Target-Pursuing Scheduling and Routing Policies for Multiclass Queueing Networks

Ioannis Ch. Paschalidis, *Member, IEEE*, Chang Su, and Michael C. Caramanis, *Member, IEEE*

Abstract—**We propose a parametric class of myopic scheduling and routing policies for open and closed multiclass queueing networks. In open networks, they steer the state of the system toward a predetermined and fixed target, while, in closed networks they steer instantaneous throughputs toward a fixed target. In both cases, the proposed policies measure distance from the target using a weighted norm. In open networks, we establish that for an L2 norm the corresponding policies are stable. In closed networks, we establish that with proper target selection the corresponding policy is efficient, that is, attains bottleneck throughput in the infinite population limit. In both open and closed networks, the proposed policies are amenable to distributed implementation using local state information. We exploit the work in a previous paper to select appropriate parameter values and outline how optimal parameter values can be computed. We report numerical results indicating that we obtain near-optimal policies (when the optimal can be computed) and significantly outperform heuristic alternatives. This work has applications in a number of areas including optimizing the processing of information in sensor networks.**

Index Terms—**Fluid models, multiclass queueing networks, routing, scheduling, sensor networks.**

I. INTRODUCTION

WE CONSIDER the problems of *scheduling* and *routing* in open and closed Markovian *multiclass queueing networks (MQNETs)*. Such networks process jobs that belong to multiple *types*, differing in their arrival processes, routes through the network, processing times, and cost per unit of waiting time. *Scheduling* or *sequencing* decisions determine which job is being processed at each point in time in the various network nodes. *Routing decisions* determine the sequence of nodes at which a job undergoes processing as it traverses the network. In open networks, the objective is to minimize a weighted sum of mean waiting times, while, in closed networks, we seek to maximize a weighted sum of mean throughputs over the various job types.

Such problems have applications in a number of domains, including, manufacturing systems, multiprocessor computer systems, communication networks, and clusters of computing servers. One particular application area of interest concerns *sensor networks*. These networks consist of "sensing nodes" and "processing nodes." The term "processing nodes" is meant to describe all nodes that collect and process information gathered by the sensors. Such processing can range from simple storage to full processing of data in order to make decisions (e.g., as a controller in a networked control system). Often times information collected by sensors needs to undergo several stages of processing at several processing nodes. Sensor networks typically operate in adverse environments using battery-powered sensors with limited local processing capabilities. It follows that the "response time" of processing nodes needs to be highly optimized to avoid loosing information from sensor nodes that are nearing the end of their lifetime, or even avoid delayed action based on critical and time-sensitive information. We can model the collection of processing nodes as an MQNET. Control actions that affect performance include routing and scheduling. Routing includes both routing of messages from sensor nodes to processing nodes, as well as, routing of messages between processing nodes. Scheduling, also, can be done at both the processing node level, among jobs that wait to be processed, and within a processing node among jobs that wait to access the various node resources (e.g., CPU, disk, network interface card, etc.).

Performance analysis in MQNETs is notoriously hard. Naturally, optimizing an MQNET is even harder. A version of the scheduling problem we consider was shown to be EXP-complete [2]. Under Markovian assumptions the problem can be formulated as a stochastic *dynamic programming (DP)* problem. This is not very useful in practice for two reasons: 1) Bellman's "curse of dimensionality" prohibits us from computing the optimal policy in large instances, and 2) implementing the optimal policy is rather challenging, since typically, nodes need to have global state information.

There is, by now, a fair amount of work in optimizing MQNETs. A part of the literature has focused on heavy-traffic, Brownian, approximations to derive policies in special cases [3], [4]. [1] and [5] provide a polyhedral relaxation of the region of achievable performance and obtain bounds on optimal performance. This relaxation is shown to be exact in Klimov's model [6]. Stability is an important and more basic question than optimization. It should be noted that in open MQNETs the usual condition of node utilizations being less than one is not sufficient for the stability of all policies. [7] proves a seminal result establishing that the stability of a fluid model is a sufficient condition for the stability of the stochastic open

I. C. Paschalidis is with the Center for Information and Systems Engineering, and the Department of Manufacturing Engineering, Boston University, Boston, MA 02215 USA (e-mail: yannisp@bu.edu).

C. Su and M. C. Caramanis are with the Center for Information and Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: csu@bu.edu; mcaraman@bu.edu).

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302 Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number

1. REPORT DATE **OCT 2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**
4. TITLE AND SUBTITLE **Target-Pursuing Scheduling and Routing Policies for Multiclass Queueing Networks**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Boston University,Department of Manufacturing Engineering,Boston,MA,02215**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

We propose a parametric class of myopic scheduling and routing policies for open and closed multiclass queueing networks. In open networks, they steer the state of the system toward a predetermined and fixed target, while, in closed networks they steer instantaneous throughputs toward a fixed target. In both cases, the proposed policies measure distance from the target using a weighted norm. In open networks, we establish that for an L2 norm the corresponding policies are stable. In closed networks we establish that with proper target selection the corresponding policy is efficient, that is, attains bottleneck throughput in the infinite population limit. In both open and closed networks, the proposed policies are amenable to distributed implementation using local state information. We exploit the work in a previous paper to select appropriate parameter values and outline how optimal parameter values can be computed. We report numerical results indicating that we obtain near-optimal policies (when the optimal can be computed) and significantly outperform heuristic alternatives. This work has applications in a number of areas including optimizing the processing of information in sensor networks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a REPORT **unclassified**	b ABSTRACT **unclassified**	c THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

MQNET. Several scheduling policies have been proposed for open MQNETs, including, fluctuation smoothing policies [8], affine shifts of policies for the fluid model [9], tracking of heavy-traffic-based policies [10], and tracking optimal trajectories of the fluid model [11]. The latter policies perform well far from equilibrium but not necessarily equally well in steady state. Other approaches using diffusion models have been proposed in [12], [13].

The work in [7] has been extended in closed networks with a single job type [14]. Closed networks are always stable since the total number of jobs in the network is constant. The notion of *efficiency* of a scheduling policy has been introduced in [14] and can be seen as analogous to stability in open networks. Specifically, a scheduling policy is called efficient if it attains the maximum throughput of the bottleneck node as the population of the network grows large.

Our work is also related to problems in manufacturing systems (see [15]–[19]). In this literature, optimal stochastic scheduling policies have been shown to yield controlled dynamics that follow piecewise linear trajectories characterized by attractors of monotonically decreasing dimension leading to a point attractor (termed hedging point).

Inspired by the body of work we outlined, in this paper we introduce a class of scheduling and routing policies for MQNETs which we call *target-pursuing (TP)* policies. In both open and closed network they "steer" appropriate state variables toward a predetermined and fixed "target." In open networks, this is done for the vector of jobs present from all classes, while in closed networks for the instantaneous throughput rates of all job classes. These policies were especially motivated by the observations: 1) state feedback tracking policies in control are often effective, and 2) the polyhedral relaxations of the region of achievable performance in [1] are often tight, thus, a policy that seeks to maintain the state of the system in the neighborhood of optimal points in these polyhedra can be rather effective. Our main findings are as follows.

1) In open networks, we show that TP policies are *stable* and in closed networks we establish that TP policies are *efficient*. To that end, we work with a fluid model.

2) We demonstrate that TP policies are amenable to distributed implementation without the need to maintain global state information. This is key in making these policies attractive to implement.

3) We discuss ways of tuning policy parameters, notably the targets, in order to select the best policy within the class.

4) We provide a set of illustrative numerical results suggesting that TP policies perform close to optimal (when it can be computed) and outperform heuristic alternatives.

The remainder of this paper is organized as follows. Section II presents our model of open MQNETs where only scheduling is subject to optimization. Section III introduces TP policies for open networks. Section IV discusses implementation issues. Section V establishes stability. Section VI outlines how to tune policy parameters. Section VII considers open MQNETs where routing is also subject to optimization. Section VIII focuses on closed MQNETs. Section IX contains our numerical results. Concluding remarks are in Section X.

Notational Conventions: Throughout this paper all vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write $\mathbf{x} = (x_1, \ldots, x_R)$ for the column vector $\mathbf{x}$. Matrices are denoted by boldface upper case letters and prime denotes transpose. We use $\mathbf{e}$ to denote the vector of all ones, $\mathbf{0}$ for the vector of all zeroes, $\mathbf{e}_i$ for the ith unit vector, and $\mathbf{I}$ for the identity matrix. For any event A, $\overline{A}$ denotes its complement and $\mathbf{1}\{A\}$ its indicator function. For any $\mathbf{x} \in \mathbb{R}^R$, we denote $|x| = \sum_{i=1}^{R} x_i$. We also use the weighted L2 norm

$$\|\mathbf{x}\|_\beta^2 = \sum_{i=1}^{R} \beta_i (x_i)^2. \tag{1}$$

When we write $\|\mathbf{x}\|^2$ it is assumed that $\beta = \mathbf{e}$.

II. MODEL AND KEY QUANTITIES

In this section, we present the model of our open MQNET. Initially, we consider only *sequencing decisions*.

Consider a network consisting of N single-server nodes. Jobs entering the network are being processed at a series of nodes before, eventually, leaving the system. Externally arriving jobs can be of multiple *types* differing in their arrival processes, routes through the network, processing requirements, and costs per unit of waiting time. To account for jobs processed at different nodes we define the *class* of a job as the pair of job type and node at which it is receiving service. For example, for a network with K job types there can be up to $K \times N$ classes. Let R be the total number of classes.

We let $\sigma(r)$ denote the node at which class r is served and $C_i \triangleq \{r | \sigma(r) = i\}$ the *constituency* list of node i, that is, the set of classes served at node i. Routing is *probabilistic*, namely, when a class r job finishes processing at node $\sigma(r)$ it is routed to node $\sigma(r')$ and becomes a job of class r' with probability $p_{rr'}$, or leaves the network with probability $p_{r0} = 1 - \sum_{r'=1}^{R} p_{rr'}$. Notice that we adopt the notational convention of identifying the external (to the network) world as class zero. We denote by $\mathbf{P} = \{p_{rr'}\}_{r,r'=1}^{R}$ the routing matrix, which, since the network is open, is assumed to be substochastic, or equivalently the matrix $(\mathbf{I} - \mathbf{P}')$ is invertible. External arrivals come according to R independent Poisson arrival processes, one for each class, with rate λ_{0r} for class r. Finally, service times are independent of anything else in the network and exponentially distributed with parameter μ_r for class r.

Let $\mathbf{n}(t) = (n_1(t), \ldots, n_R(t))$ denote the vector of the number of jobs present in the network from each class at time t. Under the Markovian assumptions we have imposed, and under a Markovian policy (i.e., a policy whose actions at time t depend on $\mathbf{n}(t)$ only), the network evolves according to a continuous-time Markov chain with state $\mathbf{n}(t)$. Letting λ_r denote the total (external and internal) mean arrival rate of class r jobs, the following *traffic equations* are satisfied:

$$\lambda_r = \lambda_{0r} + \sum_{r'=1}^{R} p_{r'r}\lambda_{r'}, \qquad r = 1, \ldots, R. \tag{2}$$

In matrix notation this system of equations can be written as $\lambda = \lambda_0 + \mathbf{P}'\lambda$, where $\lambda = (\lambda_1, \ldots, \lambda_R)$ and $\lambda_0 = (\lambda_{01}, \ldots, \lambda_{0R})$.

Since the network is open, (2) has a unique solution given by $\boldsymbol{\lambda} = (\mathbf{I} - \mathbf{P}')^{-1}\boldsymbol{\lambda}_0$. Let $\rho_r \overset{\triangle}{=} \lambda_r/\mu_r$ the fraction of time server $\sigma(r)$ works on class r. The utilization of server i is $\rho_i = \sum_{r \in C_i} \rho_r$. We assume $\rho_i < 1$ for all nodes i; otherwise, the network is unstable in the sense that $|\mathbf{n}(t)| \to \infty$ w.p. 1 (with probability one) as $t \to \infty$.

We are interested in a scheduling policy minimizing

$$\lim_{t \to \infty} \sum_{r=1}^{R} h_r \mathbf{E}\left[n_r(t)\right] \tag{3}$$

where $\mathbf{h} = (h_1, \ldots, h_R)$ are given weights. Equivalently, we seek to minimize a weighted sum of the mean queue lengths where the expectation is taken with respect to the steady-state distribution. Using Little's law this cost function can be easily transformed into a weighted sum of the mean waiting times.

