

# Target-Pursuing Policies for Open Multiclass Queueing Networks

Ioannis Ch. Paschalidis

Chang Su

Michael C. Caramanis

**Abstract**—We propose a new parametric class of scheduling and routing policies for open multiclass queueing networks. We establish their stability and show they are amenable to distributed implementation using localized state information. We exploit our earlier work in [1] 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.

**Index Terms**—Scheduling, Routing, Multiclass Queueing Networks, Fluid models.

## I. INTRODUCTION

RECENT trends in communications and computing have popularized the use of *application service providers (ASPs)* in running demanding applications. ASPs own a cluster of servers, including, Web, database, and other application-specific servers, often connected in a high-speed LAN. Users can access this cluster remotely to run the desired application, which can involve multiple tasks, e.g., accessing a Web interface, authentication, queries to database servers, accessing other application servers. As a result, overall performance is not only dictated by communication latencies, but, increasingly, by processing times of these tasks at the various servers.

Typical control actions that affect performance include *routing* and *scheduling* or *sequencing*. *Routing decisions* determine which server, among potentially multiple candidates, will be assigned to a particular task. *Scheduling decisions* determine which task to serve at each point in time. Scheduling can be done at both the server level, among jobs that wait to be processed by the server, and within a server among jobs that wait to access the various server resources (e.g., CPU, disk, NIC, etc.). See for example [2] on the importance of the latter sort of scheduling in Web servers.

In this paper we cast these problems in a unified framework and consider scheduling and routing in a Markovian *Multiclass open Queueing NETWORK (MQNET)*. Nodes in the network correspond to servers in the cluster and/or internal server resources

(e.g., CPU of server 1). Jobs to be processed can belong to multiple *types* differing in their arrival processes, routes through the network, processing times, and cost per unit of waiting time. The objective is to minimize a weighted sum of mean waiting times. We should note that although our main motivation is to optimize the operation of server clusters, the model we consider is rather general and applies to many other domains, including, manufacturing systems, multiprocessor computer systems, and communication networks.

Performance analysis in MQNETs is notoriously hard. Only a very special class of networks, BCMP and Kelly networks, have a product form solution. Naturally, optimizing an MQNET is an even harder problem. A version of the scheduling problem we consider has been shown to be EXP-complete in [3], i.e., an exponential-time algorithm is required to obtain an optimal policy. Under Markovian assumptions the problem can be formulated as a stochastic *dynamic programming (DP)* problem, which is only useful in solving very small instances.

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, see, e.g., [4]. [1] and [5] provide a polyhedral approximation of the region of achievable performance and obtain bounds on optimal performance. This approximation is shown to be exact in the single-node case [6]. The work on the achievable region has also led to results on stability [7]. Stability is an important and more basic question than optimization. It should be noted that in open MQNETs the usual condition of node utilizations to be less than one is not sufficient for the stability of all policies. [8] proves a seminal result establishing that the stability of a fluid model is a sufficient condition for the stability of the stochastic open MQNET. Several scheduling policies have been proposed for MQNETs, including, fluctuation smoothing policies in [9], affine shifts of policies for the fluid model [10], tracking of heavy-traffic-based policies [11], and tracking optimal trajectories of the fluid model [12].

We propose a new class of policies that “steer” the state of the system towards a pre-determined and fixed “target”. We will refer to these policies as *target-pursuing*. They are motivated by the efficiency of state feedback tracking policies in control and the work in [1]. As a first indication of their efficiency we show that they are stable. To that end, and following [8] we work with the fluid model. The selection of an appropriate target significantly affects performance. We argue that the work in [1] can lead to effective and easily computable targets. As we will see the proposed policies can be easily implemented in a decentralized manner. Scheduling and routing decisions are made at the individual nodes for the jobs they process by

Research partially supported by the NSF under a CAREER award ANI-9983221 and grant ACI-9873339 and by the ARO under the ODDR&E MURI2001 Program Grant DAAD19-01-1-0465 to the Center for Networked Communicating Control Systems.

I. Ch. Paschalidis is with Center for Information and Systems Engineering, and the Department of Manufacturing Engineering, Boston University, 15 St. Mary's St., Brookline, MA 02446, e-mail: yannis@bu.edu, url: <http://ionia.bu.edu/>.

C. Su is with the Department of Manufacturing Engineering, Boston University, e-mail: [csu@bu.edu](mailto:csu@bu.edu).

M. C. Caramanis is with the Center for Information and Systems Engineering, and the Department of Manufacturing Engineering, Boston University, e-mail: [mcaraman@bu.edu](mailto:mcaraman@bu.edu).

using only localized state information. We provide illustrative numerical results suggesting that the proposed policies are near-optimal (when the optimal can be computed) and outperform heuristic alternatives.

The remainder of the paper is organized as follows: Sec. II presents our basic model of open MQNETs where only scheduling is subject to optimization. Sec. III introduces the proposed target-pursuing policies. Sec. IV discusses implementation issues. Sec. V establishes their stability and Sec. VI outlines how to tune policy parameters. Sec. VII extends our results to MQNETs where routing is also subject to optimization. Sec. VIII reports illustrative numerical results. Conclusions are in Sec. IX.

**Notational conventions:** Throughout the paper all vectors are column vectors unless explicitly stated otherwise. We will use lower case boldface letters to denote vectors and for economy of space we will write  $\mathbf{x} = (x_1, \dots, x_R)$  for the column vector  $\mathbf{x} \in \mathbb{R}^R$ . Matrices will be denoted by boldface upper case letters and prime will denote transpose. We will use  $\mathbf{e}$  to denote the vector of all ones,  $\mathbf{0}$  for the vector of all zeroes,  $\mathbf{e}_i$  for the  $i$ th unit vector, and  $\mathbf{I}$  for the identity matrix. For any event  $A$ ,  $\bar{A}$  will denote its complement and  $\mathbf{1}\{A\}$  its indicator function. We will also use the following weighted L1 and L2 norms

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

When we drop the subscript  $\beta$  it will be assumed  $\beta = \mathbf{e}$ .

## II. THE MODEL AND KEY QUANTITIES

In this section we present the model of the open MQNET we wish to study and introduce some of our notation. We will, initially, consider a model of an MQNET involving only *sequencing decisions*. Later on, we will broaden the scope and consider open networks that involve routing decisions as well.

Consider a network that consists 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 at the various nodes, and costs per unit of waiting time. To account for jobs processed at different nodes let us define the *class* of a job as the pair of job type and node at which it is waiting for service. For example, if the network processes  $K$  job types, there can be up to  $K \times N$  classes. Let  $R$  be the total number of classes processed by the network.

We let  $\sigma(r)$  denote the node at which class  $r$  is served and  $C_i \triangleq \{r \mid \sigma(r) = i\}$  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 identify 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 processes, one for each class, with rate  $\lambda_{0r}$  for class  $r$ . Finally, service

times are independent of anything else in the network and exponentially distributed with parameter  $\mu_r$  for class  $r$ .

