad t \geq 0, \tag{25}$$

$$C\hat{T}(t) + \hat{Y}(t) = et, \quad t \geq 0, \tag{26}$$

$$\hat{Y}_j(t) \text{ can increase only if } \hat{U}_j(t) = 0, \quad j = 1, \ldots, J, \tag{27}$$

$$\hat{D}_k(t) = \mu_k \hat{T}_k(t), \quad k = 1, \ldots, K, \tag{28}$$

(29) additional equations associated with the particular dispatch policy $\tilde{\pi}$,

where, as before, e is the J-dimensional column vector of 1s. Equations (22)–(29), which define the standard fluid model, are referred to as the standard fluid model equations. As with the equations describing the setup network and the corresponding standard network, we assume that the components of the processes $\hat{T}$ and $\hat{Y}$ are zero at time zero and are nondecreasing thereafter. Any process $\widehat{\mathbb{X}} = (\hat{A}, \hat{D}, \hat{T}, \hat{U}, \hat{Y}, \hat{Z})$ satisfying (22)–(29) is called a standard fluid model solution. The component processes $\hat{A}$, $\hat{D}$, $\hat{T}$, and $\hat{Z}$ are called the fluid arrival, departure, service allocation, and buffer level processes, respectively. The quantity $\hat{U}_j(t)$ denotes the total amount of fluid at station j at time t. The process $\hat{Y}_j$ is referred to as the server idle time for station j. Standard fluid models and their solutions are fairly well

known; see, for example, Dai (1999). Such models arise from taking fluid limits of standard networks, a topic we explore in the following section.

One can make qualitative comparisons between the standard fluid model and the equations governing the standard network. The major difference is that jobs are (discrete) units of flow in the standard network whereas flow in the fluid model is continuous; hence, the term "fluid." Along these same lines, the nonidling condition in the fluid model states that the station j cumulative idling process $\hat{Y}_j$ cannot increase in the presence of any positive amount of fluid at the station, as opposed to any jobs in the corresponding standard network equation.

Next, we present a generalization of the standard fluid model. Consider the process $\breve{\mathbb{X}} = (\breve{A}, \breve{D}, \breve{T}, \breve{U}, \breve{Y}, \breve{Z})$ and the following set of equations:

$$\breve{A}(t) = \alpha t + P' \breve{D}(t), \quad t \geq 0, \tag{30}$$

$$\breve{Z}(t) = \breve{Z}(0) + \breve{A}(t) - \breve{D}(t), \quad t \geq 0, \tag{31}$$

$$\breve{Z}(t) \geq 0, \quad t \geq 0, \tag{32}$$

$$\breve{U}(t) = C\breve{Z}(t), \quad t \geq 0, \tag{33}$$

$$C\breve{T}(t) + \breve{Y}(t) = et, \quad t \geq 0, \tag{34}$$

$$\breve{Y}_j(t) \text{ can increase only if } \breve{U}_j(t) = 0, \quad j = 1, \ldots, J, \tag{35}$$

$$\breve{D}_k(t_2) - \breve{D}_k(t_1) \leq \mu_k(\breve{T}_k(t_2) - \breve{T}_k(t_1)), \quad 0 \leq t_1 < t_2, \ k = 1, \ldots, K, \tag{36}$$

$$\breve{D}_k(t_2) - \breve{D}_k(t_1) \geq \breve{\mu}_k(\breve{T}_k(t_2) - \breve{T}_k(t_1)) \quad \text{if } \breve{U}_j(s) > 0 \ \forall s \in [t_1, t_2], \quad 0 \leq t_1 < t_2, \ k \in \mathscr{C}(j), \tag{37}$$

$$\text{additional equations associated with the particular production policy } \pi, \tag{38}$$

where $\breve{\mu}_k$ is the setup-adjusted quantity, defined in (8). Equations (30)–(38) are called artificial fluid model equations, and they define the artificial fluid model of the setup network. Any process $\breve{\mathbb{X}} = (\breve{A}, \breve{D}, \breve{T}, \breve{U}, \breve{Y}, \breve{Z})$ satisfying (30)–(38) is called an artificial fluid model solution. As with the standard fluid model, the components of $\breve{T}$ and $\breve{Y}$ are initially zero at time zero and are nondecreasing for all $t > 0$.

Although the connection between the standard fluid model and the standard (discrete) network is straightforward, we cannot claim a direct derivation of the artificial fluid model from a limiting procedure on the setup network; hence the "artificial" moniker. We delay the formal justification of the artificial fluid model until the next section. Nevertheless, it is instructive to consider the salient features of the artificial fluid model. For example, note that the setup allocation process S of the setup network is missing from the artificial fluid model. The component process $\breve{T}$, which parallels the service allocation process T, subsumes the role of the setup allocation process S as well. Because of its dual role, we refer to $\breve{T}$ as the artificial *server* allocation process. The remaining processes, $\breve{A}$, $\breve{D}$, $\breve{Z}$, $\breve{U}$, and $\breve{Y}$, retain their interpretations from the standard fluid model. Next, note that Equation (28) in the standard fluid model is replaced by (36) and (37) in the artificial fluid model. In the standard fluid model, the departure rate of fluid is directly proportional to the allocated server effort. However, in the artificial fluid model, the returns on artificial server effort are not necessarily constant. Condition (36) gives the maximum rate at which allocated server effort is converted into departing units of fluid. This is analogous to avoiding switch-overs in the setup network by processing a single class for an extended period of time. Likewise, condition (37) provides a lower bound on the departure rate of fluid as a function of the rate of artificial server allocation, assuming there is positive fluid at the station. The analogous setup network scenario in this case is that the server is incurring the worse possible setups at the most frequent rate possible for an eligible class. Condition (37) is simpler than (20)

because thresholds disappear in the limit. Finally, replacing (29) with (38) acknowledges that additional standard fluid model equations associated with the dispatch policy $\tilde{\pi}$ may differ from artificial fluid model equations associated with the production policy π.

DEFINITION 3.1. An artificial fluid model is said to be *weakly stable* if for each artificial fluid model solution $\breve{\mathbb{X}}$ with $\breve{Z}(0) = 0$, $\breve{Z}(t) = 0$ for $t \geq 0$.

Weak stability of a standard fluid model can be defined similarly; see, for example, Chen (1995).

3.3. The connection between setup networks and fluid models. The criterion for including an equation in the standard fluid model is that the equation is satisfied by a *fluid limit*. As suggested in the previous section, the inclusion of an equation in the artificial fluid model has an additional step. In this section, we provide the details of both the fluid limits and the additional steps.

A fluid limit of a standard network is obtained through a law-of-large-numbers limiting procedure on the standard network process. Identically, a fluid limit of a setup network is obtained through a law-of-large-numbers limiting procedure on the setup network process. Note that the setup network process $\mathbb{X}$ (respectively, standard network process $\tilde{\mathbb{X}}$) is random, depending on the sample path ω in an underlying probability space. To denote such dependence explicitly, we sometimes use $\mathbb{X}(\cdot, \omega)$ to denote the network process with sample path ω. For an integer d, $\mathbb{D}^d[0, \infty)$ denotes the set of functions x: $[0, \infty) \to \mathbb{R}^d$ that are right continuous on $[0, \infty)$ and have left limits on $(0, \infty)$. An element x in $\mathbb{D}^d[0, \infty)$ is sometimes denoted by $x(\cdot)$ to emphasize that x is a function of time. For each ω, $\mathbb{X}(\omega)$ is an element in $\mathbb{D}^{5K+2J}[0, \infty)$.

For each $r > 0$, define

$$\bar{\mathbb{X}}^r(t, \omega) = r^{-1}\mathbb{X}(rt, \omega), \quad t \geq 0. \tag{39}$$

Again, note that for each $r > 0$, $\bar{\mathbb{X}}^r(\cdot, \omega)$ is an element in $\mathbb{D}^{5K+2J}[0, \infty)$. The scaling in (39) is called the fluid or law-of-large-numbers scaling.

DEFINITION 3.2. A function $\bar{\mathbb{X}} \in \mathbb{D}^{5K+2J}[0, \infty)$ is said to be a *fluid limit* of the setup network if there exists a sequence $r_n \to \infty$ and a sample path ω satisfying (1) such that

$$\lim_{n\to\infty} \bar{\mathbb{X}}^{r_n}(\cdot, \omega) \to \bar{\mathbb{X}}(\cdot),$$

where, throughout this paper, the convergence is interpreted as the uniform convergence on compact sets (u.o.c.).

Uniform convergence on compact sets, as it pertains to fluid limits of networks, is discussed, for example, in Chen and Mandelbaum (1991).

PROPOSITION 3.1. *Take any sample path on which the strong law-of-large-numbers for the primitive processes holds*; *that is,* (1) *holds. For any sequence* $\{r_n\} \subset \mathbb{R}_+$, *with* $r_n \to \infty$ *as* $n \to \infty$, *there exists a subsequence* $\{r_{n_p}\}$, *with* $n_p \to \infty$ *as* $p \to \infty$, *such that*

$$\bar{\mathbb{X}}^{r_{n_p}} = (\bar{A}^{r_{n_p}}, \bar{D}^{r_{n_p}}, \bar{S}^{r_{n_p}}, \bar{T}^{r_{n_p}}, \bar{U}^{r_{n_p}}, \bar{Y}^{r_{n_p}}, \bar{Z}^{r_{n_p}}) \to \bar{\mathbb{X}} = (\bar{A}, \bar{D}, \bar{S}, \bar{T}, \bar{U}, \bar{Y}, \bar{Z}) \quad \text{as } p \to \infty.$$

We refer to the process $\bar{\mathbb{X}}$ as a fluid limit. The existence of fluid limits is well known. A standard argument, like the one in Dai (1995), shows that, for any $r_n \to \infty$ as $n \to \infty$ and almost every sample path ω, there is a subsequence r_{n_p} such that $\bar{S}^{r_{n_p}}(\cdot, \omega)$ and $\bar{T}^{r_{n_p}}(\cdot, \omega)$ converge as $p \to \infty$ and $n_p \to \infty$. Fix an ω that satisfies (1). The convergence of $\bar{T}^{r_n}$, Equation (18), and condition (1) imply that $\bar{D}^{r_n}$ converges. This latter convergence, together with Equation (12) and condition (1), implies that $\bar{A}^{r_n}$ converges. The convergence of other components of $\bar{\mathbb{X}}^{r_n}$ then readily follows. Thus, $\bar{\mathbb{X}}^{r_n}$ converges to a fluid limit as $n \to \infty$.

We now convert the fluid limit $\bar{\mathbb{X}}$ into an artificial fluid "limit" $\breve{\mathbb{X}}$. We use the term "limit" facetiously. In fact, the process $\breve{\mathbb{X}}$ is not a limit at all. Our true intentions are embodied in the following proposition. By the expression $\breve{T} = \bar{S} + \bar{T}$ we mean $\breve{T}(t) = \bar{S}(t) + \bar{T}(t)$ for each $t \geq 0$.