III. TARGET-PURSUING POLICIES

Next, we introduce the family of policies of interest. As mentioned earlier, [1] and [5] provided a characterization of the *achievable region* for the performance vector $\lim_{t \to \infty} \mathbf{E}[\mathbf{n}(t)]$ under all Markovian, preemptive, and stable policies. By allowing randomized (non-Markovian) policies, the resulting achievable region can be seen to be convex. We denote by $\mathcal{A}$ this convex achievable region; every point in $\mathcal{A}$ is achievable by randomizing among Markovian policies.[1] (We note that relaxing achievable state variables, but in a fluid model setting, has also been exploited in [13].)

More specifically, [1] derives a polyhedron, say $\mathcal{P}$, that contains the achievable region for $\lim_{t \to \infty} \mathbf{E}[\mathbf{n}(t)]$ (see Fig. 1). Optimizing $\lim_{t \to \infty} \mathbf{h}'\mathbf{E}[\mathbf{n}(t)]$ over $\mathcal{P}$ yields an optimal solution, say $\mathbf{w}^*$, whose cost is a lower bound on the optimal performance. Although the polyhedron $\mathcal{P}$ has an exponential number of constraints in R, optimizing $\lim_{t \to \infty} \mathbf{h}'\mathbf{E}[\mathbf{n}(t)]$ can be done in polynomial time by solving a *linear programming (LP)* problem in an associated higher dimensional polyhedron with a polynomial in R number of variables and constraints [1]. The bound is often quite tight, but in general $\mathbf{w}^* \notin \mathcal{A}$ and cannot be achieved by any policy. An interesting question is whether $\mathbf{w}^*$ "contains information" leading to a "good" policy.

Motivated by the fact that $\mathbf{w}^*$ can be computed efficiently (in polynomial time) and that it is often "close" to the optimal $\mathbf{z}^* = \arg\min_{\mathbf{z} \in \mathcal{A}} \mathbf{h}'\mathbf{z}$, we consider a myopic state feedback policy that aims at "steering" the state of the system toward $\mathbf{w}^*$. Such a policy belongs to the following class of policies.

Definition 1: We define as TP the class of scheduling policies which at each time t, and for a finite time interval Δt, minimize

$$\mathbf{E}\left[\|\mathbf{n}(t + \Delta t) - \boldsymbol{\theta}\| \,|\, \mathbf{n}(t)\right]$$

for some norm $\|\cdot\|$.

The review interval Δt can be selected to be smaller than the timescale of arrivals and services. The expectation in the previous definition is with respect to the probability distribution of $\mathbf{n}(t + \Delta t)$ conditional on the state being $\mathbf{n}(t)$ at time t. Furthermore, the minimization is over all scheduling decisions made
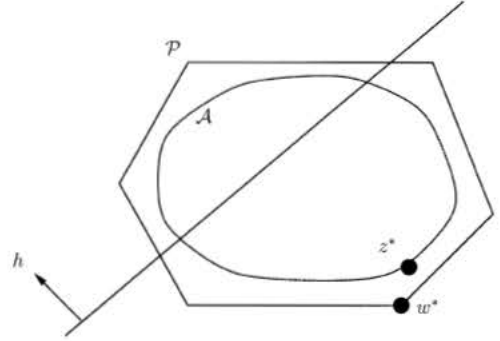
Fig. 1. Achievable region $\mathcal{A}$ included in a polyhedron $\mathcal{P}$.

at time t, i.e., we seek to select which job class is processed at every node at time t. Note that the selection of the norm and of $\boldsymbol{\theta}$ are left open. As defined, TP policies are not necessarily work-conserving (i.e., servers can idle even if there is work to be done). Henceforth, we refer to their work-conserving versions as *work-conserving* TP policies.

In the sequel, we will consider the norm of (1) and explore several ways of selecting an appropriate "target" $\boldsymbol{\theta}$. As we have indicated above, one potential target is $\mathbf{w}^*$. We will see that setting $\boldsymbol{\theta} = \mathbf{w}^*$ often leads to a good policy. We note that TP policies are myopic and greedy, thus, we do not claim any optimality properties. We only establish stability results and provide some analytical and numerical evidence that they perform quite well.

IV. IMPLEMENTATION ISSUES

In this section, we discuss how to best implement TP policies. We will see that the implementation complexity amounts to solving an LP problem at each decision epoch. However, the computations can be decomposed across nodes and nodes require only (limited) local state information to perform them.

Consider the network of Section II and let us uniformize the corresponding continuous-time Markov chain. In particular, set $\nu = \sum_{r=1}^{R} \lambda_{0r} + \sum_{r=1}^{R} \mu_r$ and consider the uniformized chain with uniform transition rate ν. Let $\{\tau_k\}$ be the sequence of epochs at which the uniformized chain makes transitions; this is also the sequence of ticks from a "Poisson clock" with rate ν. As $\mathbf{n}(t)$ is right-continuous, $\mathbf{n}(\tau_k)$ refers to the state right after the kth transition. Select $\Delta t \ll \min_r \min\{1, 1/\lambda_{0r}, 1/\mu_r\}$, i.e., Δt is small enough and in a much faster timescale than arrivals and services. In $(\tau_k, \tau_k + \Delta t]$ we have: an external class r arrival with probability $\lambda_{0r}\Delta t$, a class r service completion with probability $\mu_r \Delta t$ if node $\sigma(r)$ is working on class r at τ_k, a self-transition with probability $\mu_r \Delta t$ if node $\sigma(r)$ is not working on class r at τ_k, no transition with probability $1 - \nu\Delta t$, and more than one transitions with probability $o(\Delta t)$.

In the uniformized Markov chain, scheduling decisions need only to be made at the epochs τ_k. Let $B_r(\tau_k)$ the event that node $\sigma(r)$ is working on class r right after τ_k, and $\overline{B_r(\tau_k)}$ its complement. For any $\boldsymbol{\theta} \in \mathbb{R}$, Δt as defined before, and using norm (1), the TP policy minimizes

$$\frac{1}{\Delta t}\mathbf{E}\left[\|\mathbf{n}(\tau_k + \Delta t) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 - \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 \,|\, \mathbf{n}(\tau_k)\right]$$
$$= -\nu\|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 + \sum_{r=1}^{R} \lambda_{0r}\|\mathbf{n}(\tau_k) + \mathbf{e}_r - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2$$

---

[1]Here randomization among two policies A and B means "time-sharing," i.e., take a large time interval and follow A for a fraction of that interval and B for the remaining fraction. The resulting policy is non-Markovian.

$$+ \sum_{r=1}^{R} \mu_r \mathbf{1}\{B_r(\tau_k)\} \left[\sum_{r'=1}^{R} p_{rr'} \|\mathbf{n}(\tau_k) - \mathbf{e}_r + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 \right.$$
$$\left. + p_{r0} \|\mathbf{n}(\tau_k) - \mathbf{e}_r - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 \right]$$
$$+ \sum_{r=1}^{R} \mu_r \mathbf{1}\{\overline{B_r(\tau_k)}\} \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 + \frac{o(\Delta t)}{\Delta t}. \qquad (4)$$

Set $\mathbf{x}(\tau_k) = (\mathbf{1}\{B_1(\tau_k)\}, \ldots, \mathbf{1}\{B_R(\tau_k)\})$. Discarding constants, notice that the right-hand side (RHS) of (4) can be written as $\mathbf{x}(\tau_k)' \mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) + o(\Delta t)/\Delta t$ where $\mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^R$ is appropriately defined. Since for small enough Δt the first term dominates, to implement the TP policy with norm $\|\cdot\|_{\boldsymbol{\beta}}^2$ we will be solving at each epoch τ_k the following LP problem:

$$(\mathbf{LP1}) \quad \min \quad \mathbf{x}(\tau_k)' \mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$$
$$\text{s.t.} \quad \sum_{r \in C_i} x_r(\tau_k) \le 1, \qquad i = 1, \ldots, N$$
$$0 \le \mathbf{x}(\tau_k) \le \mathbf{n}(\tau_k) \qquad (5)$$

where $\mathbf{x}(\tau_k)$ is the decision vector.[2] The first inequality constraint above bounds the utilization of each server by one and the constraint $\mathbf{x}(\tau_k) \le \mathbf{n}(\tau_k)$ ensures that no capacity is allocated to empty classes. If we impose work conservation, the first inequality constraint becomes an equality at all nodes i with jobs present. Under both work-conserving and nonwork-conserving TP policies, the constraint matrix is totally unimodular, hence, the feasible set is a polytope with integer extreme points and $(\mathbf{LP1})$ yields an integer optimal solution.

A couple of remarks on the implementation complexity are in order. The size of $(\mathbf{LP1})$ is $O(RN)$ which is polynomial in the size of the MQNET. Very large instances of LP problems can be solved efficiently (in polynomial time) using interior-point algorithms. For large networks, though, the computational requirements for solving $(\mathbf{LP1})$ can be substantial. Moreover, the formulation in (5) requires a centralized computation with global state information. Fortunately, the work can be decentralized and distributed across various nodes. To see that, and for simplicity of the exposition, let $\boldsymbol{\beta} = \mathbf{e}$. Decomposing $(\mathbf{LP1})$ across nodes, node i has to solve

$$(\mathbf{Node - LP1})$$
$$\min \quad \sum_{r \in C_i} \mu_r x_r(\tau_k)(2 - p_{r0})$$
$$- 2 \sum_{r \in C_i} \mu_r x_r(\tau_k) p_{r0} (n_r(\tau_k) - \theta_r)$$
$$+ 2 \sum_{r \in C_i} \mu_r x_r(\tau_k) \sum_{r'=1}^{R} p_{rr'}$$
$$\times [n_{r'}(\tau_k) - \theta_{r'} - n_r(\tau_k) + \theta_r]$$
$$\text{s.t.} \quad \sum_{r \in C_i} x_r(\tau_k) \le 1$$
$$0 \le x_r(\tau_k) \le n_r(\tau_k) \qquad r \in C_i$$

where $x_r(\tau_k), r \in C_i$, are the decision variables. Typically the number of classes served at an arbitrary node i is much less than R. Moreover, to solve $(\mathbf{Node - LP1})$, node i needs state

---

[2]Equivalently, and to avoid inaccuracies due to Δt, we can define the TP policy as the policy obtained through $(\mathbf{LP1})$.

information for classes $r \in C_i$, and all classes within one hop, i.e., r' with $p_{rr'} > 0$ for all $r \in C_i$. The number of such r' would also be much less than R in most cases. Thus, $(\mathbf{Node - LP1})$ can be solved by each node using local information much faster than solving $(\mathbf{LP1})$.

V. STABILITY ANALYSIS

In this section, we show that TP policies are *stable*. To that end, and following [7] and [20], we consider a fluid model, establish stability of the fluid model, and then infer the stability of the stochastic system.

A. Fluid Model

To avoid overburdening our notation we use $\mathbf{n}(t)$ to denote the queue length vector in the fluid model as well; it will be evident from the context whether we refer to the fluid model or the stochastic system. Let $T_r(t)$ the cumulative amount of time server $\sigma(r)$ has spent working on class r in $[0, t]$. Let $\mathbf{M} = \text{diag}(\mu_1, \mu_2, \ldots, \mu_R)$, and $\mathbf{u}(t) = (u_1(t), \ldots, u_R(t))$, where $\text{diag}(x_1, \ldots, x_R)$ denotes the diagonal matrix with main diagonal $x_1, \ldots, x_R$ and zeroes elsewhere. Let also $\mathbf{C} = (c_{ir})$ be the constituency matrix of the network with $c_{ir} = \mathbf{1}\{\sigma(r) = i\}$, for all $r = 1, \ldots, R$ and $i = 1, \ldots, N$. In the fluid model, for all $t \ge 0$ we have

$$\dot{\mathbf{n}}(t) = \boldsymbol{\lambda}_0 - (\mathbf{I} - \mathbf{P}')\mathbf{M}\,\mathbf{u}(t)$$
$$\mathbf{C}\,\mathbf{u}(t) \le \mathbf{e}$$
$$\mathbf{n}(t), \mathbf{u}(t) \ge \mathbf{0}. \qquad (6)$$

Here, $u_r(t) = \dot{T}_r(t)$ which can be interpreted as the fraction of the capacity of server $\sigma(r)$ allocated to class r at time t. The functions $n_r(t)$ and $T_r(t)$ are absolutely continuous, and thus, differentiable almost everywhere. The equations in (6) hold for all times t at which $n_r(t)$ and $T_r(t)$ are differentiable; these points in time will be referred to as *regular*. We next derive the fluid version of the TP policy under the norm in (1).