Let  $\mathbf{n}(t) = (n_1(t), \dots, n_R(t))$  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  $\lambda_r$  denote the total (external and internal) mean arrival rate of class  $r$  jobs, the following *traffic equations* are satisfied

$$\lambda_r = \lambda_{0r} + \sum_{r'=1}^R p_{r'r} \lambda_{r'}, \quad r = 1, \dots, R. \quad (2)$$

In matrix notation this system of equations can be written as  $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \mathbf{P}'\boldsymbol{\lambda}$ , where  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_R)$  and  $\boldsymbol{\lambda}_0 = (\lambda_{01}, \dots, \lambda_{0R})$ . Since the network is open, (2) has a unique solution given by  $\boldsymbol{\lambda} = (\mathbf{I} - \mathbf{P}')^{-1}\boldsymbol{\lambda}_0$ . Let us now denote by  $\rho_r \triangleq \lambda_r / \mu_r$  the fraction of time that server  $\sigma(r)$  spends working on class  $r$  jobs. The utilization of server  $i$  is given by  $\rho_i = \sum_{r \in C_i} \rho_r$ . For stability purposes we will be assuming that  $\rho_i < 1$  for all nodes  $i$ . Otherwise, the network is unstable in the sense that  $|\mathbf{n}(t)| \rightarrow \infty$  w.p.1 (with probability one) as  $t \rightarrow \infty$ .

Another key quantity of interest we will use later on is the solution  $\mathbf{L}(t) = (L_1(t), \dots, L_R(t))$  to the following system of equations

$$L_r(t) = n_r(t) + \sum_{r'=1}^R p_{r'r} L_{r'}(t), \quad r = 1, \dots, R, \quad (3)$$

or equivalently,  $\mathbf{L}(t) = (\mathbf{I} - \mathbf{P}')^{-1}\mathbf{n}(t)$ .  $L_r(t)$  can be interpreted as the expected (with respect to the routing process) number of jobs in the network at time  $t$  that will need to undergo class  $r$  service. Clearly,  $L_r(t) \geq n_r(t)$  a.s. (almost surely) for all  $r$  and  $t$ .

We are interested in devising a scheduling policy that minimizes the linear cost function

$$\sum_{r=1}^R h_r \mathbf{E}[n_r(t)], \quad (4)$$

where  $\mathbf{h} = (h_1, \dots, h_R)$  are given weights. Using Little's law this cost function can be easily transformed into a linear cost function involving mean waiting times.

## III. TARGET-PURSUING POLICIES

In this section we introduce the family of scheduling policies we will consider. Our motivation comes from [1] and [5] that provided a characterization of the *achievable region* for the performance vector  $\mathbf{E}[\mathbf{n}(t)]$  under all Markovian, preemptive, and stable policies. Note that by allowing randomized (non-Markovian) policies, the achievable region can be seen to be convex. We will denote by  $\mathcal{A}$  this convex achievable region; every point in  $\mathcal{A}$  is achievable by randomizing or "time-sharing" among Markovian policies.

More specifically, [1] derives a polyhedron, say  $\mathcal{P}$ , that contains the achievable region for  $\mathbf{E}[\mathbf{n}(t)]$  (see Fig. 1 for an illustration). Optimizing the linear function  $\mathbf{h}'\mathbf{E}[\mathbf{n}(t)]$  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  $\mathbf{h}'\mathbf{E}[\mathbf{n}(t)]$  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 and is exact in the case of a multi-class queue with Bernoulli feedback (Klimov’s model, see [6]). In the general network case though,  $\mathbf{w}^*$  is not in  $\mathcal{A}$  and can not be achieved by any policy. An interesting question we will try to address is whether  $\mathbf{w}^*$  “contains information” that can lead to a “good” policy.

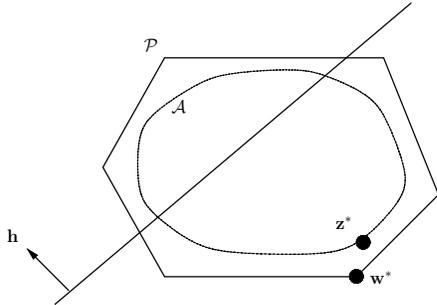


Fig. 1. The achievable region  $\mathcal{A}$  (dashed line) included in a polyhedron  $\mathcal{P}$  (solid line) obtained in [BPT94]. We denote  $\mathbf{w}^* \triangleq \arg \min_{\mathbf{z} \in \mathcal{P}} \mathbf{h}'\mathbf{z}$  and  $\mathbf{z}^* \triangleq \arg \min_{\mathbf{z} \in \mathcal{A}} \mathbf{h}'\mathbf{z}$ .

Motivated by the fact that  $\mathbf{w}^*$  can be computed efficiently (in polynomial time) and that is often “close” to the achievable region  $\mathcal{A}$  we will consider a myopic state feedback policy that aims at “steering” the state of the system towards  $\mathbf{w}^*$ . Such a policy belongs to the following class of policies.

### Definition 1

We define as target-pursuing (TP) the class of scheduling policies which at each time  $t$  minimize

$$\lim_{dt \rightarrow 0} \mathbf{E}[\|\mathbf{n}(t+dt) - \boldsymbol{\theta}\| - \|\mathbf{n}(t) - \boldsymbol{\theta}\| \mid \mathbf{n}(t)]/dt$$

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

More intuitively, TP policies select control actions to minimize  $\mathbf{E}[\|\mathbf{n}(t+dt) - \boldsymbol{\theta}\| \mid \mathbf{n}(t)]$ , for small enough  $dt$ , and are state feedback laws leading to an attractor. Similar policies have been proven useful in control systems. Note that the selection of the norm and of  $\boldsymbol{\theta}$  are intentionally left open. Moreover, as defined, TP policies are not necessarily *work-conserving* (i.e., idling a server only if no jobs are waiting). We will say *work-conserving* TP to refer to their work-conserving versions. In the sequel, we will consider the weighted norms 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.

## IV. IMPLEMENTATION ISSUES

In this section we discuss how to best implement the proposed class of 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) localized state information to perform them.

Consider the network of Section II and let us uniformize the corresponding continuous-time Markov chain with state  $\mathbf{n}(t)$ . In particular, define  $\nu = \sum_{r=1}^R \lambda_{0r} + \sum_{r=1}^R \mu_r$ , and consider the uniformized version of  $\mathbf{n}(t)$  with uniform transition rate  $\nu$ . Let  $\{\tau_k\}$  be the sequence of epochs at which the uniformized Markov chain makes transitions; this is also the sequence of ticks from a “Poisson clock” with rate  $\nu$ . As  $\mathbf{n}(t)$  is right-continuous in time,  $\mathbf{n}(\tau_k)$  refers to the state right after the  $k$ th transition. In the uniformized Markov chain, scheduling decisions need only to be made at those epochs.