PROPOSITION 3.2. *Given a fluid limit* $\bar{\mathbb{X}} = (\bar{A}, \bar{D}, \bar{S}, \bar{T}, \bar{U}, \bar{Y}, \bar{Z})$ *of a setup network operating under a sensible production policy* π*, the process* $\breve{\mathbb{X}} = (\breve{A}, \breve{D}, \breve{T}, \breve{U}, \breve{Y}, \breve{Z}) = (\bar{A}, \bar{D}, \bar{S} + \bar{T}, \bar{U}, \bar{Y}, \bar{Z})$ *is an artificial fluid model solution.*

PROOF. Fix the fluid limit $\bar{\mathbb{X}}$ and construct $\breve{\mathbb{X}}$ by collapsing the allocation processes $\bar{S}$ and $\bar{T}$ to $\breve{T}$, i.e., $\breve{T} = \bar{S} + \bar{T}$. Equation (36) follows from (18) and the fact that $\breve{T}_k(t_2) - \breve{T}_k(t_1) \geq \bar{T}_k(t_2) - \bar{T}_k(t_1)$ for every $0 \leq t_1 \leq t_2$. As for Equation (37), from the strong law-of-large-numbers (3) (which follows from (1)), for each $k = 1, \ldots, K$ and $0 \leq t_1 < t_2$,

$$\lim_{r_n \to \infty} \sum_{k' \in \mathscr{C}(j)} (\bar{\Upsilon}^{r_n}_{k'k}(\bar{S}^{r_n}_{k'k}(t_2)) - \bar{\Upsilon}^{r_n}_{k'k}(\bar{S}^{r_n}_{k'k}(t_1))) \geq \frac{1}{s_k}(\bar{S}_k(t_2) - \bar{S}_k(t_1)), \tag{40}$$

and, from (18),

$$\bar{D}_k(t_2) - \bar{D}_k(t_1) = \mu_k(\bar{T}_k(t_2) - \bar{T}_k(t_1)) \quad \text{for each } k \in \mathscr{C}(j), \tag{41}$$

whenever $\bar{U}_j(s) > 0$ for each $s \in [t_1, t_2]$. By (20), (40), and (41) we have

$$\bar{S}_k(t_2) - \bar{S}_k(t_1) \leq \frac{s_k}{l_k m_k}(\bar{T}_k(t_2) - \bar{T}_k(t_1)), \tag{42}$$

and, hence, by (8),

$$\begin{aligned} \breve{T}_k(t_2) - \breve{T}_k(t_1) &= \bar{T}_k(t_2) + \bar{S}_k(t_2) - (\bar{T}_k(t_1) + \bar{S}_k(t_1)) \\ &\leq \frac{\breve{m}_k}{m_k}(\bar{T}_k(t_2) - \bar{T}_k(t_1)), \end{aligned} \tag{43}$$

whenever $\bar{U}_j(s) = \breve{U}_j(s) > 0$ for each $s \in [t_1, t_2]$. Equation (37) follows from (41) and (43). Other fluid model equations can be verified as in Dai (1995). □

By now it should be abundantly clear that artificial fluid models are the combination of a law-of-large-numbers limiting procedure, the limitations on setup allocation via sensible policy constraints, and the collapsing of setup and service allocation processes into a single artificial server allocation process.

THEOREM 3.3. *Let a sensible production policy* π *be fixed. If the artificial fluid model is weakly stable, then the corresponding setup network is rate stable.*

PROOF. For standard networks, the analogous result is a simple consequence of Theorem 4.1 of Chen (1995). The only difference here is recognizing that a fluid limit of the setup network is a solution to the artificial fluid model once the allocation processes are collapsed. The remainder of the proof is identical to one for the standard network. See, for example, Dai (1999). □

4. The connection between the artificial and standard fluid models. Our ultimate goal is to craft a sensible production policy π such that the setup network operating under the policy π is rate stable, if at all possible. As stated earlier, this pursuit is possible only if the usual traffic condition (6) holds. It turns out that when (6) holds, coming up with a stabilizing π is always possible; see Theorems 5.8 and 5.9 in the next section.

The standard fluid model has become the conventional tool for demonstrating stability of the standard network operating under a given dispatch policy. The connection between stability of the standard network and its associated fluid model was made concrete through the standard network analog to Theorem 3.3, that is, Theorem 4.1 of Chen (1995). A sizeable body of literature is devoted to investigating standard fluid models, primarily in demonstrating some form of stability. One particularly useful technique for demonstrating stability is via Lyapunov functions. Not surprisingly, a large portion of the literature focuses on finding

the right Lyapunov function to demonstrate stability, given the dispatch policy in question. For example, Bramson (1996, 1997) uses entropy type Lyapunov functions, Down and Meyn (1997) and Dai and VandeVate (2000) advocate piecewise linear Lyapunov functions, and Chen and Zhang (2000) forward linear Lyapunov functions.

Given the similarities between the standard fluid model and the artificial fluid model, we would like to piggyback on these efforts. In particular, it would be ideal if the Lyapunov function that demonstrates the stability of a given standard fluid model also works for the artificial fluid model analog. Along these lines, Dai and Li (2003) analyze standard networks that have batch processing, or *batch processing networks*. They adapt dispatch policies that work well for standard networks into so-called full batch policies for use in scheduling batch processing networks. For the special class of "normal" dispatch policies $\tilde{\pi}$, the fluid model of the standard network operating under $\tilde{\pi}$ is effectively identical to the fluid model of the batch processing network operating under the adapted full batch policy. Hence, Lyapunov functions that prove weak stability of the standard fluid model also work for the batch fluid model. So all of the work in stabilizing a batch processing network is embodied in locating normal policies.

The presence of processing delays due to setups renders the techniques for proving the stability of batch processing networks difficult to transfer to the setup network setting. For one, extending the definition of normal policies would not be fruitful under our framework for production policies. The problem is that, for the artificial fluid model, conversion of allocated server effort to the departure process (embodied in Equations (36) and (37)) is not constant, as it is for a batch fluid model. Accordingly, we require a more hands-on approach.

The use of Lyapunov functions being our main thrust, naturally we should consider more carefully how the functions are generally employed. The following is one method of using a Lyapunov function. For more examples of its usage, see Dai (1999). Given a standard fluid model solution $\widehat{\mathbb{X}} \in \mathbb{D}^{4K+2J}[0, \infty)$: Find a functional L which maps $\widehat{\mathbb{X}}$ to a nonnegative function f such that

- $f(t) = L(\widehat{\mathbb{X}})(t)$ is absolutely continuous in t,
- $f(t) > 0$ if and only if $\hat{Z}(t) \neq 0$,
- $\hat{Z}(t) \neq 0$ implies $(d/dt)f(t) \leq 0$, when the derivative exists.

Points t, for which the derivative of $\widehat{\mathbb{X}}$ exists, are referred to as *regular*. Given such a functional L with the absolute continuity of f, one has $\hat{Z}(t) = 0$ for all t whenever $\hat{Z}(0) = 0$.

Clearly, Lyapunov functions can also be used for showing weak stability of artificial fluid models. In fact, in some instances:

The Lyapunov function that shows a given dispatch policy $\tilde{\pi}$ stabilizes a standard network may also show that a corresponding sensible production policy $\pi = (\theta, \tilde{\pi}, l)$ stabilizes a setup network.

This nebulous statement is the closest we can come to replicating, for general setup networks, the normal policy paradigm for batch fluid models. However, in the following section, we carry out the details with three examples.

5. Examples. We now illustrate the techniques discussed for proving stability under specific sensible production policies. In two of the three examples provided, we take advantage of special network structure in devising our production policies. The third example shows that there is always a stabilizing production policy, provided the usual traffic condition (6) holds.

5.1. FIFO Kelly network. Perhaps the most well-known dispatch policy for standard networks is the first-in-first-out (FIFO) policy. Under FIFO, those jobs that have been present at the station the longest are the first to be processed. Of course, in a setup network, the FIFO policy must be modified to avoid an excessive number of setups.

Even without setups, FIFO is not guaranteed to stabilize standard networks. Indeed, in independent investigations, Bramson (1994a, b) and Seidman (1994) found standard networks that are unstable when operating under the FIFO policy, even under the usual traffic conditions (6). However, there is some hope of stability under FIFO when the network has special structure.

A *standard Kelly network* is a network for which the mean processing times associated with the classes at a given station are identical. That is, for any pair of classes k and k', $m_k = m_{k'}$ if $k, k' \in \mathscr{C}(j)$. It was shown by Bramson (1996) that as long as the traffic intensity is less than one for each station, a standard Kelly network operating under a FIFO dispatch policy is stable. The main result of this section is that a *sensible* FIFO production policy $\pi = (\theta, \text{FIFO}, l)$ is stable in a setup Kelly network when the production runs l_ks are appropriately chosen. Recall that in a sensible production policy, it is assumed that $\theta_k \geq l_k$ for each class k.

THEOREM 5.1. *Consider a setup network whose corresponding standard network is Kelly type*; *that is,* $m_k = m_{k'}$ *for each* $k, k' \in \mathscr{C}(j)$. *Assume that the usual traffic condition* (6) *is satisfied. For sufficiently large* l_k*s, the network operating under a sensible production policy* $(\theta, FIFO, l)$ *is stable.*

Bramson proved that a standard Kelly network operating under the FIFO dispatch policy is stable by showing that the corresponding fluid model is stable. The standard FIFO fluid model (not just for Kelly type) is defined by (22)–(29) with (29) taking the form

$$\hat{D}_k(t + \widehat{W}_j(t)) = \hat{Z}_k(0) + \hat{A}_k(t) \quad \forall\, k \in \mathscr{C}(j), \tag{44}$$

where $\widehat{W}_j(t)$, referred to as the (immediate) *fluid workload* for server j at time t, is defined as

$$\widehat{W}_j(t) \equiv \sum_{k \in \mathscr{C}(j)} m_k \hat{Z}_k(t). \tag{45}$$

In general, *workload* measures the effort required to process the current contents of a given station. As depicted in (45), workload for station j in a standard fluid model is simply the sum of that station's buffer levels, weighted by their associated mean processing times. For standard networks, the workload $\widetilde{W}_j(t)$ is the sum of the residual service times for all jobs currently at station j. For a standard network operating under the FIFO policy, the associated workload process plays as analogous role as in (44): $\tilde{D}_k(t + \widetilde{W}_j(t)) = \tilde{Z}_k(0) + \tilde{A}_k(t)$; see, for example, Harrison and Nguyen (1990). The fluid Equation (44) follows naturally from the functional law of large numbers.

The virtual waiting time process W in setup networks is analogous to the workload process $\widetilde{W}$ in standard networks. That is, in a setup network, the station j virtual waiting time captures the amount of effort required to process all jobs currently residing at station j, where effort includes time allocated to performing setups. This definition extends to fluid limits of setup networks as well as to artificial fluid models in the logical way, with jobs replaced by fluid. We express W explicitly later.

Recall that the artificial fluid model is defined by Equations (30)–(38). Let $\breve{\mathbb{X}} = (\breve{A},\ \breve{D},\ \breve{T},\ \breve{U},\ \breve{Y},\ \breve{Z},\ \breve{W}) = (\bar{A}, \bar{D}, \bar{S} + \bar{T}, \bar{U}, \bar{Y}, \bar{Z}, \bar{W})$ be an artificial fluid model solution. Note the addition of the station-level virtual waiting time components $\breve{W}$ and $\bar{W}$, missing from our previous discussion of fluid limits. In Proposition 5.4 below, we will justify these augmented fluid limits. By Lemma 5.1, which appears later in this section, under the sensible FIFO production policy, Equation (38) takes the form

$$\breve{D}_k(t + \breve{W}_j(t)) = \breve{Z}_k(0) + \breve{A}_k(t) \quad \text{for } t \geq 0,\ \ k = 1, \ldots, K, \tag{46}$$

where $\breve{W}_j(t)$ is the *artificial fluid virtual waiting time* which, by Lemma 5.5 from later in this section, satisfies

$$\sum_{k\in\mathscr{C}(j)} m_k \breve{Z}_k(t) \le \breve{W}_j(t) \le \sum_{k\in\mathscr{C}(j)} \breve{m}_k \breve{Z}_k(t) \tag{47}$$

for each station j. Equation (46) is identical to the standard FIFO fluid model Equation (44). However, due to the setups incurred, Equation (45) becomes the pair of inequalities in (47).

To better understand the upper and lower bounds in (47), consider the role of the workload in the standard network analog to (44). For the standard network, the FIFO policy dictates that in $\widetilde{W}_j(t)$ time units from time t, all jobs currently present at station j will have been processed. The implication is that (1) the server never idles until the present jobs are processed, and (2) jobs that have yet to arrive will not receive service before any job present at time t. For the setup network, the production run portion of the sensible production policy effectively alters the mean processing times, now ranging between m_k and $\breve{m}_k$; see Equation (8). Thus, it is intuitive that the virtual waiting time satisfies (47) in the artificial fluid model.

In Theorem 5.2 to follow, we show that the artificial fluid model for a Kelly-type setup network, operating under a well-chosen sensible FIFO production policy, is weakly stable. Together with Theorem 3.3, this result proves Theorem 5.1. We first introduce the following notation. For a K-dimensional vector a,

$$\vee_j(a) = \max_{k\in\mathscr{C}(j)} a_k \qquad \text{and} \qquad \wedge_j(a) = \min_{k\in\mathscr{C}(j)} a_k. \tag{48}$$

For two positive K-dimensional vectors a and a', define $\Delta(a, a') = \sum_j \Delta_j(a, a')$, where

$$\Delta_j(a, a') = \frac{1}{\wedge_j(a)} - \frac{1}{\vee_j(a')}.$$

Using the mean processing times and their setup-adjusted analogs as parameters, an interpretation is that $\Delta_j(m, \breve{m}) = 1/\wedge_j(m) - 1/\vee_j(\breve{m})$ denotes the difference between the fastest and slowest possible departure rates from station j when there is positive fluid present.