Note that at every decision epoch TP policies minimize the expression given in Definition 1 subject to the constraints of $(\mathbf{LP1})$ [cf. (5)]. Let us first consider the objective function in this minimization. Let $|\mathbf{n}(0)| = k > 0$ and consider the fluid scaling of the stochastic system

$$\bar{\mathbf{n}}^k(t) = \frac{1}{k}\mathbf{n}^k(kt)$$

where $\mathbf{n}^k(\cdot)$ denotes the queue length vector in the stochastic system initialized with $|\mathbf{n}(0)| = k$. Since we will deal with the limit $k \to \infty$ in the space of sample paths of $\mathbf{n}^k(t)$, let us explicitly write $\mathbf{n}^k(t, \omega)$ for a particular sample path ω of $\mathbf{n}^k(t)$. We restrict ourselves only to ω satisfying the *strong law of large numbers (SLLN)* for the arrival, service, and routing processes. [20] proves that if $|\mathbf{n}^k(0, \omega)|/k$ is bounded as $k \to \infty$, then $\bar{\mathbf{n}}^k(\cdot, \omega)$ is precompact as $k \to \infty$ in the Skorohod path space $\mathbb{D}^R[0, \infty)$ endowed with the u.o.c. (uniformly on compact sets) topology. This implies that $\bar{\mathbf{n}}^k(\cdot, \omega)$ is tight as $k \to \infty$ ([21]–[23]). Thus, for each sequence $k \to \infty$ there exists a subsequence $k_s \to \infty$ along which

$$\bar{\mathbf{n}}^{k_s}(\cdot, \omega) \to \bar{\mathbf{n}}(\cdot), \qquad \text{u.o.c.}$$

for some process $\bar{\mathbf{n}} \in \mathbb{D}^R[0, \infty)$ which is called *fluid limit* and satisfies (6). At time kt the TP policy minimizes

$$\mathbf{E}\left[\|\mathbf{n}(kt + \Delta t) - \boldsymbol{\theta}\| - \|\mathbf{n}(kt) - \boldsymbol{\theta}\| \,|\, \mathbf{n}(kt)\right].$$

Scaling by $1/k$, this is equivalent to minimizing

$$\mathbf{E}\left[\left[\left\|\frac{\mathbf{n}\left(k\left(t + \frac{\Delta t}{k}\right)\right)}{k} - \frac{\boldsymbol{\theta}}{k}\right\| - \left\|\frac{\mathbf{n}(kt)}{k} - \frac{\boldsymbol{\theta}}{k}\right\|\right] \,\Big|\, \mathbf{n}(kt)\right].$$

Taking $k \to \infty$, and since the stochastic system converges to the fluid limit for all ω considered before, we conclude that for all $\boldsymbol{\theta}$ the fluid version of the TP policy seeks to minimize

$$\frac{d}{dt}\|\bar{\mathbf{n}}(t)\| \tag{7}$$

at regular t. Using the norm in (1) and taking the fluid limit of the policy prescribed by (**LP1**) the corresponding fluid version minimizes $d\|\bar{\mathbf{n}}(t)\|_{\boldsymbol{\beta}}^2/dt$.

Consider next the constraints of (**LP1**) under which the minimization of (7) is done. The constraints of (**LP1**) under the fluid scaling translate to the constraints in (6) with the additional constraint that for all classes r it holds $u_r(t) = 0$ whenever $n_r(t) = 0$. Therefore, the fluid version of the TP policy minimizes the expression in (7) subject to the fluid feasibility constraints of (6) and the additional "idle when empty" constraint indicated above. This policy is well defined for all regular t and we will refer to it as the *fluid target-pursuing (FTP)* policy. FTP aims at maximizing the negative drift and driving the fluid level toward zero. It can be shown that the amounts of time allocated to various classes in the stochastic system under the TP policy converge to corresponding quantities in the fluid model under the FTP.

B. Stability of the Fluid Model

We next establish the stability of the fluid model operating under the FTP policy, that is, the nonwork-conserving policy minimizing $d\|\mathbf{n}(t)\|_{\boldsymbol{\beta}}^2/dt$ for each t.

Proposition V.1: Consider the fluid model operating under the nonwork-conserving FTP policy which uses the weighted L2 norm $\|\mathbf{n}(t)\|_{\boldsymbol{\beta}}^2$, where $\boldsymbol{\beta} > 0$. For every solution of the fluid equations (6) satisfying $|\mathbf{n}(0)| \leq 1$ and $u_r(t) = 0$ whenever $n_r(t) = 0$ for all r, there exists some $\delta(\eta) > 0$ such that for all $0 < \eta < 1$ and all $t \geq \delta$ it follows $|\mathbf{n}(t)| \leq \eta$.

Proof: Fix $\eta \in (0, 1)$. Let $\mathbf{B} = \mathrm{diag}(\beta_1, \ldots, \beta_R)$ and

$$G(t) \triangleq \|\mathbf{n}(t)\|_{\boldsymbol{\beta}}^2 = \mathbf{n}'(t)\mathbf{Bn}(t).$$

Clearly, $G(t) = 0$ if and only if $\mathbf{n}(t) = 0$, and $G(\mathbf{n}(t))$ can be shown to be locally Lipschitz continuous in $\mathbf{n}(t)$, i.e., for any compact set $\mathcal{O}$, there exists a constant $\kappa(\mathcal{O})$ such that for any $\mathbf{n}_1(t), \mathbf{n}_2(t) \in \mathcal{O}$ it holds that $|G(\mathbf{n}_1(t)) - G(\mathbf{n}_2(t))| \leq \kappa(\mathcal{O})|\mathbf{n}_1(t) - \mathbf{n}_2(t)|$. We show next that $G(t)$ is nonincreasing in t. Using the fluid model dynamics of (6) we obtain

$$\dot{G}(t) = 2\mathbf{n}'(t)\mathbf{B}[\boldsymbol{\lambda}_0 + \mathbf{P}'\mathbf{Mu}(t) - \mathbf{Mu}(t)]. \tag{8}$$

Let us now employ the nonwork-conserving policy that assigns a fraction $u_r(t) = \rho_r$ of node's $\sigma(r)$ capacity to all nonempty classes r and zero capacity to all empty classes. Let $\mathbf{u}(t)$ be

the vector induced by this policy. Let also $\tilde{\mathbf{u}} = (\rho_1, \ldots, \rho_R)$. Note that for all t $\mathbf{P}'\mathbf{Mu}(t) \leq \mathbf{P}'\mathbf{M}\tilde{\mathbf{u}}$, and $\mathbf{n}'(t)\mathbf{BMu}(t) = \mathbf{n}'(t)\mathbf{BM}\tilde{\mathbf{u}}$. As a result, (8) implies

$$\dot{G}(t) \leq 2\mathbf{n}'(t)\mathbf{B}[\boldsymbol{\lambda}_0 - (\mathbf{I} - \mathbf{P}')\mathbf{M}\tilde{\mathbf{u}}] = 0 \tag{9}$$

where in the last equation we used the traffic equations (cf. (2)). Since the FTP policy described in the statement of the Proposition minimizes $\dot{G}(t)$ for all t, we will have $\dot{G}(t) \leq 0$ for this latter policy. Thus, $G(t)$ is nonincreasing in time and $G(t) \leq G(0)$ for all $t \geq 0$.

Let $\xi_r(t) = \beta_r n_r(t) - \sum_{r'=1}^R p_{rr'}\beta_{r'}n_{r'}(t)$, $\boldsymbol{\xi}(t) = (\xi_1(t), \ldots, \xi_R(t))$, and note that $\boldsymbol{\xi}(t) = (\mathbf{I} - \mathbf{P})\mathbf{Bn}(t)$. Due to the monotonicity of $G(t)$, for all $t \geq 0$ and r we have

$$\beta_{\min}(n_r(t))^2 \leq \beta_r(n_r(t))^2 \leq \sum_{r=1}^R \beta_r(n_r(t))^2 \leq G(0)$$

where $\beta_{\min} = \min_r \beta_r$. Thus, $n_r(t) \leq \sqrt{G(0)/\beta_{\min}}$, for all $t \geq 0$ and r. This implies that for all $t \geq 0$ and r

$$|\xi_r(t)| \leq \beta_r n_r(t) + \sum_{r'=1}^R p_{rr'}\beta_{r'}n_{r'}(t)$$

$$\leq \beta_{\max}\sqrt{\left(\frac{G(0)}{\beta_{\min}}\right)}(2 - p_{r0}) \tag{10}$$

where $\beta_{\max} = \max_r \beta_r$. Consider the nonwork-conserving policy which allocates to class r

$$u_r(t) = \begin{cases} \frac{\lambda_r + \epsilon\xi_r(t)}{\mu_r}, & \text{if } n_r(t) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Let $\epsilon > 0$ be such that at all $t \geq 0$ we have $\rho_j < 1$ for all nodes j and $\lambda_r + \epsilon\xi_r(t) \geq 0$ for all r. Such an $\epsilon > 0$ exists due to (10). Using this policy, from (8), we obtain

$$\dot{G}(t) \leq 2\sum_{r|n_r(t)>0}\beta_r n_r(t)$$

$$\times \left[\lambda_{0r} - \lambda_r - \epsilon\xi_r(t) + \sum_{r'}p_{r'r}(\lambda_{r'} + \epsilon\xi_{r'}(t))\right]$$

$$= 2\epsilon\sum_r \beta_r n_r(t)\left[\sum_{r'}p_{r'r}\xi_{r'}(t) - \xi_r(t)\right]$$

$$= -2\epsilon\mathbf{n}'(t)\mathbf{B}(\mathbf{I} - \mathbf{P}')(\mathbf{I} - \mathbf{P})\mathbf{Bn}(t).$$

In the first inequality shown above, we used the fact $\lambda_r + \epsilon\xi_r(t) \geq 0$ for all r and $t \geq 0$. In the first equality above we used the traffic equations in (2). Let $\mathbf{D} = \mathbf{B}(\mathbf{I} - \mathbf{P}')(\mathbf{I} - \mathbf{P})\mathbf{B}$ and note that $\mathbf{D}$ is symmetric. Since we are dealing with an open queueing network, $(\mathbf{I} - \mathbf{P}')$ is invertible, thus, $\mathbf{B}(\mathbf{I} - \mathbf{P}')$ is also invertible. Hence, $\mathbf{D}$ is positive definite and has real and strictly positive eigenvalues. Letting $s_{\min}$ be the smallest eigenvalue of $\mathbf{D}$ we obtain

$$\dot{G}(t) \leq -2\epsilon\mathbf{n}'(t)\mathbf{Dn}(t) \leq -2s_{\min}\epsilon\|\mathbf{n}(t)\|^2. \tag{12}$$

Whenever $|\mathbf{n}(t)| \geq \eta$, we have

$$G(t) \geq \beta_{\min}\sum_i (n_i(t))^2 \geq \beta_{\min}\frac{\left(\sum_i n_i(t)\right)^2}{R} \geq \beta_{\min}\frac{\eta^2}{R}.$$

This implies

$$\beta_{\max} \|\mathbf{n}(t)\|^2 \geq \mathbf{n}'(t)\mathbf{B}\mathbf{n}(t) \geq \beta_{\min} \frac{\eta^2}{R}$$

which implies in turn

$$\|\mathbf{n}(t)\|^2 \geq \frac{\beta_{\min}}{\beta_{\max}} \frac{\eta^2}{R} \triangleq \tilde{\phi}^*. \tag{13}$$

Consequently, in the fluid model under policy (11), whenever $G(t) \geq \beta_{\min}\eta^2/R$, (12) and (13) yield

$$\dot{G}(t) \leq -2s_{\min}\epsilon\tilde{\phi}^*. \tag{14}$$

Since the FTP policy described in the statement of the Proposition minimizes $\dot{G}(t)$ for all t, $\dot{G}(t)$ will be upper-bounded by $-2s_{\min}\epsilon\tilde{\phi}^*$ under the latter policy as well.