Denote by  $B_r(\tau_k)$  the event that node  $\sigma(r)$  is working on class  $r$  at time  $\tau_k$ , and by  $\overline{B_r(\tau_k)}$  its complement. For any  $\boldsymbol{\theta} \in \mathbb{R}$  and any weighted norm  $\|\cdot\|_{\beta}$  it can be verified that the corresponding TP policy minimizes

$$\begin{aligned} & \nu \mathbf{E}[\|\mathbf{n}(\tau_{k+1}) - \boldsymbol{\theta}\|_{\beta} \mid \mathbf{n}(\tau_k)] - \nu \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\beta} = \\ & \sum_{r=1}^R \lambda_{0r} \|\mathbf{n}(\tau_k) + \mathbf{e}_r - \boldsymbol{\theta}\|_{\beta} \\ & + \sum_{r=1}^R \mu_r \mathbf{1}\{B_r(\tau_k)\} \left[ \sum_{r'=1}^R p_{rr'} \|\mathbf{n}(\tau_k) - \mathbf{e}_r + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\beta} \right. \\ & \quad \left. + p_{r0} \|\mathbf{n}(\tau_k) - \mathbf{e}_r - \boldsymbol{\theta}\|_{\beta} \right] \\ & + \sum_{r=1}^R \mu_r \mathbf{1}\{\overline{B_r(\tau_k)}\} \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\beta} - \nu \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\beta}. \end{aligned} \quad (5)$$

Set  $\mathbf{x}(\tau_k) = (\mathbf{1}\{B_1(\tau_k)\}, \dots, \mathbf{1}\{B_R(\tau_k)\})$ . Notice that the rhs (right hand side) of (5) is linear in  $\mathbf{x}(\tau_k)$  and write it as  $\mathbf{x}(\tau_k)' \mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$  where  $\mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^R$  is properly defined.

It can now be seen that implementing the TP policy which uses a weighted norm  $\|\cdot\|_{\beta}$  amounts to solving the following LP problem at each epoch  $\tau_k$

$$\begin{aligned} (\mathbf{LP1}) \quad & \text{minimize} \quad \mathbf{x}(\tau_k)' \mathbf{q}(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) \\ & \text{subject to} \quad \sum_{r \in C_i} x_r(\tau_k) \leq 1, \quad \forall i, \\ & \quad \mathbf{0} \leq \mathbf{x}(\tau_k) \leq \mathbf{n}(\tau_k), \end{aligned} \quad (6)$$

where  $\mathbf{x}(\tau_k)$  is the decision vector. The first inequality constraint above bounds the utilization of each server by one and the constraint  $\mathbf{x}(\tau_k) \leq \mathbf{n}(\tau_k)$  ensures that no capacity is allocated to empty classes. In the case of the work-conserving TP, the first inequality constraint becomes an equality. Under both work-conserving and non-work-conserving TP policies, the feasible set is a polytope with integer extreme points, thus, the optimal solution, say  $\mathbf{x}^*(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ , is an integer. This implies, that at each epoch  $\tau_k$  each node allocates its entire capacity to a single class (assuming a single class is also selected to break ties).

A couple of remarks on the implementation complexity are in order. The size of (LP1) is  $O(RN)$  which is polynomial in the size of the queueing network. Very large instances of linear programming problems can be solved efficiently (in polynomial time) using interior-point algorithms. For large networks, though, the computational requirements for solving (LP1) (in the form it appears in (6)) can be substantial. Fortunately, the work can be decentralized and distributed across nodes of the

network. To see that, and for simplicity of the exposition, let us consider a specific norm, namely, the L2 norm of (1) with  $\beta = \mathbf{e}$ . Similar analysis can also be done with the remaining norms that will be considered in Section V. We will use  $\mathbf{E}[\|\mathbf{n}(\tau_{k+1}) - \boldsymbol{\theta}\|^2 \mid \mathbf{n}(\tau_k)]$  as the objective function of **(LP1)**, which is equivalent to the one in (5). After a fair amount of algebra and dropping constant terms this objective function becomes equal to

$$\begin{aligned} & \sum_{r=1}^R \mu_r x_r(\tau_k)(2 - p_{r0}) + 2 \sum_{r=1}^R \mu_r x_r(\tau_k) p_{r0} (n_r(\tau_k) - \theta_r) \\ & + 2 \sum_{r=1}^R \mu_r x_r(\tau_k) \sum_{r'=1}^R p_{rr'} [n_{r'}(\tau_k) - \theta_{r'} - n_r(\tau_k) + \theta_r]. \end{aligned}$$

Let  $\mathbf{x}_i(\tau_k)' \mathbf{q}_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \beta)$  the part of the above involving only  $x_r(\tau_k)$  with  $r \in C_i$ , where  $\mathbf{x}_i(\tau_k)$  is the vector of such  $x_r(\tau_k)$  and  $\mathbf{q}_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \beta)$  is properly defined. Decomposing **(LP1)** across nodes, node  $i$  has to solve

$$\begin{aligned} \text{(D-LP1)} \quad & \min \quad \mathbf{x}_i(\tau_k)' \mathbf{q}_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \beta) \\ & \text{s.t.} \quad \sum_{r \in C_i} x_r(\tau_k) \leq 1, \\ & \quad 0 \leq x_r(\tau_k) \leq n_r(\tau_k), \quad \forall r \in C_i. \end{aligned} \quad (7)$$

Typically the number of classes served at an arbitrary node  $i$  is much less than  $R$ . Moreover, in order to solve **(D-LP1)**, node  $i$  needs state information for all local classes  $r \in C_i$ , and all classes in its local neighborhood, that is, classes  $r'$  with  $p_{rr'} > 0$  for all  $r \in C_i$ . The number of such classes  $r'$  would also be much less than  $R$  in most practical situations. As a result, **(D-LP1)** can be solved by each node using localized information and much faster than solving **(LP1)** at a central location.

## V. STABILITY ANALYSIS

In this section we establish that TP policies are *stable*. To that end, and following the work in [8], [13], we will consider a fluid model for the network, establish stability of the fluid model, and then infer the stability of the stochastic system.

### A. 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  $T_r(t)$  denote the cumulative amount of time server  $\sigma(r)$  has spent working on class  $r$  in the time interval  $[0, t]$ . In the fluid model, for all  $t \geq 0$  the dynamics of the network satisfy

$$\begin{aligned} \dot{n}_r(t) &= \lambda_{0r} + \sum_{r'=1}^R \mu_{r'} p_{r'r} u_{r'}(t) - \mu_r u_r(t), \quad \forall r, \\ \sum_{r \in C_i} u_r(t) &\leq 1, \quad i = 1, \dots, M, \\ n_r(t), u_r(t) &\geq 0, \quad r = 1, \dots, R. \end{aligned} \quad (8)$$

Here,  $u_r(t) = \dot{T}_r(t)$  and can be interpreted as the fraction of server's  $\sigma(r)$  capacity allocated to class  $r$  at time  $t$ . We will write  $\mathbf{u}(t) = (u_1(t), \dots, u_R(t))$ . The functions  $n_r(t)$  and  $u_r(t)$  are absolutely continuous, and thus, differentiable almost everywhere. The equations in (8) hold for all times  $t$  at which

$n_r(t)$  and  $u_r(t)$  are differentiable; these points in time will be referred to as *regular*.