THEOREM 5.2. *Consider a setup network that is of Kelly type with the inflated mean processing times $\breve{m}$ defined in* (8). *There exists a $\delta > 0$ such that, if $\Delta(m, \breve{m}) < \delta$ and*

$$\vee_j(\breve{m}) \sum_{k\in\mathscr{C}(j)} \lambda_k < 1, \tag{49}$$

the corresponding artificial fluid model operating under the sensible FIFO production policy is weakly stable.

To prove Theorem 5.2, we borrow heavily from the proof of Theorem 1 in Bramson (1996) for his *subcritical case*. Subcritical is synonymous to saying the usual traffic condition (5) holds, a condition implied by (49). The structure of Bramson's proof of stability requires the strict Kelly condition (all processing times at a station are equal). However, the production policy effectively alters the mean processing times, ranging between m_k and $\breve{m}_k$. Thus, the strict Kelly condition is violated for the artificial fluid model. Furthermore, conditions (36) and (37) in the artificial fluid model, relating to processing efficiency, differ from the corresponding standard fluid model Equation (28). As a result, significant modifications to Bramson's proof will be carried out here. However, one fortuitous byproduct of our analysis is the extension of Bramson's result to standard networks that are "almost" Kelly, i.e., when $\Delta(m, m)$ is small.

COROLLARY 5.3. *Consider a standard network* (*without setups*). *There exists a* $\delta > 0$ *such that if* $\Delta(m, m) < \delta$ *and*

$$\vee_j(m) \sum_{k \in \mathscr{C}(j)} \lambda_k < 1,$$

the network is stable when operating under the FIFO dispatch policy.

Before beginning the proof of Theorem 5.2, we manipulate Equation (46). Differentiating (46) and taking the sum over $k \in \mathscr{C}(j)$ of both sides, we have

$$(1 + \dot{\breve{W}}_j(t)) \sum_{k \in \mathscr{C}(j)} \dot{\breve{D}}_k(t + \breve{W}_j(t)) = \sum_{k \in \mathscr{C}(j)} \dot{\breve{A}}_k(t). \tag{50}$$

PROOF OF THEOREM 5.2. Again, we will borrow heavily from the proof of Theorem 1 in Bramson (1996). We will only sketch the proof, pointing out, where necessary, where our proof differs from that of Bramson. As in the proof in Bramson, define the following functions:

$$h(x) = x \log x, \quad x \geq 0,$$

$$h_k(x) = \lambda_k h(x/\lambda_k), \quad x \geq 0, \quad k = 1, \ldots, K.$$

For an artificial FIFO fluid model solution $\breve{\mathbb{X}} = (\breve{A}, \breve{D}, \breve{T}, \breve{U}, \breve{Y}, \breve{Z}, \breve{W})$, define

$$\mathscr{H}(t) = \sum_k \int_t^{t+\breve{W}_j(t)} h_k(\dot{\breve{D}}_k(r))\, dr,$$

where k and j together imply $j = \sigma(k)$. The function $\mathscr{H}$ is referred to as the entropy Lyapunov function. In Bramson (1996), it is shown that $\mathscr{H}(t) \geq 0$ for all $t \geq 0$ in Proposition 4.1 and that $\dot{\mathscr{H}}(t) \leq 0$ in Proposition 4.2. We will show that Proposition 4.1 still holds for our system. Although Proposition 4.2 is not necessarily true in our case, we have an alternative approach.

To see that the entropy function is nonnegative, first note that by Lemma 5.2 to follow, $\breve{W}_j(t) > 0$ implies $\breve{W}_j(s) > 0$ for each $s \in [t, t + \breve{W}_j(t))$. By (47), $\breve{U}_j(s) > 0$ for each $s \in [t, t + \breve{W}_j(t))$. By conditions (35), (37), and (49) we have

$$\breve{W}_j(t) > 0 \qquad \text{implies} \qquad \sum_{k \in \mathscr{C}(j)} \frac{\dot{\breve{D}}_k(s)}{\lambda_j^{\Sigma}} \geq \sum_{k \in \mathscr{C}(j)} \frac{\breve{m}_k}{\vee_j(\breve{m})} \frac{\dot{\breve{D}}_k(s)}{\lambda_j^{\Sigma}} \geq 1 \quad \forall\, s \in [t, t + \breve{W}_j(t)),$$

where $\lambda_j^{\Sigma} = \sum_{k \in \mathscr{C}(j)} \lambda_k$. Hence, the left-hand side of Equation (4.5) of Bramson is nonnegative almost surely and Proposition 4.1 of Bramson holds.

Now we provide an alternative to Proposition 4.2 of Bramson. Starting at Equation (4.6) of Bramson, considerable effort is spent dealing with the derivative of the entropy function. We employ the same strategy. We start with the thread of reasoning at Equation (4.8) of Bramson, which we repeat here:

$$\dot{\mathscr{H}}(t) = \sum_k \big[h_k\big(\dot{\breve{A}}_k(t)\big) - h_k\big(\dot{\breve{D}}_k(t)\big)\big] - \sum_j \Bigg(\sum_{k \in \mathscr{C}(j)} \dot{\breve{A}}_k(t) \Bigg) \log(1 + \dot{\breve{W}}_j(t)). \tag{51}$$

The last term is the focus of Equation (4.9) and Lemmas 4.1 and 5.1 of Bramson. The first term is the focus of his Lemma 4.2 and requires no adjustment. That is, Equation (4.12) of Bramson, which reads

$$\sum_k \big[h_k\big(\dot{\breve{A}}_k(t)\big) - h_k\big(\dot{\breve{D}}_k(t)\big)\big] \leq \sum_k \dot{\breve{Z}}_k(t) \tag{52}$$

still holds.

Consider the last term of (51). By (50) and Lemma 5.7 to follow, for each station j, we have

$$\Big(\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t)\Big)\log(1+\dot{\breve{W}}_j(t)) = \gamma_j(t)(1+\dot{\breve{W}}_j(t))\log(1+\dot{\breve{W}}_j(t)) = \gamma_j(t)\, h(1+\dot{\breve{W}}_j(t)), \tag{53}$$

where

$$\gamma_j(t) = \begin{cases} \vee_j(\mu) & \text{if } \sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) = 0 \text{ and } \breve{W}_j(t) > 0, \\ \sum_{k\in\mathscr{C}(j)} \dot{\breve{D}}_k(t+\breve{W}_j(t)) & \text{otherwise.} \end{cases} \tag{54}$$

Using identical arguments from Lemma 5.1 of Bramson, we have

$$\gamma_j(t)\, h(1+\dot{\breve{W}}_j(t)) \geq \gamma_j(t)\big[B_{13}\big(\dot{\breve{W}}_j(t)\big)^2 + \dot{\breve{W}}_j(t)\big] \tag{55}$$

for some appropriate $B_{13} > 0$ as in (5.3) of Bramson. By Equations (31), (50), and (54) and Lemma 5.7 to follow,

$$\gamma_j(t)\,\dot{\breve{W}}_j(t) = \sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) - \gamma_j(t) = \sum_{k\in\mathscr{C}(j)} [\dot{\breve{Z}}_k(t) + \dot{\breve{D}}_k(t)] - \gamma_j(t). \tag{56}$$

Consider the case when $\breve{W}_j(t) > 0$. By (54) and Lemma 5.6,

$$\sum_{k\in\mathscr{C}(j)} \dot{\breve{D}}_k(t) - \gamma_j(t) \geq -\Delta_j(m, \breve{m}). \tag{57}$$

By (54), condition (57) also holds when $\breve{W}_j(t) = 0$. It follows from Equations (53), (55), (56), and (57) that

$$\sum_j \Big(\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t)\Big)\log(1+\dot{\breve{W}}_j(t)) \geq \sum_k \dot{\breve{Z}}_k(t) - \Delta(m, \breve{m}) + B_7 \sum_j (\dot{\breve{W}}_j(t))^2, \tag{58}$$

where $B_7 = B_{13} \min_k \breve{\mu}_k$. Bringing (51), (52), and (58) together we have an analog to (4.22) of Bramson:

$$\dot{\mathscr{H}}(t) \leq -B_7(\dot{\breve{W}}_j(t))^2 + \Delta(m, \breve{m}) \quad \text{for each station } j. \tag{59}$$

As in §3 of Bramson, we define $\tau_j(t)$ so that $t' = t + \tau_j(t)$ is the first time $t' \geq t$ at which the virtual waiting time at station j is zero. It is shown in (3.6) of Bramson that $\tau_j(t) \leq B_1 \breve{W}^M(t)$, where $\breve{W}^M(t) = \max_j \breve{W}_j(t)$ is the largest virtual waiting time among the stations and B_1 is some constant. As is argued in Bramson for both Equation (4.23) and the equation that follows it, for any station j,

$$\mathscr{H}(t) - \mathscr{H}(t+\tau_j(t)) \geq \frac{B_7(\breve{W}_j(t))^2}{B_1 \breve{W}^M(t)} - B_1 \breve{W}^M(t)\Delta(m, \breve{m}).$$

It follows that if j is the index of the station with the largest virtual waiting time, then

$$\mathscr{H}(t) - \mathscr{H}(t+\tau_j(t)) \geq B_1(B_9 - \Delta(m, \breve{m}))\breve{W}^M(t),$$

where

$$B_9 = B_7/(B_1)^2 \tag{60}$$

is the δ referred to in the statement of the theorem. Hence, we assume that $\Delta(m, \check{m}) < B_9$. Now suppose that the entropy function is positive at time t. We immediately know that the virtual waiting time must also be positive. The previous equation argues that the entropy function will decrease by some value proportional to the virtual waiting time in an amount of time also proportional to the virtual waiting time. We can argue then that there is a sequence of times $\{t_i\}$ such that t_i is the first emptying time after t_{i-1} of the largest virtual waiting time station at time t_{i-1}, $t_i \le t_{i-1} + B_1 \check{W}^M(t_{i-1})$, and

$$\mathcal{H}(0) - \mathcal{H}(t_i) \ge (B_9 - \Delta(m, \check{m}))t_i,$$

which is analogous to the equation just before (4.25) in Bramson. Because $\mathcal{H}(t_i) \ge 0$, $t_i \le \mathcal{H}(0)/(B_9 - \Delta(m))$ for each i. Hence, $\{t_i\}$ is a nondecreasing sequence that has an upper bound, and thus has a limit, which is denoted by t_∞. Because the empty time is at least the current virtual waiting time,

$$t_i - t_{i-1} \ge \check{W}_j(t_{i-1}) = \check{W}^M(t_{i-1}).$$

Taking limit as $i \to \infty$ and using the continuity of $\check{W}^M$, we have $\check{W}^M(t_\infty) = 0$. By the same argument as in (4.27) of Bramson, the entropy is bounded by a factor of the largest virtual waiting time function: $\mathcal{H}(t) \le B_{10}\check{W}^M(t)$. Hence, $\mathcal{H}(0) \le B_{10}\check{W}(0)$ so that the hitting time t_∞ to zero is bounded by $B_{11}\check{W}(0)$ for some constant B_{11}.