Suppose now $G(0) \geq \beta_{\min}\eta^2/R$. Equation (14) implies that $G(t)$ will reach the region $G(t) \leq \beta_{\min}\eta^2/R$ within time t_η where

$$t_\eta \leq \frac{G(0) - \beta_{\min}\frac{\eta^2}{R}}{2s_{\min}\epsilon\tilde{\phi}^*}.$$

Furthermore, $G(t)$ will remain in this region for all $t \geq t_\eta$ since it is a nonincreasing function of time. We conclude that for all $t \geq t_\eta$, it holds that $|\mathbf{n}(t)| \leq \eta$, since otherwise $G(t) \geq \beta_{\min}\eta^2/R$. Finally, in the case $G(0) \leq \beta_{\min}\eta^2/R$, the same argument applies and $|\mathbf{n}(t)| \leq \eta$ for all $t \geq 0$. ∎

C. Stability of the Stochastic Network

We conclude this section by establishing that the MQNET is stable under the TP policy using the norm in (1). Note that for any target $\boldsymbol{\theta} \in \mathbb{R}^R$ the TP policy is Markovian and under this policy the state of the network is the queue length vector $\mathbf{n}(t) \in \mathbb{Z}_+^R$ that evolves as a continuous-time Markov chain. The next theorem establishes that this Markov chain is *positive Harris recurrent* (see [7], [20], [21]).

Theorem V.2: Consider the MQNET of Section II operated under the TP policy that uses the norm $\|\mathbf{n}(t)\|_{\boldsymbol{\beta}}$, where $\boldsymbol{\beta} > 0$. The Markov chain $\mathbf{n}(t)$ is positive Harris recurrent.

Proof: We will slightly modify the proof in [7]. Proposition V.1 establishes that there exists some $\delta \geq 1$ such that for any solution $\tilde{\mathbf{n}}(t)$ of the fluid model equations and any $0 < \eta < 1$ we have $|\tilde{\mathbf{n}}(t)| \leq \eta$ for all $t \geq \delta$. Let $\{\mathbf{z}_k\}$ be any sequence of initial states $\mathbf{n}(0)$ with $|\mathbf{z}_k| \to \infty$ as $k \to \infty$. From the existence of the fluid limit (see [7]) there exists a subsequence $\{\mathbf{z}_{k_j}\}$ such that

$$\lim_{j\to\infty} \frac{1}{|\mathbf{z}_{k_j}|}\mathbf{n}^{\mathbf{z}_{k_j}}\left(|\mathbf{z}_{k_j}|\delta\right) = \bar{\mathbf{n}}(\delta)$$

where, as in Section V-A, $\bar{\mathbf{n}}(\cdot)$ denotes the fluid limit. The fluid limit satisfies the fluid model equations, thus,

$$\lim_{j\to\infty} \frac{1}{|\mathbf{z}_{k_j}|}\left|\mathbf{n}^{\mathbf{z}_{k_j}}\left(|\mathbf{z}_{k_j}|\delta\right)\right| \leq \eta.$$

Using the uniform integrability (see [7, Lemma 4.5]) of the sequence on the left-hand side, we obtain

$$\lim_{j\to\infty} \frac{1}{|\mathbf{z}_{k_j}|}\mathbf{E}\left[\left|\mathbf{n}^{\mathbf{z}_{k_j}}\left(|\mathbf{z}_{k_j}|\delta\right)\right|\right] \leq \eta.$$

Since $\{\mathbf{z}_k\}$ is an arbitrary sequence we have

$$\lim_{|\mathbf{z}|\to\infty} \frac{1}{|\mathbf{z}|}\mathbf{E}\left[\left|\mathbf{n}^{\mathbf{z}}\left(|\mathbf{z}|\delta\right)\right|\right] \leq \eta.$$

Let $0 < \epsilon < 1 - \eta$. There exists some $\kappa \geq 1$ such that

$$\frac{1}{|\mathbf{z}|}\mathbf{E}\left[\left|\mathbf{n}^{\mathbf{z}}\left(|\mathbf{z}|\delta\right)\right|\right] \leq 1 - \epsilon$$

for all $\mathbf{z}$ with $|\mathbf{z}| > \kappa$. The remainder of the proof follows exactly the proof of [7, Th. 3.1]. ∎

VI. OPTIMIZING OVER POLICY PARAMETERS

(**LP1**) suggests that if class i and j jobs are processed at the same node, $q_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) = q_j(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ constitutes a policy switching hyperplane. Namely, TP policies are characterized by switching hyperplanes determined by policy parameters. In this section, we discuss how we can optimize over these parameters, that is, the target $\boldsymbol{\theta}$ and the weight vector $\boldsymbol{\beta}$, in order to obtain the best policy within the class [i.e., minimizing (3)]. As mentioned in Section III, the achievable region LP provides a tentative value of $\boldsymbol{\theta}$ equal to $\mathbf{w}^*$ which, as we will see, performs quite well. Here, we are interested in further improving the selection of $\boldsymbol{\theta}$ and optimizing over $\boldsymbol{\beta}$ as well. To that end, we use a simulation-based method developed in [24]. The underlying idea is rather simple: During the course of a simulation of the system we obtain "gradient information" which we then use to optimize over the parameters.

1) Smooth Target-Pursuing Policies: To fix notation, consider the uniformized Markov chain of Section IV and the TP policy outlined there with weight vector $\boldsymbol{\beta} > 0$. At each transition epoch τ_k, scheduling decisions are made according to the optimal solution, say $\mathbf{x}^*(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$, of (**LP1**). Note that $\mathbf{x}^*(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ is piecewise constant in $(\boldsymbol{\theta}, \boldsymbol{\beta})$ with the jumps occurring at the points that the optimal solution switches from one extreme point of the feasible set to another. Consequently, using a simulation-based gradient optimization method to optimize over the parameters would not be very successful since the gradients would be zero most of the time.

To bypass this difficulty we use randomization to introduce a smoother version of our TP policies. For simplicity of the exposition, we concentrate on work-conserving TP policies; the nonwork-conserving case can be handled similarly. Let $\mathbf{y}^{(r)}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ be a feasible solution of (**LP1**) such that at time τ_k class r is served at node $\sigma(r)$ and the remaining decisions at all other nodes coincide with $\mathbf{x}^*(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$. Let $\gamma > 0$ be a scalar and set

$$\hat{\alpha}_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$$
$$= \frac{e^{-\gamma\mathbf{y}^{(r)}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})'\mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})}}{\displaystyle\sum_{r'\in C_{\sigma(r)}, \mathbf{n}_{r'}(\tau_k)>0} e^{-\gamma\mathbf{y}^{(r')}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})'\mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})}}. \tag{15}$$

At time τ_k, we serve class r at node $\sigma(r)$ with probability

$$\alpha_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) = \begin{cases} \hat{\alpha}_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}), & \text{if } \mathbf{n}_r(\tau_k) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Notice that as $\gamma \to 0$ all nonempty classes at a node have equal probability of being served, and as $\gamma \to \infty$ the randomized

policy converges to the policy implied by $\mathbf{x}^*(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$. The expression in (15) can be simplified as

$$\hat{\alpha}_r\left(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}\right) = \frac{e^{-\gamma q_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})}}{\displaystyle\sum_{r' \in C_{\sigma(r)}, \mathbf{n}_{r'}(\tau_k) > 0} e^{-\gamma q_{r'}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})}}$$

where $q_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ is the rth coordinate of $\mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$. Henceforth, we will be referring to this policy as the work-conserving *smooth target-pursuing (STP)* policy. This randomization scheme is useful in satisfying the conditions required by the simulation-based optimization algorithm we implement.

2) Simulation-Based Optimization: We adopted the STP policy and used the simulation-based method of [24] to optimize the objective of (3) over the parameters $(\boldsymbol{\theta}, \boldsymbol{\beta})$. In Section IX, we report illustrative numerical results and compare with a set of other scheduling policies. Under a set of stability and regularity conditions and a standard diminishing step-size rule, the algorithm in [24] is shown to convergence w.p. 1 to a local minimum. In our setting, the required stability condition is satisfied due to the result in Section V. The remaining regularity conditions, however, are not satisfied in all cases of interest.

Under the STP policy, $\alpha_r(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ satisfies the required regularity conditions with respect to $\boldsymbol{\theta}$ but not with respect to $\boldsymbol{\beta}$. As a result, we fixed $\boldsymbol{\beta}$ and used the method in [24] to optimize over $\boldsymbol{\theta}$. We then employed random search around $\boldsymbol{\beta} = \mathbf{e}$ to select a good $\boldsymbol{\beta}$. Admittedly, using a simulation-based method to optimize over $\boldsymbol{\theta}$ can be slow. We were encouraged to notice that initializing the algorithm with a tentative value equal to $\mathbf{w}^*$, obtained from the achievable region LP, led to considerably faster convergence.

VII. COMBINED ROUTING/SCHEDULING DECISIONS

In this section, we extend the basic queueing network model of Section II to consider the case where routing is not fixed but also subject to optimization.

We adopt the model and notation of Section II, indicating only the differences with the extended model we consider here. As in Section II, jobs of class $r = 1, \ldots, R$ arrive to the network according to a Poisson arrival process with rate λ_{0r}. Upon arrival, though, and before joining the corresponding queue, a router selects a particular class and routes the arriving job to that class. Let $A_{rr'}(t)$ denote the event that an externally arriving job of class r is routed to class r' upon its arrival at time t. Routing decisions are also made at the various nodes when jobs are admitted for service. Let $B_{rr'}(t)$ denote the event that at time t node $\sigma(r)$ is working on a class r job that will be routed to class r' upon completion of service.

In this modified setting, we are interested in devising a combined scheduling and routing policy to minimize the cost function (3). Target-pursuing (TP) policies are defined exactly as in Section III (cf. Definition 1) with the only exception that the minimization is with respect to both scheduling and routing decisions at each time t. A polyhedral relaxation $\mathcal{P}$ of the achievable region $\mathcal{A}$ can be obtained in this case as well (see [1]); an optimal solution of this achievable region LP, denoted again by $\mathbf{w}^*$, is one particular choice for $\boldsymbol{\theta}$.

To describe the implementation of TP policies in the extended model we uniformize the Markov chain $\mathbf{n}(t)$ as in Section IV. Let again ν denote the uniform transition rate and $\{\tau_k\}$ the sequence of transition epochs in the uniformized Markov chain. For any $\boldsymbol{\theta}$, Δt small enough and as specified in Section IV, and using norm (1), the TP policy minimizes

$$\frac{1}{\Delta t}\mathbf{E}\left[\|\mathbf{n}(\tau_k + \Delta t) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 \,|\, \mathbf{n}(\tau_k)\right]$$
$$= \left(\frac{1}{\Delta t} - \nu\right)\|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2$$
$$+ \sum_{r=1}^{R}\sum_{r'=1}^{R}\lambda_{0r}\mathbf{1}\{A_{rr'}(\tau_k)\}\|\mathbf{n}(\tau_k) + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2$$
$$+ \sum_{r=1}^{R}\mu_r\left[\sum_{r'=1}^{R}\mathbf{1}\{B_{rr'}(\tau_k)\}\|\mathbf{n}(\tau_k) - \mathbf{e}_r + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2\right.$$
$$\left. + \mathbf{1}\{B_{r0}(\tau_k)\}\|\mathbf{n}(\tau_k) - \mathbf{e}_r - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2\right]$$
$$+ \sum_{r=1}^{R}\mu_r\sum_{r'=0}^{R}\mathbf{1}\{\overline{B_{rr'}(\tau_k)}\}\|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\boldsymbol{\beta}}^2 + \frac{o(\Delta t)}{\Delta t}. \quad (17)$$

Let now $x_{rr'}(\tau_k) = \mathbf{1}\{B_{rr'}(\tau_k)\}$ for $r = 1, \ldots, R$ and $r' = 0, 1, \ldots, R$, $y_{rr'}(\tau_k) = \mathbf{1}\{A_{rr'}(\tau_k)\}$ for $r, r' = 1, \ldots, R$ and denote $\mathbf{x}(\tau_k) = (\mathbf{1}\{B_{10}(\tau_k)\}, \ldots, \mathbf{1}\{B_{RR}(\tau_k)\})$, $\mathbf{y}(\tau_k) = (\mathbf{1}\{A_{11}(\tau_k)\}, \ldots, \mathbf{1}\{A_{RR}(\tau_k)\})$. Discarding constants, the RHS of (17) can be written as

$$\mathbf{x}(\tau_k)'\mathbf{q}_1\left(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}\right) + \mathbf{y}(\tau_k)'\mathbf{q}_2\left(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}\right) + \frac{o(\Delta t)}{\Delta t}$$

where $\mathbf{q}_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$, $i = 1, 2$, are appropriately defined. Since for small enough Δt the first two terms dominate, to implement the TP policy with norm $\|\cdot\|_{\boldsymbol{\beta}}^2$ we will be solving the following LP problem at each epoch τ_k:

$$\textbf{(LP2)} \quad \min \quad \mathbf{x}(\tau_k)'\mathbf{q}_1(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) + \mathbf{y}(\tau_k)'\mathbf{q}_2(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$$
$$\text{s.t.} \quad \sum_{r \in C_i}\sum_{r'=0}^{R} x_{rr'}(\tau_k) \leq 1, \ \forall i$$
$$\sum_{r'=0}^{R} x_{rr'}(\tau_k) \leq n_r(\tau_k), \ \forall r$$
$$\sum_{r'=1}^{R} y_{rr'}(\tau_k) = 1, \ \forall r$$
$$\mathbf{x}(\tau_k), \mathbf{y}(\tau_k) \geq 0 \quad (18)$$

where $(\mathbf{x}(\tau_k), \mathbf{y}(\tau_k))$ is the decision vector. In the case of a work-conserving TP policy, the first inequality constraint above becomes an equality, except at nodes with no jobs present. It should be noted that situations where a class can only be routed to a subset of other class are easily accommodated; one needs to simply add constraints of the form $x_{rr'}(\tau_k) = 0$ and $y_{rr'}(\tau_k) = 0$ if r can not be routed to r'. Again, as it was the case with **(LP1)**, the work to solve **(LP2)** can be distributed across the nodes of the network with node i deciding for $x_{rr'}(\tau_k)$ and $y_{rr'}(\tau_k)$ with $r \in C_i$. Furthermore, each node needs only local

state information, i.e., state information for all classes served at the node and all classes the node can route jobs to.