It is convenient to write the fluid model dynamics in (8) using matrix notation. Let  $\mathbf{M} = \text{diag}(\mu_1, \mu_2, \dots, \mu_R)$ , and  $\mathbf{U}(t) = \text{diag}(u_1(t), \dots, u_R(t))$ , where  $\text{diag}(x_1, \dots, x_R)$  denotes the diagonal matrix with main diagonal  $x_1, \dots, x_R$  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  $r = 1, \dots, R$  and  $i = 1, \dots, N$ . Then, the fluid model dynamics can be written as

$$\begin{aligned} \dot{\mathbf{n}}(t) &= \boldsymbol{\lambda}_0 - (\mathbf{I} - \mathbf{P}') \mathbf{M} \mathbf{U}(t) \mathbf{e}, \\ \mathbf{C} \mathbf{u}(t) &\leq \mathbf{e}, \\ \mathbf{n}(t), \mathbf{u}(t) &\geq \mathbf{0}. \end{aligned} \quad (9)$$

For establishing our stability results, and in the numerical results we will present in Section VIII, we consider three different weighted norms with norm weights  $\beta > 0$ : (i) the L1 norm of  $\mathbf{L}(t)$  (cf. (3)),  $\beta' \mathbf{L}(t)$ , (ii) the L2 norm of  $\mathbf{L}(t)$ ,  $\|\mathbf{L}(t)\|_\beta$ , and (iii) the L2 norm of  $\mathbf{n}(t)$ ,  $\|\mathbf{n}(t)\|_\beta$ . We will next derive the fluid versions for each of these TP policies.

Note that at every decision epoch the TP policy minimizes the expression given in Definition 1 subject to the constraints of **(LP1)** (cf. (6)). 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  $k \rightarrow \infty$  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  $\omega$  of  $\mathbf{n}^k(t)$ . Let us restrict ourselves only to  $\omega$  satisfying the *strong law of large numbers (SLLN)* for the arrival, service, and routing processes. [13] proves that if  $|\mathbf{n}^k(0, \omega)|/k$  is bounded as  $k \rightarrow \infty$ , then  $\bar{\mathbf{n}}^k(\cdot, \omega)$  is precompact as  $k \rightarrow \infty$  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  $k \rightarrow \infty$  (see [14], [15]). Thus, for each sequence  $k_s \rightarrow \infty$  there exists a subsequence  $k_s \rightarrow \infty$  along which

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

for some process  $\bar{\mathbf{n}} \in \mathbb{D}^R[0, \infty)$  which is called *fluid limit* and satisfies the fluid model equations in (8). At time  $kt$  and for any of the three norms introduced above the corresponding TP policy minimizes

$$\lim_{dt \rightarrow 0} \mathbf{E}[\|\mathbf{n}(kt + dt) - \boldsymbol{\theta}\| - \|\mathbf{n}(kt) - \boldsymbol{\theta}\| \mid \mathbf{n}(kt)]/dt.$$

Scaling by  $1/k$ , and since the stochastic system converges to the fluid model for all  $\omega$  considered above, this is equivalent to minimizing

$$\lim_{dt \rightarrow 0} [\|\bar{\mathbf{n}}(t + dt/k) - \boldsymbol{\theta}/k\| - \|\bar{\mathbf{n}}(t) - \boldsymbol{\theta}/k\|]/dt.$$

Taking  $k \rightarrow \infty$ , we conclude that for all  $\boldsymbol{\theta}$  the fluid version of the TP policy seeks to minimize

$$\frac{d}{dt} \|\bar{\mathbf{n}}(t)\| \quad (10)$$

at regular  $t$ .

Consider next the constraints of **(LP1)** under which the minimization of (10) is done. We distinguish two cases based on the three norms introduced above: (a) the case of norm (iii), and (b) the case of norms (i) and (ii). In case (a) the constraints of **(LP1)** under the fluid scaling translate to the constraints in (9) with the additional constraint that for all classes  $r$  it holds  $u_r(t) = 0$  whenever  $n_r(t) = 0$ . In case (b), imposing such a constraint will not allow us to prove stability of the corresponding fluid model. In the fluid model we can in fact serve a class with zero queue length at exactly the incoming flow rate. To use this opportunity, which as we will see will enable us to prove stability, we have to slightly modify the stochastic version of our TP policies. In the interest of space, we will not get into the technical details. Briefly, at all queues of the network we introduce threshold levels (or safety stocks) that are on the order of  $\log k$ , where  $k$  is the initial total population as before. We use the TP policy of **(LP1)** when all queues are above their threshold levels. Otherwise, we implement a policy that takes all queues above their threshold levels in the minimal possible amount of time. Using results from [16] it can be shown that the fluid scaling of this modified TP policy leads to the policy that minimizes the objective in (10) subject to the constraints in (9), which permits serving a class in the fluid model even when the corresponding queue length is zero. Hereafter, and for all three norms considered, we will be referring to the fluid version of the TP policy as the *fluid target-pursuing (FTP)* policy. It aims at making scheduling decisions in order to maximize the negative drift and drive the state of the fluid model towards zero. We will also refer to the modified (with safety stocks) TP policy for the stochastic system in case (b) above as the *modified target-pursuing (MTP)* policy.

### B. Stability of the Fluid Model

We next establish the stability of the fluid model operating under the FTP policy. We start the analysis by considering the FTP policy using the weighted L1 norm  $\|\mathbf{n}(t)\|_{\beta'(\mathbf{I}-\mathbf{P}')}^{-1}$ . We show that under this policy each fluid solution  $\mathbf{n}(t)$  (i.e., a solution of the fluid model equations in (8)) reaches zero in finite time. According to the terminology in [13], the fluid model is *stable*. The result is stated in the following Proposition.

**Proposition V.1** *Consider the fluid model operating under the FTP policy which uses the weighted L1 norm  $\|\mathbf{n}(t)\|_{\beta'(\mathbf{I}-\mathbf{P}')}^{-1} = \beta'(\mathbf{I}-\mathbf{P}')^{-1}\mathbf{n}(t)$ , where  $\beta > \mathbf{0}$ . For every solution of the fluid equations (8) with  $\|\mathbf{n}(0)\| \leq 1$  there exists some  $\delta > 0$  such that  $\mathbf{n}(t) = \mathbf{0}$  for all  $t \geq \delta$ .*

**Proof :** We use a Lyapunov argument similar to the one used in [13] to show the stability of a processor sharing policy. Let  $G(t) = \|\mathbf{n}(t)\|_{\beta'(\mathbf{I}-\mathbf{P}')}^{-1} = \beta'(\mathbf{I}-\mathbf{P}')^{-1}\mathbf{n}(t)$ . To prove stability of the fluid model we will rely on [13, Lemma 2.4.7].