Because the original entropy function in Bramson is monotonic, once the function hits zero, it remains there. What is left to show is that our almost Kelly system behaves the same way. Suppose there exists some $t > t_\infty$ such that $\mathcal{H}(t) > 0$. It must be the case that $\check{W}(t) > 0$ as well. Define ϵ to be the minimum of $\check{W}(t)/2$ and $(B_9/B_{10})\check{W}(t)$ and define $s_\epsilon = \sup\{s \in (t_\infty, t]\colon \check{W}(s) \le \epsilon\}$. That is, within the interval $(s_\epsilon, t]$ the virtual waiting time function is strictly larger than ϵ. Note that by our choice of ϵ, the quantity s_ϵ must be strictly less than t. We know that $\mathcal{H}(s_\epsilon) \le B_{10}\check{W}(s_\epsilon) \le B_9\check{W}(t)$ and that $\mathcal{H}$ hits zero within $\mathcal{H}(s_\epsilon)/B_9 \le \check{W}(t)$ time units after s_ϵ. That is, for some $r \in (s_\epsilon, s_\epsilon + \check{W}(t)]$, $\mathcal{H}(r) = 0$. Note that this time is strictly less than $t + \check{W}(t)$. By Lemma 5.2, the virtual waiting time $\check{W}$ cannot decrease faster than at rate 1. Recall that $\check{W}$ is greater than ϵ from s_ϵ until t. So $\check{W}$ must be positive until time $t + \check{W}(t)$. This is a contradiction. So our premise that there is some time t such that $\check{W}(t) > 0$ cannot be true. This concludes the proof. □

Equation (46) and inequalities (47) for the artificial FIFO fluid model are justified through fluid limits. For that, we need to properly define $W_j(t)$, the station j virtual waiting time process for the setup network. The definition should capture the amount of time needed to process most of the jobs residing at the station. The fact that jobs are not processed in a strict FIFO order leads to some difficulty in crafting a useful definition. One must consider that jobs are processed in production runs, where the launching of the run is determined by the first job in the run. Moreover, if some class k has fewer jobs present than dictated by the threshold θ_k, that class may be ignored. The following quantity tracks the amount of time required to process the number of class k jobs present, in excess of the threshold θ_k:

$$\tau_k(t) = \inf\{s \ge 0\colon D_k(t+s) - D_k(t) \ge Z_k(t) - \theta_k + 1\}. \tag{61}$$

Recall that under a sensible production policy, any class that has fewer than θ_k jobs is considered to be "empty." Thus, we ignore the last $\theta_k - 1$ jobs because the server may ignore the jobs until their numbers reach the threshold. That is, we cannot bound their sojourn. We define the station j virtual waiting time as the maximum of these values for constituent classes: $W_j(t) = \max_{k \in \mathcal{C}(j)} \tau_k(t)$.

PROPOSITION 5.4. *Take any sample path on which the strong law-of-large-numbers for the primitive processes holds*; *that is,* (1) *holds. For any sequence* $\{r_n\} \subset \mathbb{R}_+$ *with* $r_n \to \infty$ *as* $n \to \infty$, *there exists a subsequence* $\{r_{n_p}\}$, *with* $n_p \to \infty$ *as* $p \to \infty$, *such that*

$$\begin{aligned}\bar{\mathbb{X}}^{r_{n_p}} &= (\bar{A}^{r_{n_p}}, \bar{D}^{r_{n_p}}, \bar{S}^{r_{n_p}}, \bar{T}^{r_{n_p}}, \bar{U}^{r_{n_p}}, \bar{Y}^{r_{n_p}}, \bar{Z}^{r_{n_p}}, \bar{W}^{r_{n_p}}) \to \bar{\mathbb{X}} \\ &= (\bar{A}, \bar{D}, \bar{S}, \bar{T}, \bar{U}, \bar{Y}, \bar{Z}, \bar{W}) \quad \text{as } n_p \to \infty.\end{aligned}$$

Moreover, the process $\breve{\mathbb{X}} = (\breve{A}, \breve{D}, \breve{T}, \breve{U}, \breve{Y}, \breve{Z}, \breve{W}) = (\bar{A}, \bar{D}, \bar{S} + \bar{T}, \bar{U}, \bar{Y}, \bar{Z}, \bar{W})$ *is an artificial fluid model solution*; *that is,* $\breve{\mathbb{X}}$ *satisfies Equations* (30)–(38). *The processes* $\bar{\mathbb{X}}$ *and* $\breve{\mathbb{X}}$ *are absolutely continuous.*

The convergence is assumed to be uniform on compact intervals in $\mathbb{D}^{5K+3J}[0, \infty)$. We delay the proof until the appendix. Each limit $\bar{\mathbb{X}}$ is said to be a fluid limit for the FIFO setup network. Readers should note the extra virtual waiting time component in both the scaled queueing processes $\bar{\mathbb{X}}^{r_{n_p}}$ as well as the limiting process $\bar{\mathbb{X}}$ and the process $\breve{\mathbb{X}}$. For the remainder of this section, let $\bar{\mathbb{X}}$ be a fluid limit and $\breve{\mathbb{X}}$ be its corresponding artificial fluid model solution.

The following lemma justifies the FIFO-specific additional artificial fluid model Equation (46).

LEMMA 5.1. *For station* j *and constituents* $k \in \mathcal{C}(j)$,

$$\breve{D}_k(t + \breve{W}_j(t)) = \breve{Z}_k(0) + \breve{A}_k(t), \quad t \geq 0.$$

PROOF. We first show that the setup network process obeys the following inequalities:

$$Z_k(0) + A_k(t) - \theta_k + 1 \leq D_k(t + W_j(t)) \leq Z_k(0) + A_k(t) + l_k - 1, \quad k = 1, \ldots, K, \tag{62}$$

which clearly imply the result by taking the fluid limit. The first inequality of (62) follows from the observation that

$$D_k(t + W_j(t)) - D_k(t) \geq D_k(t + \tau_k(t)) - D_k(t) \geq Z_k(t) - \theta_k + 1,$$

where $\tau_k(\cdot)$ is defined in (61). Clearly, for each station j, $W_j(t) \geq \tau_k(t)$ for all $t \geq 0$ and $k \in \mathcal{C}(j)$. For the second inequality of (62), note that, for any class k production run that completes by time $t + \tau_k(t)$, at least one of these jobs must have been present by time t. Furthermore, no other class k jobs can be served in the interval $[t + \tau_k(t), t + W_j(t)]$, leading to the second inequality. □

To justify additional inequalities (47) for the FIFO artificial fluid model, we need a few lemmas. The following lemma states that the artificial fluid virtual waiting time process has a lower bound on its rate of descent. The process $e(t)$ is J-dimensional, where each component process $e_j(t) = t$.

LEMMA 5.2. *The components of the process* $e(\cdot) + \breve{W}(\cdot)$ *are nondecreasing.*

PROOF. We consider the components individually; that is, we will show that $t + \breve{W}_j(t)$ is a nondecreasing function. Fix j. From the definition of τ_k, note that, for any k and $s, t \geq 0$,

$$D_k(t + s + \tau_k(t + s)) \geq Z_k(0) + A_k(t + s) \geq Z_k(0) + A_k(t)$$

and, hence, $s + \tau_k(t + s) \geq \tau_k(t)$. For some $k' \in \mathcal{C}(j)$, $\tau_{k'}(t) = W_j(t)$. It follows that $s + t + W_j(t + s) \geq s + t + \tau_{k'}(t + s) \geq t + \tau_{k'}(t) = t + W_j(t)$. This shows that the function $t + W_j(t)$ is nondecreasing in t. Thus, the fluid limit $t + \bar{W}_j(t)$ and the artificial process $t + \breve{W}_j(t)$ are also nondecreasing. □

A consequence of the previous result is that there is strictly positive virtual waiting time throughout any nontrivial interval $[t, t+\breve{W}_j(t))$. The following result states that a positive amount of virtual waiting time coupled with a positive rate of flow at regular time t makes the previous interval a closed one.

LEMMA 5.3. *For any station j and regular point $t > 0$, if $\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) > 0$ and $\breve{W}_j(t) > 0$, then $\breve{W}_j(t+\breve{W}_j(t)) > 0$.*

PROOF. Fix $t > 0$ and assume it is a regular point. From Lemma 5.2, $\dot{\breve{W}}_j(s) \geq -1$ for all regular $s \in [t, t+\breve{W}_j(t))$. For some $k \in \mathscr{C}(j)$, $\dot{\breve{A}}_k(t) > 0$. There exists a $\delta > 0$ such that for all $0 < h < \delta$, $\breve{A}_k(t+h) - \breve{A}_k(t) > \dot{\breve{A}}_k(t)h/2$. Choose any $h < \min(\delta, W_j(t))$. By Lemma 5.1,

$$\breve{D}_k(t+h+\breve{W}_j(t+h)) = \breve{Z}_k(0) + \breve{A}_k(t+h) > \breve{Z}_k(0) + \breve{A}_k(t) + \dot{\breve{A}}_k(t)h/2$$

and

$$\breve{D}_k(t+h+\breve{W}_j(t+h)) - \breve{D}_k(t+\breve{W}_j(t)) > \dot{\breve{A}}_k(t)h/2 > 0.$$

By the monotonicity of $\breve{D}_k$, $\breve{W}_j(t+h) > \breve{W}_j(t) - h$, thus $t < t+h < t+\breve{W}_j(t) < t+h+\breve{W}_j(t+h)$. Because $\breve{W}_j(t+h) > 0$ for sufficiently small h, and $\breve{W}_j(s) > 0$ for $s \in [t+h, t+h+\breve{W}_j(t+h))$, we have $\breve{W}_j(t+\breve{W}_j(t)) > 0$. □

The nonidling condition of the artificial fluid model (35) provides a lower bound on service allocation when there is positive fluid at a given station. The following lemma provides the analogous consequence when there is positive virtual waiting time at a given station.

LEMMA 5.4. *For any station j,*

$$\breve{W}_j(t) > 0 \quad \textit{implies} \quad \dot{\breve{Y}}_j(t) = 0.$$

PROOF. Let $\{r_n\}$ be a sequence of positive real numbers associated with the fluid limit $\overline{\mathbb{X}}$ and the corresponding artificial fluid model solution $\breve{\mathbb{X}}$, where $r_n \to \infty$ as $n \to \infty$. Suppose $\overline{W}_j(t) = \breve{W}_j(t) > 0$. By the continuity of $\breve{W}_j$, there exists a $\delta > 0$ and an $N \geq 0$ such that

$$W_j(r_n s) \geq r_n \breve{W}_j(t)/2 \quad \forall\, s \in [t-\delta, t+\delta] \text{ and } n \geq N.$$

Fix an $n \geq N$. For each $s \in [r_n(t-\delta), r_n(t+\delta)]$, $W_j(s) > 0$. By the definition of the virtual waiting time process W_j, for each $s \in [r_n(t-\delta), r_n(t+\delta)]$, there exists a $k \in \mathscr{C}(j)$ such that $Z_k(s) \geq \theta_k$. That is to say, there is always an eligible class throughout the interval. Thus, throughout the interval $[r_n(t-\delta), r_n(t+\delta)]$, the server never idles. Namely, $Y_j(r_n(t+\delta)) - Y_j(r_n(t-\delta)) = 0$. Taking limits, we have $\overline{Y}_j(t+\delta) - \overline{Y}_j(t-\delta) = \breve{Y}_j(t+\delta) - \breve{Y}_j(t-\delta) = 0$. □

We are now equipped to justify the bounds for the artificial virtual waiting time process as presented in (47).

LEMMA 5.5. *For each station j and $t \geq 0$,*

$$\sum_{k\in\mathscr{C}(j)} m_k \breve{Z}_k(t) \leq \breve{W}_j(t) \leq \sum_{k\in\mathscr{C}(j)} \breve{m}_k \breve{Z}_k(t).$$

PROOF. Fix $t \geq 0$. As for the first inequality, by Lemma 5.1 and (36),

$$\begin{aligned}\sum_{k\in\mathscr{C}(j)} m_k \breve{Z}_k(t) &= \sum_{k\in\mathscr{C}(j)} m_k (\breve{Z}_k(0) + \breve{A}_k(t) - \breve{D}_k(t)) \\ &= \sum_{k\in\mathscr{C}(j)} m_k (\breve{D}_k(t+\breve{W}_j(t)) - \breve{D}_k(t))\end{aligned}$$

$$\leq \sum_{k\in\mathscr{C}(j)} \breve{T}_k(t+\breve{W}_j(t)) - \breve{T}_k(t)$$
$$\leq \breve{W}_j(t).$$

To prove the second inequality of the result, we first note that it holds trivially when $\breve{W}_j(t) = 0$. Assume that $\breve{W}_j(t) > 0$. It follows from Lemmas 5.2 and 5.4 that $\sum_{k\in\mathscr{C}(j)} \breve{m}_k \dot{\breve{D}}_k(s) \geq 1$ for almost every $s \in [t, t+\breve{W}_j(t))$. Thus, we have

$$\sum_{k\in\mathscr{C}(j)} \breve{m}_k \breve{Z}_k(t) = \sum_{k\in\mathscr{C}(j)} \breve{m}_k(\breve{D}_k(t+\breve{W}_j(t)) - \breve{D}_k(t))$$
$$= \int_t^{t+\breve{W}_j(t)} \sum_{k\in\mathscr{C}(j)} \breve{m}_k \dot{\breve{D}}_k(s)\, ds \geq \breve{W}_j(t). \quad \square$$

Finally, we provide a result directly required in the proof of Theorem 5.2.