Finally, the discussion of Section VI applies intact to the extended model considered here and a simulation-based method to optimize over policy parameters, θ and β, is applicable. Since the optimal solution of (**LP2**) is integer, a smooth TP policy (as in Section VI) must be employed to that end.

A. Fluid Model and the Fluid TP Policy

We now proceed to establish the stability of TP policies in the combined scheduling/routing model. Let $\tilde{A}_{rr'}(t)$, $r, r' = 1, \ldots, R$, denote the number of external class r arrivals routed to class r' upon arrival in the time interval $[0, t]$. Let also $T_{rr'}(t)$, $r = 1, \ldots, R$, $r' = 0, \ldots, R$, denote the cumulative amount of time server $\sigma(r)$ has spent in the time interval $[0, t]$ working on class r jobs that are routed to class r'. In the fluid model, for all $t \geq 0$, the dynamics of the network satisfy

$$
\dot{n}_r(t) = \sum_{r'=1}^{R} \lambda_{0r'} \tilde{a}_{r'r}(t) + \sum_{r'=1}^{R} \mu_{r'} u_{r'r}(t)
$$
$$
- \mu_r \sum_{r'=0}^{R} u_{rr'}(t) \qquad r = 1, \ldots, R
$$
$$
\sum_{r \in C_i} \sum_{r'=0}^{R} u_{rr'}(t) \leq 1 \qquad i = 1, \ldots, N
$$
$$
\sum_{r'=1}^{R} \tilde{a}_{rr'}(t) = 1 \qquad r = 1, \ldots, R
$$
$$
\tilde{a}_{rr'}(t) \geq 0 \qquad r, r' = 1, \ldots, R
$$
$$
n_r(t), u_{rr'}(t) \geq 0 \qquad r = 1, \ldots, R, \ r' = 0, \ldots, R
$$
(19)

where $u_{rr'}(t) = \dot{T}_{rr'}(t)$ is the fraction of the capacity of server $\sigma(r)$ allocated at time t to class r jobs that are routed to class r', and $\tilde{a}_{r'r}(t) = \dot{\tilde{A}}_{r'r}(t)/\lambda_{0r'}$ is the fraction of class r' external arrivals routed to class r upon their arrival at time t. The equations in (19) hold for all regular times t.

Following the same reasoning as in Section V-A, for all t and θ the fluid version of the TP policy selects the variables $\tilde{a}_{rr'}(t)$ and $u_{rr'}(t)$ to minimize

$$
\frac{d}{dt} \|\bar{\mathbf{n}}(t)\|
$$

where $\bar{\mathbf{n}}(t)$ is the fluid limit of the stochastic system satisfying (19). Regarding the constraints under which this minimization is performed, the discussion of Section V-A applies. Specifically, the FTP policy using the L2 norm of $\mathbf{n}(t)$ needs to satisfy $u_{rr'}(t) = 0$ whenever $n_r(t) = 0$ for all r, r', and t.

B. Stability Analysis

The following proposition is similar to Prop. V.1 and establishes a form of stability for the fluid model using the nonwork-conserving FTP policy under norm $\|\mathbf{n}(t)\|_\beta^2$.

Proposition VII.1: Consider the fluid model operating under the nonwork-conserving FTP policy which uses norm $\|\mathbf{n}(t)\|_\beta^2$, where $\beta > 0$. Suppose there exists a routing probability matrix

$\mathbf{P} = \{p_{rr'}\}_{r,r'=1}^{R}$ and nonnegative $y_{rr'}$, $r, r' = 1, \ldots, R$, such that

$$
\lambda_r = \sum_{r'=1}^{R} \lambda_{0r'} y_{r'r} + \sum_{r'=1}^{R} \lambda_{r'} p_{r'r}, \qquad r = 1, \ldots, R
$$
$$
\sum_{r'=1}^{R} y_{rr'} = 1 \qquad r = 1, \ldots, R
$$
$$
y_{rr'} \geq 0, \qquad r, r' = 1, \ldots, R
$$
$$
\sum_{r \in C_i} \frac{\lambda_r}{\mu_r} < 1, \qquad i = 1, \ldots, N
$$

and $(\mathbf{I} - \mathbf{P}')$ is invertible. Then for every solution of the fluid equations (19) satisfying $|\mathbf{n}(0)| \leq 1$ and $u_{rr'}(t) = 0$ whenever $n_r(t) = 0$ for all r, r', and t, there exists some $\delta(\eta) > 0$ such that for all $0 < \eta < 1$ and all $t \geq \delta$ it follows $|\mathbf{n}(t)| \leq \eta$.

Proof: The proof is similar to the one of Prop. V.1. Fix $\eta \in (0, 1)$. Let again $\mathbf{B} = \text{diag}(\beta_1, \ldots, \beta_R)$ and

$$
G(t) \triangleq \|\mathbf{n}(t)\|_\beta^2 = \mathbf{n}'(t) \mathbf{B} \mathbf{n}(t).
$$

Using the fluid model dynamics of (19) we obtain

$$
\dot{G}(t) = 2 \sum_{r=1}^{R} \beta_r n_r(t) \left[\sum_{r'=1}^{R} \lambda_{0r'} \tilde{a}_{r'r}(t) \right.
$$
$$
\left. + \sum_{r'=1}^{R} \mu_{r'} u_{r'r}(t) - \mu_r \sum_{r'=0}^{R} u_{rr'}(t) \right].
$$

Let us adopt a policy that decomposes routing and scheduling decisions. More specifically, we employ a (fixed) routing policy that uses a routing matrix $\mathbf{P}$ and nonnegative $y_{rr'}$ that satisfy the set of equations given in the statement of the proposition. As in the pure scheduling problem $u_r(t) = \sum_{r'=0}^{R} u_{rr'}(t)$ denotes the fraction of server's $\sigma(r)$ capacity allocated to class r at time t. Under this fixed routing policy, $\tilde{a}_{r'r}(t) = y_{r'r}$ and $u_{r'r}(t) = u_{r'}(t) p_{r'r}$, for all t, yielding

$$
\dot{G}(t) = 2 \sum_{r=1}^{R} \beta_r n_r(t) \left[\sum_{r'=1}^{R} \lambda_{0r'} y_{r'r} \right.
$$
$$
\left. + \sum_{r'=1}^{R} \mu_{r'} \mu_{r'}(t) p_{r'r} - \mu_r u_r(t) \right]. \quad (20)
$$

The FTP policy defined in the statement of the proposition minimizes $\dot{G}(t)$ over routing/scheduling decisions, thus, the resulting $\dot{G}(t)$ is less than or equal the one in (20) for all t.

We have now reduced the problem to the exact same scheduling problem addressed in Prop. V.1, namely, an open MQNET with fixed routing matrix $\mathbf{P}$ and external Poisson arrival rate equal to $\sum_{r'=1}^{R} \lambda_{0r'} y_{r'r}$ for class r. The first of the set of equations in the statement of the proposition is the traffic equation while the last is the usual stability condition at each node. Following the steps of the proof of Prop. V.1 we can establish the desired result. ∎

Following the same steps as in the proof of Theorem V.2 we can also establish that Proposition VII.1 implies the stability of the stochastic system. The main result for the TP policy in the combined routing/scheduling model is summarized next.

Theorem VII.2: Consider the MQNET of this section involving both sequencing and routing decisions and operated under the TP policy using norm $\|\mathbf{n}(t)\|_\beta$, where $\beta > 0$. Suppose there exists a routing probability matrix $\mathbf{P} = \{p_{rr'}\}_{r,r'=1}^R$ and nonnegative $y_{rr'}$, $r, r' = 1, \ldots, R$, such that

$$\lambda_r = \sum_{r'=1}^R \lambda_{0r'} y_{r'r} + \sum_{r'=1}^R \lambda_{r'} p_{r'r}, \qquad r = 1, \ldots, R$$

$$\sum_{r'=1}^R y_{rr'} = 1, \qquad r = 1, \ldots, R$$

$$y_{rr'} \geq 0, \qquad r, r' = 1, \ldots, R$$

$$\sum_{r \in C_i} \frac{\lambda_r}{\mu_r} < 1, \qquad i = 1, \ldots, N$$

and $(\mathbf{I} - \mathbf{P}')$ is invertible. Then, the corresponding Markov chain $\mathbf{n}(t)$ is positive Harris recurrent.

VIII. Closed Networks

In this section, we consider the case of closed MQNETs and introduce a class of TP policies for such systems. We first introduce the model and define TP policies, then discuss their implementation, and finally use fluid analysis to investigate their *efficiency*. The notion of efficiency of scheduling policies in closed networks has been introduced in [14]; to accommodate our more general model of closed networks, we will extend an efficiency sufficient condition established there.

To define the class of closed networks of interest, consider the open MQNET of Section II. Here, however, there are no external arrivals ($\lambda_{0r} = 0, \forall r$) and the probability a job exits the network is zero ($p_{r0} = 0, \forall r$). Routing is fixed and not subject to optimization; at the end of this Section we comment on how our work can be extended to address routing as well. The routing probability matrix $\mathbf{P}$ defines a number, say K, of noncommunicating classes which we call *types*; a class of type k can never be routed to a class of some other type not equal to k. We use the notation type (r) to denote the type of class r. In the closed network, we fix to S_k the number of jobs of type k and (to exclude trivial cases) assume $S_k > 0$ for all $k = 1, \ldots, K$. Let $\mathbf{S} = (S_1, \ldots, S_K)$.

Let us again uniformize the Markov chain $\mathbf{n}(t)$, use the uniform transition rate $\nu = \sum_{r=1}^R \mu_r$, and denote by $\{\tau_k\}$ the sequence of transition epochs. Denote by $\lambda_r = \lim_{k \to \infty} \mu_r \mathbf{E}[\mathbf{1}\{B_r(\tau_k)\}]$ the throughput of class r, where, as before, $B_r(\tau_k)$ denotes the event that node $\sigma(r)$ is working on class r right after time τ_k. We are interested in a scheduling policy maximizing

$$\sum_{r=1}^R h_r \lambda_r \tag{21}$$

where $\mathbf{h} = (h_1, \ldots, h_R) \geq \mathbf{0}$ are given weights.

Let $\mathbf{x}(t) = (\mathbf{1}\{B_1(t)\}, \ldots, \mathbf{1}\{B_R(t)\})$ be the vector of scheduling decisions at time t. Recall $\mathbf{M} = \mathrm{diag}(\mu_1, \ldots, \mu_R)$. We define TP policies for closed MQNETs as follows.

Definition 2: We define as target-pursuing (TP) the class of scheduling policies for closed MQNETs which at each time t minimize $\|\mathbf{Mx}(t) - \boldsymbol{\theta}\|$ for some norm $\|\cdot\|$.