First note that  $G(t) = \beta'\mathbf{L}(t)$  (cf. (3)) and that  $G(t) = 0$  if and only if  $\mathbf{n}(t) = \mathbf{0}$ . Moreover, it can be shown that  $G(\mathbf{n}(t))$  is Lipschitz continuous in  $\mathbf{n}(t)$ . We will proceed by upper bounding  $\dot{G}(t)$  for all regular  $t$ . Using the fluid model equations in (9) and letting  $\mathbf{d}(t) = \mathbf{M}\mathbf{U}(t)\mathbf{e}$  we have

$$\dot{\mathbf{L}}(t) = (\mathbf{I} - \mathbf{P}')^{-1}\dot{\mathbf{n}}(t) = \boldsymbol{\lambda} - \mathbf{d}(t).$$

Thus,

$$\dot{G}(t) = \sum_{r=1}^R \beta_r(\lambda_r - d_r(t)). \quad (11)$$

Now consider the work-conserving processor sharing policy which at each node allocates capacity according to the utilization vector  $\boldsymbol{\rho} = (\rho_1, \dots, \rho_R)$ . More specifically, for all  $r = 1, \dots, R$  this policy sets

$$u_r(t) = \begin{cases} \frac{\rho_r \left( 1 - \frac{\sum_{l \in C_{\sigma(r)} |n_l(t)=0} \rho_l}{\sum_{i \in C_{\sigma(r)} |n_i(t)>0} \rho_i} \right)}{\rho_r}, & \text{if } n_r(t) > 0, \\ \rho_r, & \text{otherwise.} \end{cases} \quad (12)$$

That is, at node  $\sigma(r)$  the policy allocates a fraction  $\rho_i$  of the capacity to all empty classes  $i$  and the remaining capacity is allocated to all nonempty classes  $j$  according to their utilization  $\rho_j$ . Since  $\sum_{i \in C_j} \rho_i < 1$  is assumed for all nodes  $j$ , it can be seen that for all classes  $r$  we have  $u_r(t) > \rho_r + \epsilon$  for some  $\epsilon > 0$  when  $n_r(t) > 0$ . Therefore, using this processor sharing policy, (11) yields

$$\begin{aligned} \dot{G}(t) &\leq -\epsilon \sum_{r|n_r(t) \neq 0} \beta_r + \sum_{r|n_r(t)=0} \beta_r(\lambda_r - d_r(t)) \\ &= -\epsilon \sum_{r|n_r(t) \neq 0} \beta_r \\ &\leq -\epsilon \beta_{\min} < 0, \end{aligned} \quad (13)$$

where  $\beta_{\min} = \min_i \beta_i$ . In the equality above we have used the fact that the processor sharing policy of (12) sets  $d_r(t) = \lambda_r$  for all empty classes. Since the FTP policy described in the statement of the Proposition minimizes  $\dot{G}(t)$  for all  $t$  we have  $\dot{G}(t) \leq -\epsilon \beta_{\min} < 0$  under this latter policy as well. Hence, due to [13, Lemma 2.4.7] the fluid model under this FTP policy converges to zero within time  $G(0)/(\epsilon \beta_{\min})$ . ■

**Remark :** It should be noted that the FTP policy considered in the Proposition above is work-conserving. To see that note from Eq. (11) that a policy that minimizes  $\dot{G}(t)$  is necessarily work-conserving.

We next consider the FTP policy under the L2 norm  $\|\mathbf{L}(t)\|_{\beta}$ . In particular, as defined in Section V-A, this FTP policy minimizes  $\frac{d}{dt} \|\mathbf{L}(t)\|_{\beta}$  for each  $t$ . The following proposition proves a result which is a bit weaker than the one in Proposition V.1 but, as we will see, still sufficient to prove the stability of the stochastic system.

**Proposition V.2** *Consider the fluid model operating under the FTP policy which uses the weighted L2 norm  $\|\mathbf{L}(t)\|_{\beta}^2$ , where  $\beta > \mathbf{0}$ . For every solution of the fluid equations (8) with  $\|\mathbf{n}(0)\| \leq 1$  there exists some  $\delta > 0$  such that for all  $0 < \eta < 1$  and all  $t \geq \delta$  it follows  $\|\mathbf{n}(t)\| \leq \eta$ .*

**Proof :** Fix  $\eta \in (0, 1)$ . As in the proof of Proposition V.1 we use a Lyapunov argument to show convergence of the fluid

model in the region  $|\mathbf{n}(t)| \leq \eta$ . Let  $\mathbf{B} = \text{diag}(\beta_1, \dots, \beta_R)$  and define

$$G(t) \triangleq \|\mathbf{L}(t)\|_{\beta}^2 = \mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}(\mathbf{I} - \mathbf{P}')^{-1}\mathbf{n}(t).$$

Since  $\mathbf{L}(t) \geq \mathbf{n}(t)$  it can be seen that  $G(t) = 0$  if and only if  $\mathbf{n}(t) = \mathbf{0}$ . Furthermore,  $G(\mathbf{n}(t))$  can be shown to be locally Lipschitz continuous in  $\mathbf{n}(t)$ , that is, 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  $|G(\mathbf{n}_1(t)) - G(\mathbf{n}_2(t))| \leq \kappa(\mathcal{O})|\mathbf{n}_1(t) - \mathbf{n}_2(t)|$ .

We will derive an upper bound on the time derivative of  $G(t)$ . Using the fluid model dynamics of (9) we obtain

$$\begin{aligned} \dot{G}(t) &= 2\mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}(\mathbf{I} - \mathbf{P}')^{-1}\dot{\mathbf{n}}(t) \\ &= 2\mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}\boldsymbol{\lambda} \\ &\quad - 2\mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}\mathbf{M}\mathbf{U}(t)\mathbf{e}, \end{aligned} \quad (14)$$

where in the last equation we used the traffic equations (cf. (2)). As in the proof of Proposition V.1 we employ the work-conserving processor sharing policy of (12). As we argued there, all nonempty classes  $r$  are allocated a fraction  $u_r(t) > \rho_r + \epsilon/\mu_r$  of the capacity of node  $\sigma(r)$  for some  $\epsilon > 0$ , and all empty classes are kept empty by allocating a fraction  $\rho_r$  of the capacity to them. Using this policy and letting  $\zeta = (\mathbf{1}\{n_1(t) > 0\}, \dots, \mathbf{1}\{n_R(t) > 0\})$ , Eq. (14) yields

$$\begin{aligned} \dot{G}(t) &\leq -2\epsilon\mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}\zeta \\ &= -2\epsilon\mathbf{L}'(t)\zeta \\ &\leq -2\epsilon \sum_{i|n_i(t)>0} n_i(t) = -2\epsilon|\mathbf{n}(t)|. \end{aligned} \quad (15)$$

In the last inequality above we used the fact  $\mathbf{L}(t) \geq \mathbf{n}(t)$ . Eq. (15) establishes that  $G(t)$  is non-increasing in time.