LEMMA 5.6. *Fix station j and regular point $t > 0$. If $\breve{W}_j(t) > 0$, then*

$$\frac{1}{\vee_j(\breve{m})} \leq \sum_{k\in\mathscr{C}(j)} \dot{\breve{D}}_k(t) \leq \frac{1}{\wedge_j}(m). \tag{63}$$

Moreover, if $\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) > 0$, then

$$\frac{1}{\vee_j(\breve{m})} \leq \sum_{k\in\mathscr{C}(j)} \dot{\breve{D}}_k(t+\breve{W}_j(t)) \leq \frac{1}{\wedge_j(m)}. \tag{64}$$

PROOF. Fix j and the regular point $t > 0$. The upper bounds of both (63) and (64) hold by (36) and the fact that $\breve{U}_j$ is nondecreasing. By Lemma 5.5, $\breve{U}_j(t) > 0$ and, hence, the lower bound in (63) follows from (37). By Lemmas 5.3 and 5.5 and (37), if $\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) > 0$, then $\breve{U}_j(t+\breve{W}_j(t)) > 0$ and the lower bound in (64) holds. $\square$

For a standard fluid network, computing the derivative of the workload function is a simple matter; see (45). For our setup network, the derivative of the virtual waiting time is generally unknown. However, when the rate of arrivals to a station is zero and the virtual waiting time is positive, the virtual waiting time decreases as fast as possible, that is at rate one. The following lemma is used in Theorem 5.2.

LEMMA 5.7. *Assume that $\breve{W}_j(t) > 0$ for some station j. If $\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) = 0$, then $\dot{\breve{W}}_j(t) = -1$.*

PROOF. Suppose that, on the contrary, $\sum_{k\in\mathscr{C}(j)} \dot{\breve{A}}_k(t) = 0$ and $\dot{\breve{W}}_j(t) = -1 + 2\epsilon$ for some $\epsilon > 0$. There exists a $\delta > 0$ such that for each $h < \delta$,

$$\breve{W}_j(t+h) - \breve{W}_j(t) > -(1-\epsilon)h. \tag{65}$$

By (65) and Lemma 5.2, $\breve{W}_j(s) > 0$ for each $s \in [t, t+\breve{W}_j(t)+\epsilon h)$. By Lemma 5.6,

$$\sum_{k\in\mathscr{C}(j)} \dot{\breve{D}}_k(s) \geq \wedge_j(\breve{\mu}), \quad s \in [t, t+\breve{W}_j(t)+\epsilon h). \tag{66}$$

Hence, the left-hand side of (50) is greater than $2\epsilon(\wedge_j(\breve{\mu}))$, but the right-hand side is equal to zero, a contradiction. $\square$

5.2. Early steps first. Consider a family of standard networks where each job follows some deterministic *route*, or sequence of buffers, through the network. Suppose that for each buffer there is only one manner in which jobs arrive, exogenously or from one (upstream) buffer. Buffers that receive jobs exogenously are referred to as *sources*. Let $\mathscr{E}$ denote the set of all source buffers and $\|\mathscr{E}\|$ denote the number of elements of $\mathscr{E}$. Without loss of generality, $\alpha_k > 0$ for each $k \in \mathscr{E}$. Jobs that enter the system through a common buffer k are said to belong to the same job *type* and follow the same route of buffers through the network. We refer to such networks as multitype queueing networks, or *standard multitype networks*. (Bertsimas et al. 2002 provide performance analysis of standard multitype networks.) When there is one source of jobs ($\|\mathscr{E}\| = 1$), the network is said to be a *reentrant line*.

For a multitype network, the corresponding transition matrix P and arrival rate vector α exhibit special structure. If some element of α is nonzero, then the corresponding column of P has all zeros. Otherwise, $\alpha_k = 0$ and exactly one element of the corresponding column of P is one and the other elements are all zero.

To facilitate the description of the *early-steps-first* (ESF) dispatch policy we alter the notation slightly. In the new notation, classes are denoted by their (type, step) pair. The types are indexed $q = 1, \ldots, \|\mathscr{E}\|$ and the steps are indexed $k = 1, \ldots, K_q$, where K_q denotes the length of the type q route. All of the class-specific quantities now have this alternative notation. For example, $m_{(q,k)}$ denotes the class (q, k) mean processing time and $Z_{(q,k)}(t)$ records the number of class (q, k) jobs at time t. In a slight abuse of notation, we replace the arrival rate of type q jobs $\alpha_{(q,1)}$ with the quantity α_q.

We are now equipped to describe the ESF dispatch policy. Suppose the server at station j requires dispatching at time t. Among all of the nonempty buffers at the station, the server will be dispatched to a buffer (q, k), where k is the first nonempty step at the station. Any other constituent buffer $(q', k') \in \mathscr{C}(j)$ with $k' < k$, must therefore be empty. Note that it is possible that station j houses multiple classes that are the kth step for their respective routes. The policy does not explicitly say how to choose among such classes; that is, ties are broken arbitrarily.

THEOREM 5.5. *Under the usual traffic condition* (6)*, a standard multitype network, operating under the ESF dispatch policy, is rate stable.*

A proof of Theorem 5.5 might use the following linear Lyapunov function:

$$L(t) = \sum_{(q,k)} \beta^+_{(q,k)} \hat{Z}_{(q,k)}(t), \tag{67}$$

where $\beta^+_{(q,k)}$ is defined recursively in the following way:

$$\beta_{(q,k)} = \begin{cases} m_{(q,k)}\left(1 + \dfrac{\gamma_{j,k}}{1-\rho_j}\right), & k = 1, \ldots, K_q, \\ 0, & k = K_q + 1, \end{cases} \tag{68}$$

$$\gamma_{j,k} = \sum_{(q,k) \in \mathscr{C}(j,k)} \alpha_q \beta^+_{(q,k+1)}, \tag{69}$$

and

$$\beta^+_{(q,k)} = \sum_{k'=k}^{K_q} \beta_{(q,k')}. \tag{70}$$

The set $\mathscr{C}(j,k) = \{(q,k') \in \mathscr{C}(j)\colon k' = k\}$, used in (69), contains the constituent classes of station j that are composed of the kth step of some route. The proof of Theorem 5.5 is implied by the proof of Theorem 5.6 below.

Multitype setup networks have the same routing and arrival characteristics as their standard multitype network counterparts. We can adapt the ESF dispatch policy to form a sensible ESF production policy $\pi = (\theta, \text{ESF}, l)$. The policy evolves as follows: When a server at station j is free for dispatching, we create a list of eligible constituent buffers. The server will be dispatched to a buffer (q, k), where k is the earliest step with an eligible buffer at the station. Any other constituent buffer $(q', k') \in \mathscr{C}(j)$ with $k' < k$, must be ineligible. Assume that buffer (q, k) passed the more stringent eligibility test, so that, if (q', k') is ineligible, it must be the case that $Z_{(q', k')}(t) < \theta_{(q', k')}$. If the server is dispatched to buffer (q, k), a setup is performed (if necessary) for class (q, k) and then $l_{(q, k)}$ jobs are processed in a row before the server is freed for subsequent dispatching. As in the standard multitype network setting, there may be more than one eligible buffer from step k at station j. Any arbitrarily chosen tie-breaking scheme will yield the same stability results.

THEOREM 5.6. *A multitype setup network operating under a sensible ESF production policy $\pi = (\theta, ESF, l)$ is rate stable.*

Before we start the proof, we first state the artificial fluid model Equation (38) that corresponds to the ESF production policy. The justification of the equation is stated in the following lemma, whose proof is delayed until the appendix.

LEMMA 5.8. *Under the sensible ESF production policy $\pi = (\theta, ESF, l)$, the artificial fluid model Equation* (38) *takes the form*

$$\sum_{k=1}^{k'} \sum_{(q,k)\in\mathscr{C}(j)} \breve{Z}_{(q,k)}(t) > 0 \quad \textit{implies} \quad \sum_{k=1}^{k'} \sum_{(q,k)\in\mathscr{C}(j)} \dot{\breve{T}}_{(q,k)}(t) = 1 \tag{71}$$

for every step $k' \geq 1$ and each station $j = 1, \ldots, J$.

PROOF OF THEOREM 5.6. Let $\breve{\mathbb{X}}$ be a solution to the artificial fluid model operating under a sensible ESF production policy. We adapt the linear Lyapunov function devised for the ESF dispatch policy to obtain

$$L(t) = \sum_{(q,k)} \beta^{+}_{(q,k)} \breve{Z}_{(q,k)}(t), \tag{72}$$

where $\beta^{+}_{(q,k)}$ is defined recursively in the following way:

$$\beta_{q,k} = \begin{cases} \breve{m}_{(q,k)}\left(1 + \dfrac{\gamma_{j,k}}{1 - \breve{\rho}_j}\right), & k = 1, \ldots, K_q, \\ 0, & k = K_q + 1, \end{cases} \tag{73}$$

$$\gamma_{j,k} = \sum_{(q,k)\in\mathscr{C}(j,k)} \alpha_q \beta^{+}_{(q,k+1)}, \tag{74}$$

and

$$\beta^{+}_{(q,k)} = \sum_{k'=k}^{K_q} \beta_{(q,k')}. \tag{75}$$

Note that with the exception of the $\breve{m}_{(q,k)}$ term replacing $m_{(q,k)}$ in (73), Equations (72)–(75) are identical to (67)–(70).

Assume that $\breve{Z}(t) \neq 0$ and $\breve{\mathbb{X}}$ is differentiable at time t. Let k be the first systemwide, nonempty step. That is, there is some class (q, k) such that $\breve{Z}_{(q,k)}(t) > 0$ and $\breve{Z}_{(q,k')}(t) = 0$

for all classes (q, k') such that $k' < k$. Fix step k and time t for the remainder of the proof. We investigate the derivative of the Lyapunov function in (72):

$$\begin{aligned}\dot{L}(t) &= \sum_{(q',k')} \beta^+_{(q',k')} \dot{\breve{Z}}_{(q',k')}(t) = \sum_{q'=1}^{|\mathscr{E}|} \sum_{k'=1}^{K_{q'}} \beta^+_{(q',k')} \dot{\breve{Z}}_{(q',k')}(t) \\ &= \sum_{q'=1}^{|\mathscr{E}|} \sum_{k'=1}^{K_{q'}} \beta^+_{(q',k')} (\dot{\breve{D}}_{(q',k'-1)}(t) - \dot{\breve{D}}_{(q',k')}(t)) \\ &= \sum_{q'=1}^{|\mathscr{E}|} \left[\alpha_{q'} \beta^+_{(q',1)} - \sum_{k'=1}^{K_{q'}} (\beta^+_{(q',k')} - \beta^+_{(q',k'+1)}) \dot{\breve{D}}_{(q',k')}(t) \right] \\ &= \sum_{q'=1}^{|\mathscr{E}|} \left[\alpha_{q'} \beta^+_{(q',1)} - \sum_{k'=1}^{K_{q'}} \beta_{(q',k')} \dot{\breve{D}}_{(q',k')}(t), \right],\end{aligned}$$

where $\breve{D}_{(q',0)}(t) \equiv \alpha_{q'} t$. By the nonnegativity of $\breve{Z}$, $\breve{Z}_{(q',k')}(t) = 0$ implies $\dot{\breve{Z}}_{(q',k')}(t) = 0$. Because k is the first nonempty step, $\dot{\breve{D}}_{(q',k')}(t) = \alpha_{q'}$ for all classes (q', k') such that $k' < k$ and the derivative of the $L(t)$ equals

$$\begin{aligned}\dot{L}(t) &= \sum_{q'=1}^{|\mathscr{E}|} \left[\alpha_{q'} \beta^+_{(q',k)} - \sum_{k'=k}^{K_{q'}} \beta_{(q',k')} \dot{\breve{D}}_{(q',k')}(t) \right] \\ &\le \sum_{j:\, \mathscr{C}(j,k) \neq \varnothing} \left[\sum_{(q,k) \in \mathscr{C}(j,k)} \alpha_q \beta^+_{(q,k)} - \sum_{(q,k) \in \mathscr{C}(j,k)} \beta_{(q,k)} \dot{\breve{D}}_{(q,k)}(t) \right],\end{aligned} \tag{76}$$

where, again, $\mathscr{C}(j,k) = \{(q,k') \in \mathscr{C}(j)\colon k' = k\}$ denotes the constituent classes of station j that are composed of the kth step of some route. The transition embodied in (76) has two subtleties. For one, for each class (q', k') with step $k' \ge k$, the quantity $\dot{\breve{D}}_{(q',k')}(t)$ appears on the left-hand side of the inequality, whereas the quantity appears on the right-hand side for only those classes with step k. Second, each nonzero $\alpha_{q'} \beta^+_{q',k'}$ term on the left-hand side of the inequality appears on the right-hand side as well. The terms that are equal to zero correspond to job types with strictly fewer than k steps. The nonzero values correspond to routes that have at least k steps. Moreover, the kth step must occur at one of the stations. Hence, on the right-hand side we can exclude those stations without a resident class that is the kth step of some route. By (37), Lemma 5.8, and the fact that $\dot{\breve{D}}_{(q,k')}(t) = \alpha_q$ if $k' < K$, we have

$$\sum_{(q,k) \in \mathscr{C}(j,k)} \dot{\breve{T}}_{(q,k)}(t) = 1 - \sum_{k'=1}^{k-1} \sum_{(q,k') \in \mathscr{C}(j,k')} \dot{\breve{T}}_{(q,k')}(t) \ge 1 - \sum_{k'=1}^{k-1} \sum_{(q,k') \in \mathscr{C}(j,k')} \alpha_q \breve{m}_{(q,k')}. \tag{77}$$