In the uniformized Markov chain and for any weighted norm $\|\cdot\|_\beta$, with $\beta > 0$, implementing a TP policy amounts to solving the following optimization problem at every epoch τ_k

$$(\textbf{OPT3}) \quad \min \quad \|\mathbf{Mx}(\tau_k) - \boldsymbol{\theta}\|_\beta$$

$$\text{s.t.} \quad \sum_{r \in C_i} x_r(\tau_k) \leq 1 \qquad i = 1, \ldots, N$$

$$\mathbf{0} \leq \mathbf{x}(\tau_k) \leq \mathbf{n}(\tau_k) \tag{22}$$

where $\mathbf{x}(\tau_k)$ is the decision vector. For the weighted L2 norm of (1) this is a *quadratic programming (QP)* problem for which efficient (i.e., polynomial time) interior-point algorithms exist. The work for solving (**OPT3**) can be decomposed and distributed across the various nodes along the lines of Section IV. In the case of a work-conserving TP policy, the first inequality constraint of (**OPT3**) becomes an equality at all nodes with jobs present. It should be noted that when all classes are nonempty, the optimal solution of (**OPT3**) does not depend on time and thus, it only needs to be solved once. The resulting policy is simply the projection of $(\theta_1/\mu_1, \ldots, \theta_R/\mu_R)$ onto the feasible set of (**OPT3**), which is a static (i.e., time-independent) processor sharing policy. However, when empty classes exist, some of the decision variables are forced to zero (due to the constraint $\mathbf{x}(\tau_k) \leq \mathbf{n}(\tau_k)$) and the static policy is adjusted to avoid allocating capacity to empty classes.

As with open networks, [1] derives an LP whose optimal value is an upper bound on the optimal weighted throughput of (21). This bound is often tight and the associated optimal solution can provide one potential target $\boldsymbol{\theta}$.

For closed MQNETs the discussion of Section VI applies and one can use a simulation-based method to optimize over the policy parameters $\boldsymbol{\theta}$ and β. Notice that (**OPT3**) is a QP problem, thus, the use of a randomized policy is not necessary since the optimal solution is smooth in the policy parameters.

A. Efficiency of TP Policies for Closed Networks

We next follow [14] and discuss the efficiency of the TP policy. To that end, we work with a fluid model.

1) Fluid Model: Consider the stochastic system and let $D_r(t)$ denote the number of departures from class r in $[0, t]$, and $T_r(t)$ the amount of time server $\sigma(r)$ spends working on class r in $[0, t]$. Let also $\mathbf{z} = \mathbf{n}(0)$ denote the initial condition at time zero, assuming that $\mathbf{z}$ is in the support of $\mathbf{S}$ (i.e., $\sum_{\{r|type(r)=k\}} z_r = S_k > 0$ for all $k = 1, \ldots, K$). To obtain the fluid model we use the same fluid scaling as in open networks, and consider sequences of initial condition vectors $\mathbf{z}_l = l\mathbf{z}$ with $l \to \infty$. We use a bar to indicate various quantities of interest in the fluid model, in particular, $\bar{\mathbf{n}}(t)$ denotes the queue length vector. We have

$$\bar{\mathbf{n}}^l(t) = \frac{1}{|\mathbf{z}_l|} \mathbf{n}^l(|\mathbf{z}_l|t)$$

where superscript l indicates quantities in a system initialized with $\mathbf{z}_l$ jobs. Using the exact same analysis as in [14], for every sequence of initial conditions $\mathbf{z}_l$ there exists a subsequence $\mathbf{z}_{l_j}$ such that along this subsequence and as $l_j \to \infty$

$$\bar{\mathbf{n}}^{l_j}(\cdot, \omega) \to \bar{\mathbf{n}}(\cdot) \qquad \text{u.o.c.}$$

where ω is a sample path satisfying the SLLN for service and routing processes. This result is analogous to the one obtained in [20] for open networks. For the limit processes, we have

$$\bar{D}_r(t) = \mu_r \bar{T}_r(t), \qquad r = 1, \ldots, R$$

$$\bar{n}_r(t) = \bar{n}_r(0) + \sum_{r'=1}^{R} p_{r'r} \bar{D}_{r'}(t) - \bar{D}_r(t), \qquad r = 1, \ldots, R$$

$$\sum_{r \in C_i} \bar{T}_r(t) \le t, \qquad i = 1, \ldots, N$$

$$\sum_{r=1}^{R} \bar{n}_r(0) = 1 \qquad \sum_{\{r | \text{type}(r)=k\}} \bar{n}_r(0) = \frac{S_k}{|\mathbf{z}|}$$

$$\bar{n}_r(t), \bar{T}_r(t) \ge 0, \qquad r = 1, \ldots, R \qquad (23)$$

along with some additional equations provided in [14]. Let $u_r(t) = \dot{T}_r(t)$ denote the fraction of the capacity allocated to class r by node $\sigma(r)$, where the derivative is defined at regular times. We next derive the fluid version of the TP policy of Definition 2, using a weighted L2 norm $\| \cdot \|_{\beta}$ and target $\boldsymbol{\theta}$.

It can be seen that in the fluid limit the policy selects allocations $\mathbf{u}(t) = (u_1(t), \ldots, u_R(t))$ satisfying the dynamics in (23) and minimizing $\|\mathbf{Mu}(t) - \boldsymbol{\theta}\|_{\beta}$. Consider next the constraints under hich the TP policy in the stochastic system makes decisions (cf. (OPT3)) and note that the policy idles on an empty class. The fluid version of the TP policy selects allocations $\mathbf{u}(t)$ satisfying (23) with the additional constraint that one can not allocate capacity to empty classes. We will refer to this policy as the FTP policy.

We are now ready to formally define the notion of efficiency of scheduling policies for our closed MQNET, which is an extension of a similar definition in [14] that applies to closed networks with a single type. Consider the following LP:

$$(\mathbf{Eff} - \mathbf{LP}) \quad \max \quad \sum_{r=1}^{R} h_r \lambda_r$$

$$\text{s.t.} \quad \sum_{r \in C_i} \frac{\lambda_r}{\mu_r} \le 1 \qquad i = 1, \ldots, N$$

$$\boldsymbol{\lambda} = \mathbf{P}' \boldsymbol{\lambda} \qquad (24)$$

where $\boldsymbol{\lambda}$ is the decision vector. The first inequality constraint bounds the utilization of all servers by one and the second constraint is the set of traffic equations for the closed network. These latter equations have a unique solution up to a multiplicative constant. Let $\boldsymbol{\lambda}^* = (\lambda_1^*, \ldots, \lambda_R^*)$ be an optimal solution of $(\mathbf{Eff} - \mathbf{LP})$. We can view $\sum_r h_r \lambda_r^*$ as the maximal weighted throughput sustainable by the network. Note that at least one of the inequality (utilization) constraints is tight at the optimal solution. Any node corresponding to a tight utilization constraint at $\boldsymbol{\lambda}^*$ will be called a *bottleneck node*.

Let now $\boldsymbol{\lambda}^{\pi}(\mathbf{z})$ denote the throughput vector achieved under a stationary policy π when the closed network is initialized with $\mathbf{n}(0) = \mathbf{z}$. We define the efficiency of π as follows.

Definition 3: The stationary policy π is said to be efficient under the cost structure h if for every sequence of initial conditions $\mathbf{z}_l = l\mathbf{z}$ with $\mathbf{z}$ in the support of $\mathbf{S}$ and $l \to \infty$ we have

$$\lim_{l \to \infty} \sum_{r=1}^{R} h_r \lambda_r^{\pi}(\mathbf{z}_l) = \sum_{r=1}^{R} h_r \lambda_r^*$$

where $\boldsymbol{\lambda}^*$ is an optimal solution of $(\text{Eff} - \text{LP})$.

The following theorem generalizes [14, Th. 4.2] and provides a sufficient condition on efficiency based on the fluid limit. We omit the proof since it is very similar to the corresponding proof in [14].

Theorem VIII.1: Consider a stationary scheduling policy π under which every fluid limit satisfies

$$\limsup_{t \to \infty} \sum_{r=1}^{R} \frac{h_r \bar{D}_r(t)}{t} \ge \sum_{r=1}^{R} h_r \lambda_r^* \qquad a.s.$$

Then π is efficient under the cost structure h.

Another, and perhaps more convenient, way to express this sufficient condition is provided by the following corollary. The proof is immediate since the condition below implies the sufficient condition of Theorem VIII.1.

Corollary VIII.2: Consider a stationary scheduling policy π under which for every fluid limit there exists a time $T < \infty$ such that for all regular times $t > T$

$$\frac{d}{dt} \sum_{r=1}^{R} h_r \bar{D}_r(t) \ge \sum_{r=1}^{R} h_r \lambda_r^*.$$

Then, π is efficient under the cost structure h.

We will use Corollary VIII.2 to investigate the efficiency of the TP policies for closed networks we defined earlier. Our main result is stated in the following theorem.

Theorem VIII.3: Consider the TP policy with target $\boldsymbol{\theta} = \boldsymbol{\lambda}^*$ using a weighted L2 norm $\| \cdot \|_{\beta}$ with $\boldsymbol{\beta} > 0$. This TP policy is efficient under the cost structure h.

Proof: Recall that the FTP satisfies $u_r(t) = 0$ if $n_r(t) = 0$. Without loss of generality, assume that initially class r is empty (if multiple classes are empty, the analysis is the same). Hence, the FTP policy selects $\mathbf{u}(t) = (\lambda_1^*/\mu_1, \ldots, \lambda_{r-1}^*/\mu_{r-1}, 0, \lambda_{r+1}^*/\mu_{r+1}, \ldots, \lambda_R^*/\mu_R)$ since this minimizes $\|\mathbf{Mu}(t) - \boldsymbol{\theta}\|_{\beta}$ subject to the proper constraints. Using this policy and after a small time interval δ class r will seize to be empty (due to arrivals from other classes). At that point in time, the FTP policy switches to the allocation $\mathbf{u}^*(t) = (\lambda_1^*/\mu_1, \ldots, \lambda_r^*/\mu_r, \ldots, \lambda_R^*/\mu_R)$ since it minimizes $\|\mathbf{Mu}(t) - \boldsymbol{\theta}\|_{\beta}$ subject to the proper constraints. Notice, that we now have flow balance, i.e., the departing flow rate always equals the arriving flow rate for all classes. Therefore, at any time $t > \delta$, no class is empty, and the same allocation $\mathbf{u}^*(t)$ remains in effect. This allocation achieves a throughput of $\sum_{r=1}^{R} h_r \mu_r u_r(t) = \sum_{r=1}^{R} h_r \lambda_r^*$, that is, the TP policy is efficient under cost structure h. ∎

We conclude this section by outlining how routing decisions can be incorporated in our setting. As in Section VII, we can define variables $x_{rr'}(t) = \mathbf{1}\{B_{rr'}(t)\}$, where $B_{rr'}(t)$ denotes

the event that at time t node $\sigma(r)$ is working on a class r job that will be routed to class r' upon completion of service. Let $\mathbf{y}(t) = (y_{11}(t), \ldots, y_{RR}(t))$, where $y_{rr'}(t) = \mu_r x_{rr'}(t)$, and $\boldsymbol{\theta} = (\theta_{11}, \ldots, \theta_{RR})$. We can then define TP policies as the class of combined scheduling/routing policies which at each time t minimize $\|\mathbf{y}(t) - \boldsymbol{\theta}\|$ for some norm $\|\cdot\|$. In fact, the problem can be transformed to a pure scheduling problem. To that end, split each class r to at most $O(R)$ classes, one for every possible class r' that class r jobs can be routed to. In this modified closed MQNET, there are only scheduling decisions to be made and routing is deterministic.

IX. NUMERICAL RESULTS

Next, we present some illustrative numerical results to assess the performance of TP policies. We implement the policy under work-conserving constraints. Non-work-conserving TP policies typically result in worst performance.