Let now  $\beta_{\min} = \min_i \beta_i$ . Whenever  $|\mathbf{n}(t)| \geq \eta$

$$\begin{aligned} G(t) &= \sum_i \beta_i (L_i(t))^2 \geq \sum_i \beta_i (n_i(t))^2 \geq \\ &\beta_{\min} \sum_i (n_i(t))^2 \geq \beta_{\min} \frac{(\sum_i n_i(t))^2}{R} \geq \beta_{\min} \frac{\eta^2}{R}, \end{aligned} \quad (16)$$

where the third inequality above can be shown using induction. Consider the following optimization problem with optimal value equal to  $\phi^*$

$$\begin{aligned} \phi^* &= \text{minimize} && |\mathbf{n}| \\ &\text{subject to} && G(\mathbf{n}) \geq \beta_{\min} \frac{\eta^2}{R} \\ &&& \mathbf{n} \geq \mathbf{0}, \end{aligned} \quad (17)$$

where  $G(\mathbf{n}) = \mathbf{n}'\mathbf{Q}\mathbf{n}$  and  $\mathbf{Q} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}(\mathbf{I} - \mathbf{P}')^{-1}$ . Due to (16), any  $\mathbf{n} \geq \mathbf{0}$  with  $|\mathbf{n}| = \eta$  is a feasible solution of (17) and thus,  $\eta \geq \phi^* > 0$ . Furthermore, any feasible solution of (17) satisfies  $\mathbf{n} \geq \mathbf{0}$ ,  $G(\mathbf{n}) \geq \beta_{\min}\eta^2/R$  and  $|\mathbf{n}| \geq \phi^*$ . Thus, in the fluid model operated with the processor sharing policy of (12), whenever  $G(\mathbf{n}(t)) \geq \beta_{\min}\eta^2/R$  Equation (15) yields

$$\dot{G}(t) \leq -2\epsilon|\mathbf{n}(t)| \leq -2\epsilon\phi^*. \quad (18)$$

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

Suppose now that  $G(0) \geq \beta_{\min}\eta^2/R$ . Eq. (18) implies that  $G(t)$  will reach the region  $G(t) \leq \beta_{\min}\eta^2/R$  within time  $t_\eta$  satisfying

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

Furthermore,  $G(t)$  will remain in this region for all  $t \geq t_\eta$  since it is a non-increasing function of time. We conclude that for all  $t \geq t_\eta$ , it holds  $|\mathbf{n}(t)| \leq \eta$ , 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  $t \geq 0$ . ■

**Remark :** As before, the FTP policy considered in the Proposition above is work-conserving. This is seen from Eq. (14) by noting that  $\mathbf{n}'(t)(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}$  is a non-negative row vector, thus, a policy that minimizes  $\dot{G}(t)$  is necessarily work-conserving.

Finally, we consider next the non-work-conserving FTP policy under the L2 norm  $\|\mathbf{n}(t)\|_{\beta}$ , that is, the non-work-conserving policy minimizing  $\frac{d}{dt}\|\mathbf{n}(t)\|_{\beta}$  for each  $t$ . The following result is similar to Proposition V.2. In the interest of space we omit the proof which follows the general structure of the proof of Proposition V.2.

**Proposition V.3** Consider the fluid model operating under the non-work-conserving FTP policy which uses the weighted L2 norm  $\|\mathbf{n}(t)\|_{\beta}$ , where  $\beta > \mathbf{0}$ . For every solution of the fluid equations (8) satisfying  $|\mathbf{n}(0)| \leq 1$  and  $u_r(t) = 0$  whenever  $n_r(t) = 0$  for all  $r$ , there exists some  $\delta > 0$  such that for all  $0 < \eta < 1$  and all  $t \geq \delta$  it follows  $|\mathbf{n}(t)| \leq \eta$ .

### C. Stability of the Stochastic Network

We conclude our stability results by establishing that the multiclass queueing network is stable under the TP policies using the L1 and L2 norms considered in Section V-B. 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  $\mathbf{n}(t) \in \mathbb{Z}_+^R$  which evolves as a continuous-time Markov chain. The next theorem establishes that this Markov chain is *positive Harris recurrent*. Harris recurrence is defined and discussed in detail in [14], [8], [13]. In particular, letting  $\tau_B^{\mathbf{x}} = \inf\{t \geq 0 \mid \mathbf{n}(t) \in B, \mathbf{n}(0) = \mathbf{x}\}$  denote the first passage time to the set  $B$  starting from  $\mathbf{x}$ , and denoting by  $\mathcal{B}$  the Borel  $\sigma$ -field of the state space  $\mathbb{Z}_+^R$ , we say that the Markov chain  $\mathbf{n}(t)$  is *Harris recurrent* if there exists some  $\sigma$ -finite measure  $\nu$  on  $(\mathbb{Z}_+^R, \mathcal{B})$  such that whenever  $\nu(B) > 0$  and  $B \in \mathcal{B}$  it holds  $\mathbf{P}[\tau_B^{\mathbf{x}} < \infty] = 1$ . If  $\mathbf{n}(t)$  is Harris recurrent then an essentially unique invariant measure  $\pi$  exists. If this invariant measure is finite then it can be normalized to a probability measure and the Markov chain is called *positive Harris recurrent*. This also implies ergodicity [14].

**Theorem V.4** Consider the MQNET of Section II operated under one of the following TP policies

- 1) the MTP policy that uses the norm  $|\mathbf{L}(t)|_{\beta}$ ;
- 2) the MTP policy that uses the norm  $\|\mathbf{L}(t)\|_{\beta}$ ; or
- 3) the TP policy that uses the norm  $\|\mathbf{n}(t)\|_{\beta}$ ,

where  $\beta > \mathbf{0}$ . The Markov chain  $\mathbf{n}(t)$  is positive Harris recurrent.

**Proof :** For Case 1 the result follows directly from [8] and the stability of the corresponding FTP policy which has been established in Proposition V.1. For Case 2 and 3, we need to slightly modify the proof in [8]. We omit the details in the interest of space. ■

## VI. OPTIMIZING OVER POLICY PARAMETERS

In this section we discuss how we can optimize over the parameters of TP policies, namely the target  $\theta$  and the weight vector  $\beta$ , in order to obtain the best policy within the class (for each selection of the norm). To that end, we will use a simulation-based method developed in [17]. 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. An alternative is to use an actor-critic method such as the one analyzed in [18].

1) *Smooth Target-Pursuing policies:* Consider the uniformized Markov chain of Section IV and suppose we employ a TP policy using one of the weighted norms of Section V with weight vector  $\beta > \mathbf{0}$ . At each transition epoch  $\tau_k$ , scheduling decisions are made according to the optimal solution  $\mathbf{x}^*(\mathbf{n}(\tau_k), \theta, \beta)$  of (LP1). Note that if (LP1) has a unique optimal solution at  $(\theta, \beta)$ , then  $\mathbf{x}^*(\mathbf{n}(\tau_k), \theta, \beta)$  is locally constant in  $(\theta, \beta)$ . In fact,  $\mathbf{x}^*(\mathbf{n}(\tau_k), \theta, \beta)$  is piecewise constant in  $(\theta, \beta)$  with the jumps occurring at the points that the optimal solution switches from one extreme point of the feasible set to another. Hence, using a simulation-based gradient optimization method to optimize over the parameters (e.g., the one in [17]) would not be very successful since the required gradients would be zero most of the time.