Hence, by (37), (75), and (76),

$$\dot{L}(t) \le \sum_{j:\, \mathscr{C}(j,k) \neq \varnothing} \left[\sum_{(q,k) \in \mathscr{C}(j,k)} \alpha_q \beta_{(q,k)} + \sum_{(q,k) \in \mathscr{C}(j,k)} \alpha_q \beta^+_{(q,k+1)} - \sum_{(q,k) \in \mathscr{C}(j,k)} \beta_{(q,k)} \breve{\mu}_{(q,k)} \dot{\breve{T}}_{(q,k)}(t) \right].$$

It follows from (10), (73), (74), and (77) that

$$\begin{aligned}\dot{L}(t) \le \sum_{j:\, \mathscr{C}(j,k) \neq \varnothing} \Bigg[& \sum_{(q,k) \in \mathscr{C}(j,k)} \alpha_q \breve{m}_{(q,k)} \left(1 + \frac{\gamma_{j,k}}{1 - \breve{\rho}_j} \right) + \gamma_{j,k} \\ & - \left(1 + \frac{\gamma_{j,k}}{1 - \breve{\rho}_j} \right) \left(1 - \sum_{k'=1}^{k-1} \sum_{(q,k') \in \mathscr{C}(j,k')} \alpha_q \breve{m}_{(q,k')} \right) \Bigg]\end{aligned}$$

$$= \sum_{j:\,\mathscr{C}(j,k)\neq\varnothing} \left[\gamma_{j,k} - \left(1 + \frac{\gamma_{j,k}}{1-\breve{\rho}_j}\right)\left(1 - \sum_{k'=1}^{k} \sum_{(q,k')\in\mathscr{C}(j,k')} \alpha_q \breve{m}_{(q,k')}\right)\right]$$
$$\leq \sum_{j:\,\mathscr{C}(j,k)\neq\varnothing} -(1-\breve{\rho}_j).$$

The derivative of the Lyapunov function is negative as long as there is positive fluid. Hence, the artificial fluid model is weakly stable. □

5.3. Generalized round robin. The generalized round robin (GRR) dispatch policy is often discussed in the context of single server systems subject to setups, or *polling systems*; see, for example, Takagi (1986, 1990). For standard networks, GRR is parameterized by a set of strictly positive reals $\beta = (\beta_k,\ k = 1, \ldots, K)$. When the constants are integers, the policy works as follows: At station j, the server "visits" the constituent buffers in $\mathscr{C}(j)$ in a fixed cyclic order; hence, the name round robin. In polling system literature, the order in which classes are visited is referred to as the *polling table*, a term we adopt as well. When the server visits buffer k, β_k jobs are processed, if possible. Otherwise, the buffer is exhausted and the server moves on to the next buffer. In this sense, the β_ks can be thought of as nominal allocations. The span of time, from the beginning of the visit to the first buffer in the polling table to the completion of the visit to the last buffer in the table, is referred to as a *cycle*.

When the β_ks are not integers, the spirit of the dispatch policy is the same. However, some requisite bookkeeping is in order. Consider the nth cycle of the server at station j. Let $a_k(n)$ denote the integer-valued nominal allocation for each class $k \in \mathscr{C}(j)$ and $b_k(n)$ denote the nominal residual allocation. The quantities are defined recursively:

$$a_k(n+1) = \lfloor b_k(n) + \beta_k \rfloor, \tag{78}$$

$$b_k(n+1) = b_k(n) + \beta_k - a_k(n+1) \tag{79}$$

for $n = 0, 1, \ldots$, where $b_k(0) = 0$ and, as before, $\lfloor x \rfloor$ denotes the integer part of x. When the server visits buffer k for the nth time, it processes $a_k(n)$ jobs, if possible, before moving on to the next buffer.

For any vector of positive constants β, the additional standard fluid model Equation (29) takes the form

$$\dot{\hat{D}}_k(t) \geq \frac{\beta_k}{\sum_{k'\in\mathscr{C}(j)} \beta_{k'} m_{k'}}, \quad k = 1, \ldots, K, \tag{80}$$

for each t such that $\hat{D}_k(t)$ is differentiable and $\hat{Z}_k(t) > 0$, where, as is our convention, $j = \sigma(k)$. The following theorem is proved in Dai (1999) when elements of the vector β are integers. The more general case (where the elements are reals) is a consequence of Theorem 5.8.

THEOREM 5.7. *Under the usual traffic conditions* (6)*, a standard network operating under a generalized round robin policy parameterized by β is stable if, for each $k = 1, \ldots, K$,*

$$\frac{\beta_k}{\sum_{k'\in\mathscr{C}(j)} \beta_{k'} m_{k'}} \geq \lambda_k. \tag{81}$$

We now describe how the GRR dispatch policy for standard networks is adapted to form the sensible GRR production policy for setup networks, denoted $\pi = (\theta, \mathrm{GRR}(\beta), l)$. Consider station j. As with the standard network, the server at station j visits the classes in $\mathscr{C}(j)$ in a round robin fashion. In fact, we use the same nominal values computed in (78)

and (79). However, the interpretations of $a_k(n)$ and $b_k(n)$ are slightly different. Suppose the server is conducting its nth cycle and buffer k is being considered for dispatching. If $a_k(n) \geq 1$ and class k is eligible according to the criterion in §2.3, then a setup for class k is performed. Otherwise the server moves on to the next buffer. The only time a server does not move is when there is an absence of jobs available for processing at the station. Assuming the server has performed a setup for buffer k, $a_k(n)$ determines how many production runs of length l_k the server will perform.

THEOREM 5.8. *A setup network operating under a sensible generalized round robin production policy $\pi = (\theta, \mathrm{GRR}(\beta), l)$ is rate stable if, for each $k = 1, \ldots, K$, $j = \sigma(k)$,*

$$\frac{\beta_k l_k}{\sum_{k' \in \mathscr{C}(j)} \beta_{k'} l_{k'} \breve{m}_{k'}} \geq \lambda_k. \tag{82}$$

PROOF. Let $\breve{\mathbb{X}}$ be an artificial fluid model solution. By Lemma 5.9 that immediately follows, if (82) holds, we have $\dot{\breve{D}}_k(t) \geq \lambda_k$ for any t such that $\breve{Z}_k(t) > 0$ and $\breve{\mathbb{X}}$ is differentiable at t. It follows from a slight modification of Theorem 4 of Bramson (1998) that $\breve{Z}(t) = 0$ for $t \geq 0$. Thus, the artificial fluid model is weakly stable. Rate stability of the setup network follows from Theorem 3.3. □

The following is key in the proof of Theorem 5.8.

LEMMA 5.9. *Consider a setup network operating under a sensible generalized round robin production policy $\pi = (\theta, \mathrm{GRR}(\beta), l)$. Artificial fluid solutions obey the following*:

$$\dot{\breve{D}}_k(t) \geq \frac{\beta_k l_k}{\sum_{k' \in \mathscr{C}(j)} \beta_{k'} l_{k'} \breve{m}_{k'}}, \quad k = 1, \ldots, K. \tag{83}$$

When the β_ks are strictly positive integers, the intuition behind (83) is straightforward. The typical cycle length on average would be at most $\sum_{k' \in \mathscr{C}(j)} (s_{k'} + \beta_{k'} l_{k'} m_{k'}) \leq \sum_{k' \in \mathscr{C}(j)} \beta_{k'} l_{k'} \breve{m}_{k'}$. When there are enough class k jobs present, the average time a server spends processing those jobs in a given cycle is $\beta_k l_k m_k$. Equation (83) is simply the ratio of the two quantities. The formal proof is delayed until the appendix in §7.

A convenience of GRR is that one can always construct vectors β and l such that (82) is true when the usual traffic condition (6) holds:

THEOREM 5.9. *Whenever the usual traffic condition* (6) *holds, there exists a K-dimensional vector l of strictly positive integers and a K-dimensional vector β of strictly positive reals such that* (82) *holds.*

PROOF. The proof does not use the full flexibility from having two vectors, β and l, to manipulate. Instead we simply set all elements of β to 1. For each k, set $r = 1/\min_k(\lambda_k s_j^{\Sigma})$, $\bar{l}_k = (r+1)\lambda_k s_j^{\Sigma}/(1-\rho_j)$, and $l_k = \lfloor \bar{l}_k \rfloor$, where $s_j^{\Sigma} = \sum_{k \in \mathscr{C}(j)} s_k$. We have

$$\sum_{k \in \mathscr{C}(j)} \bar{l}_k \breve{m}_k = \frac{(r+1)s_j^{\Sigma}\rho_j}{1-\rho_j} + s_j^{\Sigma} = \frac{(1+r\rho_j)s_j^{\Sigma}}{1-\rho_j}$$

so that

$$\begin{aligned} \frac{\beta_k l_k}{\sum_{k' \in \mathscr{C}(j)} \beta_{k'} l_{k'} \breve{m}_{k'}} &> \frac{\bar{l}_k - 1}{\sum_{k' \in \mathscr{C}(j)} \bar{l}_{k'} \breve{m}_{k'}} \\ &= \frac{s_j^{\Sigma}(r+1)\lambda_k - (1-\rho_j)}{s_j^{\Sigma}(1+r\rho_j)} = \lambda_k + \frac{(\lambda_k s_j^{\Sigma} r - 1)(1-\rho_j)}{s_j^{\Sigma}(1+r\rho_j)} \geq \lambda_k. \quad \square \end{aligned}$$

So far we have adhered strictly to the sensible policy paradigm; in particular, we assumed (7) holds. However, the structure of generalized round robin allows us to deviate slightly. The primary justification of the thresholds θ is that they ensure no class is neglected by the server indefinitely. Examining the proof of Lemma 5.9, which provides the key step in proving Theorem 5.8, the lemma continues to holds when $\theta = 0$. Thus, we have the following corollary.

COROLLARY 5.10. *A setup network operating under a generalized round robin production policy* $\pi = (0, \mathrm{GRR}(\beta), l)$ *is rate stable if* (82) *holds for each* $k = 1, \ldots, K$.

6. Concluding remarks. Our sensible production policy requires that conditions (7) and (10) are satisfied. As pointed out in Corollary 5.10, the threshold condition (7) can be relaxed for some production policies. Condition (10) can be relaxed as well. Recall that $s_k = \max_{k' \in \mathscr{C}(j)} s_{k'k}$. It is used to define amortized average processing time in (8), which in turn is used to define condition (10). The quantity s_k is an upper bound on the mean setup times. For a particular family of production policies, it is possible that the actual relevant mean setup times are much smaller. For example, for the GRR production policy with zero thresholds, servers, for the most part, visit and perform setups for classes in the same fixed sequence each cycle. A natural definition is then $s_k = s_{p(k)k}$, where $p(k)$ denotes the predecessor of class k in the cycle. The only exception to the setup sequence occurs when the server encounters an empty buffer and, hence, does not perform a setup. This is not a concern if for each station j, $s_{k_1k_3} \le s_{k_1k_2} + s_{k_2k_3}$ for classes $k_1, k_2, k_3 \in \mathscr{C}(j)$. The condition, which can be thought of as a triangle inequality for sequence-dependent setups, is reasonable in most cases.