A. Open Networks

The first example we consider is the two-node network of Fig. 2. In Table I we compare several work-conserving scheduling policies) with $\mathbf{h} = \mathbf{e}$. The parameters for the various traffic scenarios are listed in Table II, where we use $\boldsymbol{\rho} = (\rho_1, \rho_2)$ to denote the utilizations of nodes 1,2, respectively. We use the following abbreviations for the various traffic scenarios we considered: I.L. (imbalanced light), B.L. (balanced light), I.M. (imbalanced medium), B.M. (balanced medium), I.H. (imbalanced heavy), and B.H. (balanced heavy). The second column of Table I (ALP) lists the lower bound on optimal performance obtained by solving the achievable region LP of [1] (see Section III). The third column of Table I (DP) lists the optimal performance obtained via dynamic programming; the last row is missing because it was computationally intractable to obtain. The fourth column of Table I [TP($\mathbf{w}^*$)] reports the performance (obtained by simulation) of the TP policy using the L2 norm for $\mathbf{n}(t)$ with target $\boldsymbol{\theta}$ equal to the optimal solution $\mathbf{w}^*$ of the achievable region LP and norm weight vector $\boldsymbol{\beta} = \mathbf{e}$. The fifth column of Table I (OTP) reports the performance of the TP policy using the same norm but with optimized (as discussed in Section VI) policy parameters. For the first four rows, we only optimized over $\boldsymbol{\theta}$ and used $\boldsymbol{\beta} = \mathbf{e}$. For the last two (heavy traffic) rows we also optimized over $\boldsymbol{\beta}$ and report those results in brackets. The optimal $\boldsymbol{\beta}$ turned out to be (1,3.4,7.2) for I.H. and (1,2.6,11.2) for B.H., respectively. In the sixth column of Table I (Thr.), we list the performance of a threshold policy proposed in [3] based on heavy traffic analysis, which is conjectured to be asymptotically optimal in heavy traffic. According to this policy, priority is given to type A jobs at node 1 if the number of jobs at node 2 is below some threshold; otherwise priority is given to type B jobs. The results listed in column 6 of Table I are for the best such policy (i.e., optimized over the threshold). Finally, in the last column of Table I we report the percentage distance of the best policy we came up with (OTP column in this case) with the best other policy found. In particular, Gap $= [(\text{Best Ours}) - (\text{Best Other})] \times 100\% / (\text{Best Other})$. To facilitate the reader, we use bold for these two values.
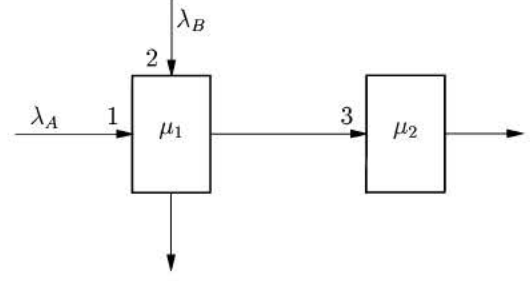

Fig. 2. Example 1: There are two types of jobs with Poisson arrival rates λ_A and λ_B (and 3 classes indicated on the figure). All jobs require exponential service time with rate μ_1 and μ_2 at nodes 1 and 2, respectively.

TABLE I
RESULTS FOR EXAMPLE 1 OF FIG. 2

	ALP	DP	TP($\mathbf{w}^*$)	OTP	Thr.	Gap
I.L.	0.63	**0.671**	0.678	**0.678**	0.679	1.0%
B.L.	0.73	**0.843**	0.856	**0.856**	0.857	1.5%
I.M.	1.9	**2.084**	2.119	**2.117**	2.129	1.6%
B.M.	2.1	**2.829**	2.96	**2.895**	2.895	2.3%
I.H.	9.6	**9.97**	10.36	10.33 **[10.13]**	10.15	1.6%
B.H.	9.9	–	18.0	17.4 **[15.5]**	**15.5**	0%

TABLE II
PARAMETERS FOR THE TRAFFIC SCENARIOS OF TABLE I

	λ_A	λ_B	μ_1	μ_2	ρ_1	ρ_2
I.L.	0.3	0.3	2	1.5	0.3	0.2
B.L.	0.3	0.3	2	1	0.3	0.3
I.M.	0.6	0.6	2	1.5	0.6	0.4
B.M.	0.6	0.6	2	1	0.6	0.6
I.H.	0.9	0.9	2	1.5	0.9	0.6
B.H.	0.9	0.9	2	1	0.9	0.9

A couple of remarks are in order. First, the TP policy using $\boldsymbol{\theta} = \mathbf{w}^*$ performs well from light to moderate traffic scenarios. This is appealing since $\mathbf{w}^*$ can be computed in polynomial-time by solving the achievable region LP. It is interesting to see that the optimal solution of this LP can lead to a fairly good policy. The optimized TP policy performs even better and is close to optimal. In the heavy-traffic cases (especially B.H.) using a weighted norm improves performance. The numerical results suggest that $\beta_3 \gg \max(\beta_1, \beta_2)$ is appropriate for those cases. This is to be expected since as $\beta_3 \to \infty$ the TP policy approaches the threshold policy of [3] with threshold θ_3 and the latter policy is known to be effective in heavy-traffic.

The second example we consider is the six-class network of Fig. 3. The results are reported in Table III, where we use the same notation and abbreviations as in Table I. The parameters for the various traffic scenarios are listed in Table IV. In the fifth column, optimization was done over $\boldsymbol{\theta}$ keeping $\boldsymbol{\beta} = \mathbf{e}$. In the last two rows of this column we also optimized over $\boldsymbol{\beta}$ and report the results in brackets. The sixth column (BPP) lists results from the best strict priority policy we were able to find. Finally, as in Table I, the last column reports the percentage gap of our best policy with the best other policy found.
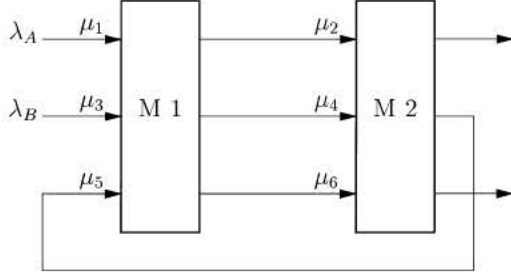
Fig. 3.   Example 2: There are two types of jobs with Poisson arrival rates λ_A and λ_B. All jobs require exponential service times with rate μ_i for class $i = 1,\ldots,6$. The cost vector $\mathbf{h}$ in the objective of (3) is set equal to $\mathbf{e}$.
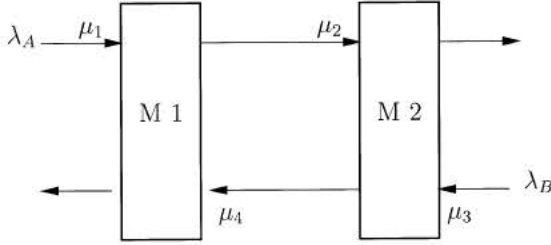

Fig. 4.   Example 3: There are two types of jobs with Poisson arrival rates λ_A and λ_B and exponential service times with rate μ_i for class i. Again, $\mathbf{h} = \mathbf{e}$.

TABLE III
RESULTS FOR EXAMPLE 2 OF FIG. 3

	ALP	DP	TP($\mathbf{w}^*$)	OTP	BPP	Gap
I.L.	0.62	**0.663**	0.684	**0.671**	0.743	1.2%
B.L.	0.71	**0.798**	0.844	**0.803**	0.916	0.3%
I.M.	1.76	**1.966**	2.15	**2.01**	2.31	2%
B.M.	1.94	**2.56**	2.81	**2.59**	3.07	0.8%
I.H.	7.63	–	9.41	8.45 [**8.32**]	9.21	-9.7%
B.H.	8.21	–	16	13.8 [**13.6**]	15.1	-9.9%

TABLE IV
PARAMETERS FOR THE TRAFFIC SCENARIOS OF TABLE III

	λ_A	λ_B	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
I.L.	3/140	3/140	1/4	2/3	1/8	1/4	1/2	3/14
B.L.	3/140	3/140	1/4	1	1/8	1/6	1/2	1/7
I.M.	6/140	6/140	1/4	2/3	1/8	1/4	1/2	3/14
B.M.	6/140	6/140	1/4	1	1/8	1/6	1/2	1/7
I.H.	9/140	9/140	1/4	2/3	1/8	1/4	1/2	3/14
B.H.	9/140	9/140	1/4	1	1/8	1/6	1/2	1/7

The conclusions in this more challenging network are similar. The TP policies with target equal to $\mathbf{w}^*$ perform quite well from light to moderate traffic scenarios. In heavy traffic, performance can further be improved by optimizing over policy parameters $(\boldsymbol{\theta}, \boldsymbol{\beta})$. Overall, we are within 2% of the optimal (when possible to compute) or we outperform by more than 9% the best other policy found.

The third example we show is the Rybko–Stolyar network [25], [26]. We used $\lambda_A = \lambda_B = 1$, $\mu_1 = \mu_3 = 6$, $\mu_2 = \mu_4 = 1.5$. It has been shown that the last-buffer–first-serve policy is unstable with these parameters. Instead, TP policies perform pretty well (Table V).

TABLE V
RESULTS FOR EXAMPLE 3 OF FIG. 4

ALP	DP	TP($\mathbf{w}^*$)	OTP	Gap
7.08	**8.18**	13.04	**8.46**	3.4%
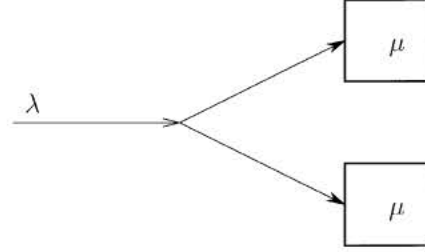
Fig. 5.   Example 4.

TABLE VI
RESULTS FOR THE ROUTING EXAMPLE (EXAMPLE 4) OF FIG. 5

Load	λ	μ	ρ	ALP	SQ	OTP$_\mathrm{n}$	Gap
Light	1.65	1.5	0.55	1.22	**1.69**	1.69	0%
Medium	2.1	1.5	0.7	2.33	**2.94**	2.94	0%
Heavy	2.7	1.5	0.9	9.00	**9.56**	9.56	0%

Our final open network example is the system of Fig. 5. Jobs arrive according to a Poisson process of rate λ and are to be routed either at the top or bottom node. Service times are exponentially distributed with rate μ at both nodes. We need to decide where to route each job in order to minimize the objective of (3) with $\mathbf{h} = \mathbf{e}$. Table VI reports our results for three traffic scenarios corresponding to $\rho = \lambda/(2\mu) = 0.55, 0.7, 0.9$, respectively. The sixth column (SQ) lists the performance of the policy that sends jobs to the shortest queue, which is known to be optimal [27]. The seventh column lists the performance of the optimized, over $\boldsymbol{\theta}$ and with $\boldsymbol{\beta} = \mathbf{e}$, TP policy using the L2 norm of $\mathbf{n}(t)$. The last column compares the two policies. It is evident that the optimized TP policy achieves optimality. This is to be expected since from the structure of $(\mathbf{LP2})$ and (17) it can be easily verified that any TP policy with target $\boldsymbol{\theta}$ such that $\theta_1 = \theta_2$ and $\boldsymbol{\beta} = \mathbf{e}$ makes routing decisions identical to the SQ policy.

B. Closed Networks

We next present two closed network examples (cf. Fig. 6). Results for these examples are reported in Table VII. The second column lists the (fixed) number of jobs in the system for each type. The third column (ALP) reports an upper bound on the optimal weighted throughput obtained by the achievable region LP of [1]. We denote by $\mathbf{w}^*$ the target obtained from the optimal solution of that LP. The fourth column (DP) lists the optimal weighted throughput obtained by solving a dynamic programming problem. The fifth column ($\mathrm{TP}(\mathbf{w}^*)$) lists the performance of the TP policy for closed networks using target $\boldsymbol{\theta} = \mathbf{w}^*$ and the L2 norm $\| \cdot \|^2$. In some instances we have optimized over the target $\boldsymbol{\theta}$. We refer to the latter policy as OTP and report the results in brackets. The sixth column lists the performance of the TP policy using target $\boldsymbol{\theta}$ equal to the optimal solution $\boldsymbol{\lambda}^*$ of $(\mathbf{Eff} - \mathbf{LP})$ in (24). Finally, the seventh

(a) Example 5   (b) Example 6
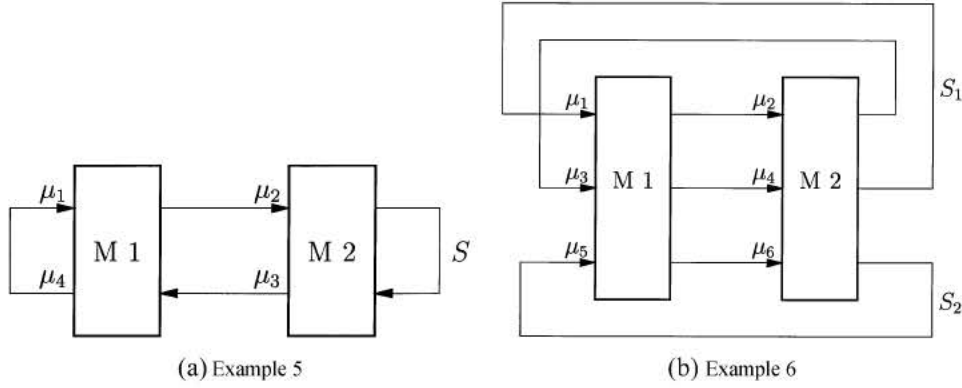
Fig. 6. Exponential service times with rate μ_i for class i. (a) Example 5: Total number of jobs equals S, $\mathbf{h} = \mathbf{e}/4$, $\boldsymbol{\mu} = (1/3, 2/7, 1, 2)$. (b) Example 6: Type 1 jobs (classes 1–4) are fixed to S_1, type 2 jobs (classes 5–6) are fixed to S_2, $\mathbf{h} = (1/4, 1/4, 1/4, 1/4, 1/2, 1/2)$, $\boldsymbol{\mu} = (8, 5, 2, 7, 4, 1)$.