To bypass this difficulty we use randomization to introduce a smoother version of our target-pursuing policies. For simplicity of the exposition, we concentrate on work-conserving TP policies; the non-work-conserving case can be handled similarly. Let  $\mathbf{y}^{(r)}(\mathbf{n}(\tau_k), \theta, \beta)$  be a feasible solution of (LP1) such that at time  $\tau_k$  class  $r$  is served at node  $\sigma(r)$  and  $\mathbf{x}^*(\mathbf{n}(\tau_k), \theta, \beta)$  is mimicked at all other nodes. Set

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

where  $\gamma > 0$  is a scalar. At time  $\tau_k$  we serve class  $r$  at node  $\sigma(r)$  with probability:

$$\alpha_r(\mathbf{n}(\tau_k), \theta, \beta) = \begin{cases} \hat{\alpha}_r(\mathbf{n}(\tau_k), \theta, \beta), & \text{if } n_r(\tau_k) > 0 \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

Notice that as  $\gamma \rightarrow 0$  all non-empty classes at a node have equal probability of being served, and as  $\gamma \rightarrow \infty$  the randomized policy converges to the policy implied by  $\mathbf{x}^*(\mathbf{n}(\tau_k), \theta, \beta)$ .

Furthermore, for each node  $j$  all quantities that involve elements of  $\mathbf{q}(\mathbf{n}(\tau_k), \theta, \beta)$  corresponding to classes not in  $C_j$  cancel from the numerator and denominator in the definition of  $\hat{\alpha}_r(\mathbf{n}(\tau_k), \theta, \beta)$ , thus,

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

where  $q_r(\cdot)$  is the  $r$ th coordinate of  $\mathbf{q}(\cdot)$ . We will be referring to this policy as the work-conserving *smooth target-pursuing (STP)* policy.

2) *Simulation-based Optimization:* We adopted the STP policy and used the simulation-based method of [17] to optimize the objective of (4) over the parameters  $(\theta, \beta)$ . In Section VIII 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 [17] (with updates at every transition of the underlying Markov chain) is shown to convergence w.p.1 to a local minimum. Stability has already been established in Section V for the TP policies for all  $\theta$  and  $\beta > \mathbf{0}$ . Thus, it will also be satisfied by the STP policy in the same parameter space for  $\gamma \gg 1$ , since the STP policy converges to the TP policy as  $\gamma \rightarrow \infty$ . Unfortunately, the required regularity conditions are not always satisfied. As a result, we had to combine random search and various heuristics with the simulation-based method of [17] to optimize over  $(\theta, \beta)$ . For example, in the case of the STP policy using any one of the L2 norms in Section V, and for fixed  $\beta$ , the transition probabilities of the corresponding Markov chain satisfy the regularity conditions of [17] and the optimization over  $\theta$  is guaranteed to converge. These transition probabilities, though, are not smooth enough with respect to  $\beta$  and we resorted to random search around  $\beta = \mathbf{e}$  to select a good  $\beta$ . Admittedly, using a simulation based method to optimize over  $\theta$  can be slow. We noticed that when we initialized the algorithm from a good target, such as  $\mathbf{w}^*$  obtained from the achievable region LP, the method converged relatively fast.

## 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. Following the line of development so far, we first discuss how to implement a TP policy in this setting and then establish stability using a fluid analysis.

We adopt the same model and notation as in Section II, indicating only the differences with the extended model we consider here. As Section II, jobs of class  $r = 1, \dots, R$  arrive to the network according to a Poisson arrival process of rate  $\lambda_{0r}$ . 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  $B_{rr'}(t)$  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 queueing network setting, we are interested in devising a combined scheduling and routing policy to minimize the linear cost function of (4). 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. 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 the target  $\boldsymbol{\theta}$ .

To describe the implementation of TP policies in the extended model we uniformize the Markov chain  $\mathbf{n}(t)$  as in Sec. IV. Let again  $\nu$  be the uniform transition rate and  $\{\tau_k\}$  the sequence of transition epochs in the uniformized Markov chain. For any  $\boldsymbol{\theta}$  and any weighted norm  $\|\cdot\|_{\beta}$  the corresponding TP policy minimizes

$$\begin{aligned} & \nu \mathbf{E}[\|\mathbf{n}(\tau_{k+1}) - \boldsymbol{\theta}\|_{\beta} \mid \mathbf{n}(\tau_k)] = \\ & \sum_{r=1}^R \sum_{r'=1}^R \lambda_{0r} \mathbf{1}\{A_{rr'}(\tau_k)\} \|\mathbf{n}(\tau_k) + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\beta} \\ & + \sum_{r=1}^R \mu_r \left[ \sum_{r'=1}^R \mathbf{1}\{B_{rr'}(\tau_k)\} \|\mathbf{n}(\tau_k) - \mathbf{e}_r + \mathbf{e}_{r'} - \boldsymbol{\theta}\|_{\beta} \right. \\ & \quad \left. + \mathbf{1}\{B_{r0}(\tau_k)\} \|\mathbf{n}(\tau_k) - \mathbf{e}_r - \boldsymbol{\theta}\|_{\beta} \right] \\ & + \sum_{r=1}^R \mu_r \sum_{r'=0}^R \mathbf{1}\{\overline{B_{rr'}}(\tau_k)\} \|\mathbf{n}(\tau_k) - \boldsymbol{\theta}\|_{\beta}. \end{aligned} \quad (20)$$

Let now  $x_{rr'}(\tau_k) = \mathbf{1}\{B_{rr'}(\tau_k)\}$  for  $r = 1, \dots, R$  and  $r' = 0, \dots, R$ ,  $y_{rr'}(\tau_k) = \mathbf{1}\{A_{rr'}(\tau_k)\}$  for  $r, r' = 1, \dots, R$  and denote  $\mathbf{x}(\tau_k) = (\mathbf{1}\{B_{10}(\tau_k)\}, \dots, \mathbf{1}\{B_{RR}(\tau_k)\})$ ,  $\mathbf{y}(\tau_k) = (\mathbf{1}\{A_{11}(\tau_k)\}, \dots, \mathbf{1}\{A_{RR}(\tau_k)\})$ . Noticing that the rhs of the above is linear in  $\mathbf{x}(\tau_k)$  and  $\mathbf{y}(\tau_k)$  we can write it as  $\mathbf{x}(\tau_k)' \mathbf{q}_1(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) + \mathbf{y}(\tau_k)' \mathbf{q}_2(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$  where  $\mathbf{q}_i(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta})$ ,  $i = 1, 2$  are appropriately defined. Implementing the TP policy with weighted norm  $\|\cdot\|_{\beta}$  amounts to solving the following LP problem at each epoch  $\tau_k$

$$\begin{aligned} (\mathbf{LP2}) \quad & \min \quad \mathbf{x}(\tau_k)' \mathbf{q}_1(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) \\ & \quad + \mathbf{y}(\tau_k)' \mathbf{q}_2(\mathbf{n}(\tau_k), \boldsymbol{\theta}, \boldsymbol{\beta}) \\ \text{s.t.} \quad & \sum_{r \in C_i} \sum_{r'=0}^R x_{rr'}(\tau_k) \leq 1, \quad \forall i \\ & \sum_{r'=0}^R x_{rr'}(\tau_k) \leq n_r(\tau_k), \quad \forall r, \\ & \sum_{r'=1}^R y_{rr'}(\tau_k) = 1, \quad \forall r, \\ & \mathbf{x}(\tau_k), \mathbf{y}(\tau_k) \geq \mathbf{0}. \end{aligned} \quad (21)$$