Even with s_k reduced, there is still further room in the refinement of the amortized mean processing time $\breve{m}$ defined in (8). Again consider the family of GRR policies in §5.3. Suppose $\beta_k > 1$ for some class k. Then, for that class k and some cycle n, $a_k(n) \ge 2$. The result is that at least two production runs of length l_k may be conducted in sequence, without a setup in between. We should redefine the setup-adjusted mean service time to reflect this change in the frequency of class k setups: $\breve{m}_k = m_k + s_k/(l_k \max(\beta_k, 1))$ for each k. The form of (82) is the same, but the cycle length, expressed in the denominator of the left-hand side, is smaller. Hence, the condition is relaxed.

The main concern of this paper is the stability of queueing networks with setups. Having established stability of some networks under some sensible policies, one may want to now turn the attention to matters of system performance. Indeed, this is the motivation of Warren's dissertation (1997), where a heuristic approach is employed, and the work of Bertsimas and Nino-Mora (1999), where bounds on optimal holding costs are obtained using the achievable region approach.

7. Appendix.

7.1. The existence of artificial fluid virtual waiting time.

PROOF OF PROPOSITION 5.4. From Proposition 3.1, there exists a subsequence $\{r_{n_p}\} \subset \mathbb{R}_+$, with $n_p \to \infty$ as $p \to \infty$, such that

$$(\bar{A}^{r_{n_p}}, \bar{D}^{r_{n_p}}, \bar{S}^{r_{n_p}}, \bar{T}^{r_{n_p}}, \bar{U}^{r_{n_p}}, \bar{Y}^{r_{n_p}}, \bar{Z}^{r_{n_p}}) \to (\bar{A}, \bar{D}, \bar{S}, \bar{T}, \bar{U}, \bar{Y}, \bar{Z}) \quad \text{as } p \to \infty.$$

To complete the proof, it suffices to show that $\bar{W}^{r_{n_p}}$ is asymptotically Lipschitz continuous, i.e., there exists an $L > 0$ such that for any $s, t \ge 0$ and $j = 1, \ldots, J$,

$$\limsup_{n_p \to \infty} \left| \bar{W}_j^{r_{n_p}}(t+s) - \bar{W}_j^{r_{n_p}}(t) \right| \le L\, s. \tag{84}$$

Condition (84) implies that $\bar{W}^{n_p}$ converges along a further subsequence $\{n'_p\}$. Thus, $\bar{\mathbb{X}}^{n'_p} \to \bar{\mathbb{X}} = (\bar{A}, \bar{D}, \bar{T}, \bar{U}, \bar{Y}, \bar{Z}, \bar{W})$, which is said to be a fluid limit.

Because, by Lemma 5.2, $e(\cdot) + W(\cdot)$ is nondecreasing (in each component) for any $t, s \geq 0$, we have

$$W_j(t+s) - W_j(t) \geq -s,$$

which implies that

$$\liminf_{n_p \to \infty} \left(\bar{W}_j^{r_{n_p}}(t+s) - \bar{W}_j^{r_{n_p}}(t) \right) \geq -s. \tag{85}$$

We now provide an upper bound on the change in the virtual waiting time. Let $\eta_k = \{\eta_k(n),\ n \geq 1\}$ denote the sequence of processing times of class k jobs, i.e., it takes $\eta_k(n)$ units of time for the server to process the nth class k job. From the definition of W_j, we have

$$W_j(t) \geq \sum_{k \in \mathscr{C}(j)} \sum_{i=D_k(t)+1}^{Z_k(0)+A_k(t)-\theta_k+1} \eta_k(i) + \sum_{k \in \mathscr{C}(j)} S_k(t + W_j(t)) - S_k(t)$$

and

$$W_j(t+s) \leq \sum_{k \in \mathscr{C}(j)} \sum_{i=D_k(t+s)+1}^{Z_k(0)+A_k(t+s)+l_k-1} \eta_k(i) + \sum_{k \in \mathscr{C}(j)} S_k(t+s+W_j(t+s)) - S_k(t+s).$$

Because S_k is nondecreasing,

$$\begin{aligned} W_j(t+s) - W_j(t) \leq & \sum_{k \in \mathscr{C}(j)} \sum_{i=Z_k(0)+A_k(t)-\theta_k+2}^{Z_k(0)+A_k(t+s)+l_k-1} \eta_k(i) \\ & + \sum_{k \in \mathscr{C}(j)} S_k(t+s+W_j(t+s)) - S_k(t+W_j(t)). \end{aligned} \tag{86}$$

To derive an estimate of $S_k(t+s+W_j(t+s)) - S_k(t+W_j(t))$, let $N_{k'k}(t)$ denote the cumulative number of setups from class k' to class k by time t and $\zeta_{k'k} = \{\zeta_{k'k}(n),\ n \geq 1\}$ denote the sequence of (k', k) setup times. Then,

$$S_k(t+s+W_j(t+s)) - S_k(t+W_j(t)) \leq \sum_{k' \in \mathscr{C}(j)} \sum_{i=N_{k'k}(t+W_j(t))}^{N_{k',k}(t+s+W_j(t+s))} \zeta_{k'k}(i).$$

Service completions initiate setup times. From the definition of the virtual waiting time, class k jobs through job number $Z_k(0) + A_k(t) - \theta_k + 1$ cannot have an effect on the cumulative setup time S_k beyond time $t + W_j(t)$. Moreover, at most $Z_k(0) + A_k(t+s) + l_k - 1$ class k jobs can have an effect on S_k up to time $t + s + W_j(t+s)$. Not surprisingly, during the interval $[t + W_j(t), t + s + W_j(t+s)]$, the maximum number of class k jobs that trigger setups is precisely the maximum number of jobs processed, $A_k(t+s) - A_k(t) + \theta_k + l_k - 2$. Thus,

$$S_k(t+s+W_j(t+s)) - S_k(t+W_j(t)) \leq \sum_{k' \in \mathscr{C}(j)} \sum_{i=N_{k'k}(t+W_j(t))}^{N_{k',k}(t+W_j(t))+A_k(t+s)-A_k(t)+\theta_k+l_k-2} \zeta_{k'k}(i). \tag{87}$$

Because

$$\frac{1}{r_{n_p}} \sum_{i=N_{k'k}(r_{n_p}t+W_j(r_{n_p}t))}^{N_{k',k}(r_{n_p}t+W_j(r_{n_p}t))+A_k(r_{n_p}t+r_{n_p}s)-A_k(r_{n_p}t)+\theta_k+l_k-2} \zeta_{k'k}(i) \to s_{k'k}(\bar{A}_k(t+s) - \bar{A}_k(t)),$$

by (86) and (87), we have

$$\limsup_{n_p \to \infty} \left(\bar{W}_j^{r_{n_p}}(t+s) - \bar{W}_j^{r_{n_p}}(t) \right) \leq \sum_{k \in \mathscr{C}(j)} \left(m_k + \sum_{k' \in \mathscr{C}(j)} s_{k'k}(\bar{A}_k(t+s) - \bar{A}_k(t)) \right). \tag{88}$$

Because $\bar{A}$ is Lipschitz continuous, (84) follows from (85) and (88). □

7.2. Proofs of Lemmas 5.8 and 5.9.

PROOF OF LEMMA 5.8. Let $\bar{\mathbb{X}}$ be a fluid limit of a setup network operating under the sensible ESF production policy $\pi = (\theta, \text{ESF}, l)$. Let $\breve{\mathbb{X}}$ be the associated artificial fluid limit, as constructed in Proposition 3.2. Let $\omega \in \Omega$ be a sample path on which (1) holds and $\{r_n\}$ be a sequence of positive reals such that $r_n \to \infty$ and $\bar{\mathbb{X}}^{r_n}(\cdot, \omega) \to \bar{\mathbb{X}}(\cdot)$ u.o.c. as $n \to \infty$. Fix station j and a time $t > 0$ such that $\bar{\mathbb{X}}(t)$ and $\breve{\mathbb{X}}(t)$ are differentiable. Suppose that, for some fixed step k_0, we have

$$\sum_{k=1}^{k_0} \sum_{(q,k)\in\mathscr{C}(j)} \breve{Z}_{(q,k)}(t) > 0.$$

Then, for some class $(q_1, k_1) \in \mathscr{C}(j)$ with $k_1 \le k_0$, we have $\breve{Z}_{(q_1,k_1)}(t) = \bar{Z}_{(q_1,k_1)}(t) > 0$. By the continuity of $\bar{Z}$, and the uniform convergence $\bar{Z}^{r_n} \to \bar{Z}$ as $n \to \infty$, there exists a $\delta > 0$ and an integer N such that, for each $n \ge N$,

$$Z_{(q_1,k_1)}(s) \ge \theta_{(q_1,k_1)} \quad \forall\, s \in [r_n t, r_n(t+\delta)].$$

This condition ensures that, throughout the interval $[r_n t, r_n(t+\delta)]$, the servers at station j are never dispatched to a class $(q,k) \in \mathscr{C}(j)$ such that $k > k_1$. Equivalently, for each class (q,k) with $k > k_1$,

$$T_{(q,k)}(r_n(t+\delta)) - T_{(q,k)}(r_n t) \le V_{(q,k)}(M^n_{(q,k)} + l_{(q,k)}) - V_{(q,k)}(M^n_{(q,k)}) \tag{89}$$

and

$$\begin{aligned} &S_{(q,k)}(r_n(t+\delta)) - S_{(q,k)}(r_n t) \\ &\quad \le \sum_{(q',k')\in\mathscr{C}(j)} F_{(q',k')(q,k)}(R^n_{(q',k')(q,k)} + 1) - F_{(q',k')(q,k)}(R^n_{(q',k')(q,k)}), \end{aligned} \tag{90}$$

where $M^n_{(q,k)}$ is the number of class (q,k) jobs processed by time $r_n t$ and $R^n_{(q',k')(q,k)}$ is the number of setups from class (q',k') to class (q,k) performed by time $r_n t$. By Lemma 7.1 to follow, the law-of-large-numbers (1), and (89) and (90),

$$\begin{aligned} &\sum_{k'>k_1} \sum_{(q',k')\in\mathscr{C}(j)} \left[\breve{T}_{(q',k')}(t+\delta) - \breve{T}_{(q',k')}(t)\right] \\ &\quad = \lim_{n\to\infty} (1/r_n) \sum_{k'>k_1} \sum_{(q',k')\in\mathscr{C}(j)} \left[S_{(q',k')}(r_n(t+\delta)) - S_{(q',k')}(r_n t)\right. \\ &\qquad\qquad \left. + T_{(q',k')}(r_n(t+\delta)) - T_{(q',k')}(r_n t)\right] = 0. \end{aligned} \tag{91}$$

By (34), (35), and (91),

$$\sum_{k'=1}^{k_1} \sum_{(q',k')\in\mathscr{C}(j)} \left[\breve{T}_{(q',k')}(t+\delta) - \breve{T}_{(q',k')}(t)\right] = \delta.$$

We obtain the result by dividing by δ and letting $\delta \downarrow 0$. □

PROOF OF LEMMA 5.9. Let $\bar{\mathbb{X}}$ be a fluid limit of a setup network operating under the sensible generalized round robin production policy $\pi = (\theta, \text{GRR}(\beta), l)$. Let $\breve{\mathbb{X}}$ be the associated artificial fluid limit, as constructed in Proposition 3.2. Let $\omega \in \Omega$ be a sample path on which (1) holds and $\{r_n\}$ be a sequence of positive reals such that $r_n \to \infty$ and $\bar{\mathbb{X}}^{r_n}(\cdot, \omega) \to \bar{\mathbb{X}}(\cdot)$ as $n \to \infty$. At time t, suppose that for some class k at station j we have $\breve{Z}_k(t) = \bar{Z}_k(t) > 0$. We would like to show that

$$\dot{\breve{D}}_k(t) = \dot{\bar{D}}_k(t) \ge \frac{\beta_k l_k}{\sum_{k'\in\mathscr{C}(j)} \beta_{k'} l_{k'} \breve{m}_{k'}}. \tag{92}$$