TABLE VII
RESULTS FOR EXAMPLES 5 AND 6 OF FIG. 6

	S	ALP	DP	TP($\mathbf{w}^*$) [OTP]	TP($\boldsymbol{\lambda}^*$)	Gap
Ex. 5	10	10.53	**10.49**	10.28 [**10.49**]	10.27	0%
Ex. 5	100	11.05	**11.05**	**11.05**	11.05	0%
Ex. 6	5/5	1.914	**1.904**	**1.882**	1.796	1.16%
Ex. 6	5/30	1.914	**1.914**	**1.895**	1.877	0.99%
Ex. 6	30/5	1.914	**1.914**	**1.899**	1.807	0.8%

column reports the distance of our best policy from the optimal, namely, Gap = [(Optimal) − (Best Ours)] × 100%/(Optimal).

In both examples, we conclude that the TP policy with target $\mathbf{w}^*$ is rather close to optimal. As in open networks, this suggests that the optimal solution to the achievable region LP contains useful information from which a "good" policy can be obtained. The TP policy with target equal to $\boldsymbol{\lambda}^*$ performs equally well. It becomes near-optimal for large populations, which is to be expected in the light of the efficiency results of Section VIII-A. Finally, the TP policy with optimized target $\boldsymbol{\theta}$ is within less than 1.2% from the optimal in all cases considered.

X. CONCLUSION

We proposed a new class of what we call TP policies for scheduling and routing in MQNETs. These networks can model a variety of systems, including sensor networks, multiprocessor computer systems, and manufacturing systems. In open networks external arrivals were assumed to be Poisson with class-dependent rates, and in both open and closed networks service times were assumed to be exponentially distributed with class-dependent rates. These assumptions, although restrictive, can even accommodate heavy-tailed service distributions by using a hyperexponential approximation of these distributions. The fluid version of TP policies belongs to a broader class of fluid policies called *greedy* in [9]. In general, greedy or myopic policies may perform extremely poorly, nonetheless, we were able to demonstrate that our proposed class is rather effective.

In open networks, TP policies "steer" the state of the system toward a fixed target $\boldsymbol{\theta}$, where distance is measured using a weighted norm with weight vector $\boldsymbol{\beta}$. We demonstrated that TP policies are stable for any $\boldsymbol{\theta}$ under an L2 norm with weight vector $\boldsymbol{\beta} > 0$. Hence, they are safe to implement even if the parameter vector $(\boldsymbol{\theta}, \boldsymbol{\beta})$ is not optimally selected (as long as $\boldsymbol{\beta} > 0$). In closed networks, TP policies "steer" the instantaneous throughput of the various classes toward a fixed target $\boldsymbol{\theta}$, where, again, distance is measured using a weighted norm with weight vector $\boldsymbol{\beta}$. We showed that appropriate target selection leads to the efficiency of the corresponding policy, meaning, that the policy achieves maximum bottleneck throughput in the infinite population limit. In both open and closed networks, the proposed policies are amenable to distributed implementation using local state information.

In open networks, our numerical results suggest that the polyhedral relaxations of achievable performance of [1] contain enough information to yield good targets $\boldsymbol{\theta}$, especially in light to moderate load conditions. This might be sufficient in many practical situations involving sensor networks, where performance considerations would lead capacity planners to avoid heavy loads. In closed networks, an optimal solution to the achievable region LP of [1] leads to effective policies, especially for large populations. This is useful in applications to processing clusters in sensor networks which block jobs above a certain threshold to avoid performance degradation; during heavy traffic conditions the cluster can be modeled as a closed network and the population will typically be large.

The performance of the proposed class of TP policies can be further improved by optimizing over the parameter vector $(\boldsymbol{\theta}, \boldsymbol{\beta})$; we outlined how this can be done using simulation-based methods. Overall, as our numerical results indicate, we obtain near-optimal policies (when the optimal can be computed) and significantly outperform heuristic alternatives.

We close by noting that although we derived our results for networks where nodes can preempt a job to accept another, TP policies can also be implemented in a nonpreemptive setting with arbitrarily distributed service times. To that end, nodes can make decisions only at service completions by minimizing an expectation along the lines of Section IV, conditioning though not only on the current number of jobs but also on the times elapsed since the most recent service completions of other nodes.

REFERENCES

[1] D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis, "Optimization of multiclass queueing networks: polyhedral and nonlinear characterizations of achievable performance," *Ann. Appl. Probab.*, vol. 4, no. 1, pp. 43–75, 1994.

[2] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," *Math. Oper. Res.*, vol. 24, no. 2, pp. 293–305, 1999.

[3] J. M. Harrison and L. M. Wein, "Scheduling networks of queues: Heavy traffic analysis of a simple open network," *Queueing Syst. Theory Applicat.*, vol. 5, pp. 265–280, 1989.

[4] J. M. Harrison, "Brownian models of queueing networks with heterogeneous customer populations," in *Stochastic Differential Systems, Stochastic Control Theory and Applications*, W. Fleming and P. L. Lions, Eds. New York: Springer-Verlag, 1988, vol. 10, pp. 147–186.

[5] S. Kumar and P. R. Kumar, "Performance bounds for queueing networks and scheduling policies," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1600–1611, Aug. 1994.

[6] D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis, "Branching bandits and Klimov's problem: achievable region and side constraints," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 2063–2075, Dec. 1995.

[7] J. G. Dai, "On the positive Harris recurrence for multiclass queueing networks: a unified approach via fluid models," *Ann. Appl. Probab.*, vol. 5, pp. 49–77, 1995.

[8] S. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Trans. Semiconductor Manufact.*, vol. 7, pp. 374–385, Mar. 1994.

[9] S. P. Meyn, "Sequencing and routing in multiclass queueing networks. Part I: feedback regulation," *SIAM J. Control Optim.*, vol. 40, no. 3, pp. 741–776, 2001.

[10] C. Maglaras, "Discrete-review policies for scheduling stochastic networks: trajectory tracking and fluid-scale asymptotic optimality," *Ann. Appl. Probab.*, vol. 10, no. 3, pp. 897–929, 2000.

[11] N. Baeuerle, "Asymptotic optimality of tracking-policies in stochastic networks," *Ann. App. Probab.*, no. 10, pp. 1065–1083, 2000.

[12] S. Henderson, S. P. Meyn, and V. Tadic, "Performance evaluation and policy selection in multiclass networks," *Discrete Event Dyna. Syst.*, pp. 149–189, 2003.

[13] S. P. Meyn, "Sequencing and routing in multiclass queueing networks. Part II: workload relaxations," *SIAM J. Control Optim.*, vol. 42, no. 1, pp. 178–217, 2003.

[14] S. Kumar and P. R. Kumar, "Closed queueing networks in heavy traffic: Fluid limits and efficiency," in *Stochastic Networks: Stability and Rare Events*, P. Glasserman, K. Sigman, and D. Yao, Eds. New York: Springer-Verlag, 1996, vol. 117, pp. 41–64.

[15] J. Kimemia and S. B. Gershwin, "An algorithm for the computer control of a flexible manufacturing system," *IIE Trans.*, vol. 15, no. 4, pp. 353–362, 1983.

[16] G. J. Van Ryzin, S. Lou, and S. B. Gershwin, "Production control for a tandem two-machine system," *IIE Trans.*, vol. 25, no. 5, pp. 5–20, 1993.

[17] J. R. Perkins and P. R. Kumar, "Buffer level control of unreliable manufacturing systems," in *Proc. Amer. Control Conf.*, vol. 1, 1992, pp. 826–827.

[18] G. Liberopoulos and M. Caramanis, "Dynamics and design of a class of parameterized manufacturing flow controllers," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1018–1028, June 1995.

[19] J. Hu and M. Caramanis, "Dynamic set-up scheduling of flexible manufacturing systems: design and stability of near optimal general round robin policies," in *Discrete Event Systems, Manufacturing Systems, and Communication Networks*, P. R. Kumar and P. Varaiya, Eds. New York: Springer-Verlag, 1995, vol. 73, pp. 73–104.

[20] J. G. Dai, "Stability of fluid and stochastic processing networks," Georgia Inst. Technol., Atlanta, GA, Tech. Rep., 1998.

[21] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. New York: Springer-Verlag, 1993.

[22] P. Billingsley, *Probability and Measure*, 3rd ed. New York: Wiley, 1995.

[23] F. Vazquez-Abad, C. Cassandras, and V. Julka, "Centralized and decentralized asynchronous optimization of stochastic discrete-event systems," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 631–655, May 1998.

[24] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 191–209, Feb. 2001.

[25] P. R. Kumar and T. I. Seidman, "Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 289–298, Mar. 1990.

[26] A. N. Rybko and A. L. Stolyar, "Ergodicity of stochastic processes describing the operation of an open queueing network," *Problemy Peredachi Informatsii*, vol. 28, no. 3, pp. 3–26, 1992.

[27] J. Walrand, *An Introduction to Queueing Networks*. Upper Saddle River, NJ: Prentice-Hall, 1988.

Ioannis Ch. Paschalidis (M'96) was born in Athens, Greece, in 1968. He received the Diploma in electrical and computer engineering from the National Technical University of Athens, in 1991, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1993 and 1996, respectively.

After a brief period as a Postdoctoral Associate at the Laboratory for Information and Decision Systems (LIDS), MIT, he joined Boston University, MA, in September 1996, where he is currently Associate Professor of Manufacturing Engineering. During his sabbatical in 2003, he held visiting appointments with LIDS, MIT, and the Columbia University Business School, New York. His current research interests include the analysis and control of stochastic systems, large deviations theory, queueing theory, optimization, pricing, and revenue management. The main application areas he is targeting include communication and sensor networks, manufacturing systems, and supply chains. He is an Associate Editor of *Operations Research Letters* and of *Automatica*.

Dr. Paschalidis has received a National Science Foundation CAREER award (2000), second prize in the 1997 George E. Nicholson paper competition by INFORMS, and was an Invited Participant at the 2002 Frontiers of Engineering Symposium, organized by the National Academy of Engineering. He has served in the program committees of several conferences, including, the INFORMS Applied Probability Conference, the IEEE Conference on Decision and Control, and INFOCOM.

Chang Su was born in Jiangyin, China, in 1974. He received the B.S. degree in optics from the University of Science and Technology of China, Hefei, in 1996, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, in 2004.

His research interests include scheduling in multiclass queueing networks, simulation, optimization, Markov decision processes, and network control.

Michael C. Caramanis (M'01) received the B.S. degree in chemical engineering from Stanford University, Stanford, CA, in 1971, and the M.S. and Ph.D. degrees in engineering with minors in microeconomics and economic development from Harvard University, Cambridge, MA, in 1972 and 1976, respectively.

He is currently a Professor of Manufacturing Engineering at the College of Engineering, Boston University, Boston, MA, where he is also Director of the Center for Information and Systems Engineering (CISE) and Department Associate Chair. His research is supported by industry and the National Science Foundation, and has resulted in numerous journal articles published in refereed journals. From 1997 to 2001, he was the Editor of Institute of Industrial Engineers (IIE) *Transactions on Design and Manufacturing*. He is active in research addressing production planning and control of manufacturing systems, enterprise integration and supply chain management.