By the continuity of $\bar{Z}$ we know there exists an $\epsilon > 0$ and a $\delta > 0$ such that $\bar{Z}_k(s) > \epsilon$ for each $s \in [t, t+\delta]$. For large enough n, $Z_k(s) \geq \theta_k$ for each $s \in [r_n t, r_n(t+\delta)]$. Hence, if the ith cycle takes place entirely within the interval $[r_n t, r_n(t+\delta)]$, the server processes exactly $a_k(i)l_k$ class k jobs during this cycle. It should be clear by (78) and (79) that, for any class k and positive integers p and q,

$$q\beta_k - 1 < \sum_{i=p+1}^{p+q} a_k(i) < q\beta_k + 1. \tag{93}$$

For each n, we refer to cycles that start after $r_n t$ and end before $r_n(t+\delta)$ as *complete*. Let N^n denote the number of complete cycles. The result (92) will follow if we can show that, on almost every sample path,

$$\lim_{n\to\infty} \frac{D_k(r_n(t+\delta)) - D_k(r_n t)}{N^n} \to \beta_k l_k \tag{94}$$

and that

$$\limsup_{n\to\infty} \frac{r_n \delta}{N^n} \leq \frac{1}{p_j} \sum_{k' \in \mathscr{C}(j)} \beta_{k'} l_{k'} \check{m}_{k'}. \tag{95}$$

Note that, in addition to the N^n complete cycles, there may be two *incomplete* cycles. The station j servers may have been in the middle of a cycle at time $r_n t$ and again at time $r_n(t+\delta)$. By (93), we can bound the number of jobs processed in $[r_n t, r_n(t+\delta)]$,

$$(N^n\beta_k - 1)l_k < D_k(r_n(t+\delta)) - D_k(r_n t) < ((N^n+2)\beta_k + 1)l_k. \tag{96}$$

By Lemma 7.2 to follow, $N^n \to \infty$. Dividing both sides of (96) by N^n and letting $n \to \infty$ yields (94).

Now we demonstrate that (95) holds. Clearly, the server does not idle during $[r_n t, r_n(t+\delta)]$ so that

$$r_n \delta = \sum_{k' \in \mathscr{C}(j)} [(T_{k'}(r_n(t+\delta)) - T_{k'}(r_n t)) + (S_{k'}(r_n(t+\delta)) - S_{k'}(r_n t))].$$

We investigate how the server effort throughout the interval $[r_n t, r_n(t+\delta)]$ is allocated. Let $M^n_{k'}$ denote the number of class k' jobs that have completed service at time $r_n t$. We can bound the total time dedicated to class k'.

$$T_{k'}(r_n(t+\delta)) - T_{k'}(r_n t) \leq V_{k'}(M^n_{k'} + \lceil (N^n+2)\beta_{k'} + 1 \rceil l_{k'}) - V_{k'}(M^n_{k'}), \tag{97}$$

where, as before, $\eta_{k'}(i) = V_{k'}(i) - V_{k'}(i-1)$. Similarly,

$$S_{k''k'}(r_n(t+\delta)) - S_{k''k'}(r_n t) \leq F_{k''k'}(R^n_{k''k'} + 2 + \lambda^n_{k''k'}) - F_{k''k'}(R^n_{k''k'}), \tag{98}$$

where $\lambda^n_{k''k'}$ is the number of type (k'', k') setups among the N^n complete cycles and $R^n_{k''k'}$ is the number of completed type (k'', k') setups before time $r_n t$. The reasoning behind (97) is as follows. As argued earlier, there are at most $N^n + 2$ cycles, where N^n of them are complete. We can assume the most extreme case, that for each class k' and in each cycle i, exactly $a_{k'}(i)l_{k'}$ jobs are processed. Using Equation (93) we bound the total number of processed jobs with $((N^n+2)\beta_{k'} + 1)l_{k'}$. It should be clear that $\sum_{k'' \in \mathscr{C}(j)} \lambda_{k''k'} \leq N^n(\max(\beta_{k'}, 1)) + 1$. Dividing both sides of Equations (97) and (98) by N^n yields

$$\limsup_{n\to\infty} (1/N^n)[T_{k'}(r_n(t+\delta)) - T_{k'}(r_n t)] \leq \beta_{k'} l_{k'} m_{k'}$$

and

$$\limsup_{n\to\infty} (1/N^n)[S_{k'}(r_n(t+\delta)) - S_{k'}(r_n t)] \leq \beta_{k'} s_{k'}.$$

Finally, by (8),

$$\beta_{k'} l_{k'} m_{k'} + \beta_{k'} s_{k'} = \beta_{k'} l_{k'} \breve{m}_{k'}$$

and, hence, (95) follows. □

The following lemma is used to prove Lemma 5.8 and Lemma 7.2 to follow.

LEMMA 7.1. *Consider any setup network. Let $\{r_n\}$ be a sequence of positive reals such that $r_n \to \infty$ as $n \to \infty$. Let M_k^n be the number of class k jobs processed by time $r_n t$ and $R_{k'k}^n$ be the number of type (k', k) setups performed by time $r_n t$. We have, on almost every sample path,*

$$\limsup_{n\to\infty} M_k^n / r_n < \infty$$

and

$$\limsup_{n\to\infty} R_{k'k}^n / r_n < \infty$$

for each $k, k' = 1, \ldots, K$.

PROOF. We prove the first result for class k jobs. The result for setups is similar. Suppose $\limsup_{n\to\infty} M_k^n / r_n = \infty$. Without loss of generality, $M_k^n / r_n \to \infty$. By definition,

$$V_k(M_k^n) \le r_n t.$$

Dividing both sides by M_k^n and taking the limit yields $m_k \le 0$, a contradiction. □

The following is used to prove Lemma 5.9.

LEMMA 7.2. *Suppose the setup network is operating under a sensible generalized round robin policy. Let $\{r_n\}$ be a sequence of positive reals such that $r_n \to \infty$ as $n \to \infty$. Fix time t, station j, and the constant $\delta > 0$. Suppose that, for large enough n, servers at station j never idle during the interval $[r_n t, r_n(t+\delta)]$. Let N^n denote the number of cycles that transpire completely during $[r_n t, r_n(t+\delta)]$. We have*

$$\liminf_{n\to\infty} N^n / r_n > 0. \tag{99}$$

PROOF. The server at station j provides $r_n \delta$ units of potential effort during the interval $[r_n t, r_n(t+\delta)]$. All of this effort goes to processing jobs or performing setups. Because there are K classes, there are at most $K^2 - K$ types of setups. This yields a maximum of K^2 activities over which the $r_n \delta$ units of server effort is distributed over the interval. There is a subsequence r_{n_q} with $n_q \to \infty$ as $q \to \infty$ such that one of the activities receives at least $r_{n_q} p_j \delta / K^2$ units of server effort during the interval for each q. Assume the activity of processing class k jobs receives this amount of effort and, without loss of generality, that this occurs for each number in the original sequence $\{r_n\}$. That is,

$$V_k(M_k^n + \lceil (N^n + 2)\beta_k + 1 \rceil l_k) - V_k(M_k^n) \ge r_n \delta / K^2, \tag{100}$$

where M_k^n denotes the number of class k jobs completed by time $r_n t$. (The case that a setup activity receives this amount of effort can be argued similarly.) The terms in (100) are identical to those in (97) of the previous proof. If $N^n / r_n \to 0$ then, by Lemma 7.1, dividing both sides of (100) by N^n yields $\delta \le 0$. Hence, (99) must hold. □

Acknowledgments. The research of the first author was supported in part by National Science Foundation Grants DMI-9457336, DMI-9813345, and DMI-0300599; by a Chinese National Science Foundation grant; and by TLI-Asia Pacific, a partnership between National University of Singapore and Georgia Institute of Technology.

References

Andradóttir, S., H. Ayhan, D. G. Down. 2003. Dynamic server allocation for queueing networks with flexible servers. *Oper. Res.* **51** 952–968.

Bertsimas, D., J. Nino-Mora. 1999. Optimization of multiclass queueing networks with changeover times via the achievable region approach: Part II, the multistation case. *Math. Oper. Res.* **24** 331–361.

Bertsimas, D., D. Gamarnik, J. Tsitsiklis. 2002. Performance of multiclass Markovian queueing networks via piecewise linear Lyapunov functions. *Ann. Appl. Probab.* **11** 1384–1428.

Bramson, M. 1994a. Instability of FIFO queueing networks. *Ann. Appl. Probab.* **4** 414–431.

Bramson, M. 1994b. Instability of FIFO queueing networks with quick service times. *Ann. Appl. Probab.* **4** 693–718.

Bramson, M. 1996. Convergence to equilibria for fluid models of FIFO queueing networks. *Queueing Systems: Theory Appl.* **22** 5–45.

Bramson, M. 1997. Convergence to equilibria for fluid models of head-of-the-line proportional processor sharing queueing networks. *Queueing Systems: Theory Appl.* **23** 1–26.

Bramson, M. 1998. Stability of two families of queueing networks and a discussion of fluid limits. *Queueing Systems: Theory Appl.* **28** 7–31.

Chen, H. 1995. Fluid approximations and stability of multiclass queueing networks: Work-conserving disciplines. *Ann. Appl. Probab.* **5** 637–665.

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

Chen, H., H. Zhang. 2000. Stability of multiclass queueing networks under priority service disciplines. *Oper. Res.* **48** 26–37.

Cooper, R. B., S. C. Niu, M. M. Srinivasan. 1998. When does forced idle time improve performance in polling models? *Management Sci.* **44** 1079–1086.

Dai, J. G. 1995. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Ann. Appl. Probab.* **5** 49–77.

Dai, J. G. 1999. Stability of fluid and stochastic processing networks. Miscellanea Publication, No. 9. Centre for Mathematical Physics and Stochastics, Aarhus, Denmark. http://www.maphysto.dk/.

Dai, J. G., O. Jennings. 2003. Stochastic scheduling and set-ups in manufacturing systems. D. D. Yao, H. Zhang, X. Y. Zhou, eds. *Stochastic Models and Optimization*. Springer, New York, 193–243.

Dai, J. G., C. Li. 2003. Stabilizing batch processing networks. *Oper. Res.* **51** 123–136.

Dai, J. G., J. VandeVate. 2000. The stability of two-station multi-type fluid networks. *Oper. Res.* **48** 721–744.

Down, D., S. P. Meyn. 1997. Piecewise linear test functions for stability and instability of queueing networks. *Queueing Systems: Theory Appl.* **27** 205–226.

El-Taha, M., S. Stidham, Jr. 1999. *Sample-Path Analysis of Queueing Systems*. Kluwer, Boston, MA.

Gershwin, S. B. 1995. Stochastic scheduling and set-ups in manufacturing systems. *Internat. J. Production Res.* **33** 1849–1870.

Harrison, J. M. 1988. Brownian models of queueing networks with heterogeneous customer populations. W. Fleming, P.-L. Lions, eds. *Stochastic Differential Systems, Stochastic Control Theory and Applications*, *IMA Volumes in Mathematics and Its Applications*, Vol. 10. Springer-Verlag, New York, 147–186.

Harrison, J. M., V. Nguyen. 1990. The QNET method for two-moment analysis of open queueing networks. *Queueing Systems: Theory Appl.* **6** 1–32.

Jennings, O. B. 2000. Multiclass queueing networks with setup delays: Stability analysis and heavy traffic approximation. Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Kumar, P. R., T. I. Seidman. 1990. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. Automatic Control* **AC-35** 289–298.

Lan, W.-M., T. L. Olsen. 2004. Multi-product systems with both setup times and costs: Fluid bounds and schedules. *Oper. Res.* Submitted.

Perkins, J. R., P. R. Kumar. 1989. Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems. *IEEE Trans. Automatic Control* **AC-34** 139–148.

Seidman, T. I. 1994. 'First come, first served' can be unstable! *IEEE Trans. Automatic Control* **39** 2166–2171.

Takagi, H. 1986. *Analysis of Polling Systems*. MIT Press, Cambridge, MA.

Takagi, H. 1990. Queueing analysis of polling systems. H. Takagi, ed. *Stochastic Analysis of Computer and Communication Systems*. North-Holland, Amsterdam, The Netherlands, 267–318.

Warren, G. M. H. 1997. Analysis of some fluid models and a queueing network analyzer for a polling system. Ph.D. thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.