In the case of a work-conserving TP, the first inequality constraint above becomes an equality. 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 localized state information, i.e., state information for all classes served at the node and all classes the node can route jobs to.

Finally, for the combined scheduling/routing case as well, the discussion of Sec. VI applies and one can use a simulation-based method to optimize over the policy parameters  $\boldsymbol{\theta}$  and  $\boldsymbol{\beta}$ .

Since the optimal solution of (LP2) is integer, one has to use a smooth TP policy (as in Sec. VI) to that end.

### A. A Fluid Model and the Fluid TP Policy

We will now proceed to establish the stability of the class of TP policies in the combined scheduling/routing model.

Let  $\dot{A}_{rr'}(t)$ ,  $r, r' = 1, \dots, R$ , denote the number of external class  $r$  arrivals routed to class  $r'$  upon arrival in the time interval  $[0, t]$ . Let also  $T_{rr'}(t)$ ,  $r = 1, \dots, R$ ,  $r' = 0, \dots, R$ , 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

$$\begin{aligned} \dot{n}_r(t) &= \sum_{r'=1}^R \lambda_{0r'} \tilde{a}_{rr'}(t) + \sum_{r'=1}^R \mu_{r'} u_{rr'}(t) \\ & \quad - \mu_r \sum_{r'=0}^R u_{rr'}(t), \quad r = 1, \dots, R, \\ \sum_{r \in C_i} \sum_{r'=0}^R u_{rr'}(t) &\leq 1, \quad i = 1, \dots, N, \\ \sum_{r'=1}^R \tilde{a}_{rr'}(t) &= 1, \quad r = 1, \dots, R, \\ \tilde{a}_{rr'}(t) &\geq 0, \quad r, r' = 1, \dots, R, \\ n_r(t), u_{rr'}(t) &\geq 0, \quad r = 1, \dots, R, r' = 0, \dots, R, \end{aligned} \quad (22)$$

where  $u_{rr'}(t) = \dot{T}_{rr'}(t)$  is the fraction of server's  $\sigma(r)$  capacity allocated at time  $t$  to class  $r$  which will be routed to class  $r'$ , and  $\tilde{a}_{rr'}(t) = \dot{A}_{rr'}(t)/\lambda_{0r'}$  is the fraction of class  $r'$  external arrivals routed to class  $r$  upon their arrival at time  $t$ . The equations in (22) hold for all regular  $t$ .

Following the same reasoning as in Section V-A, for all  $t$  and  $\boldsymbol{\theta}$  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 and satisfies the fluid model equations in (22). Regarding the constraints under which this minimization is performed, the discussion of Section V-A applies. Specifically, a modified (with safety stocks) stochastic TP policy has to be used for the L1 and L2 norms of  $\mathbf{L}(t)$ , and the FTP policy in the case of 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 Proposition V.3 and establishes a certain form of stability for the fluid model using the non-work-conserving FTP policy under the L2 norm  $\|\mathbf{n}(t)\|_{\beta}$ .

**Proposition VII.1** *Consider the fluid model operating under the non-work-conserving FTP policy which uses the weighted L2 norm  $\|\mathbf{n}(t)\|_{\beta}^2$ , where  $\boldsymbol{\beta} > \mathbf{0}$ . Suppose there exists a routing probability matrix  $\mathbf{P} = \{p_{rr'}\}_{r,r'=1}^R$  and non-negative  $y_{rr'}$ ,  $r, r' = 1, \dots, R$ , such that*

$$\begin{aligned} \lambda_r &= \sum_{r'=1}^R \lambda_{0r'} y_{r'r} + \sum_{r'=1}^R \lambda_{r'} p_{r'r}, \quad r = 1, \dots, R, \\ \sum_{r'=1}^R y_{rr'} &= 1, \quad r = 1, \dots, R, \end{aligned}$$

$$y_{rr'} \geq 0, \quad r, r' = 1, \dots, R,$$

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

and  $(\mathbf{I} - \mathbf{P}')$  is invertible. Then for every solution of the fluid equations (22) 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  $\delta > 0$  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 again  $\mathbf{B} = \text{diag}(\beta_1, \dots, \beta_R)$  and define

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

Using the fluid model dynamics of (22) we obtain

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

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 non-negative  $y_{rr'}$  that satisfy the set of equations given in the statement of the proposition. As in the pure scheduling problem, let  $u_r(t) = \sum_{r'=0}^R u_{rr'}(t)$  denote the fraction of server's  $\sigma(r)$  capacity allocated to class  $r$  at time  $t$ . With this fixed routing policy we have  $\tilde{a}_{r'r}(t) = y_{r'r}$  and  $u_{r'r}(t) = u_{r'}(t)p_{r'r}$  for all  $t$ , yielding

$$G(t) = 2 \sum_{r=1}^R \beta_r n_r(t) \left[ \sum_{r'=1}^R \lambda_{0r'} y_{r'r} + \sum_{r'=1}^R \mu_{r'} u_{r'}(t) p_{r'r} - \mu_r u_r(t) \right]. \quad (23)$$

Notice that the FTP policy described in the statement of the proposition minimizes  $\dot{G}(t)$  over routing and scheduling decisions, thus, the resulting  $\dot{G}(t)$  is less than or equal the one in (23) for all  $t$ .

We have now reduced the problem to the exact same scheduling problem addressed in Proposition V.3. Here, we have an open multiclass network 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 and the last is the usual stability condition at each node. ■

Following the same steps as the proof of Theorem V.4 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 in the following theorem.

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

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

$$\sum_{r'=1}^R y_{rr'} = 1, \quad r = 1, \dots, R,$$

$$y_{rr'} \geq 0, \quad r, r' = 1, \dots, R,$$

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

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

## VIII. NUMERICAL RESULTS

In this section we present some illustrative numerical results to assess the performance of the proposed TP policies. When using the TP policy with norm  $\|\mathbf{n}(t)\|_{\beta}$  we simply implement the policy suggested by (LP1) (or (LP2) if routing decisions are to be made as well). In the other two cases, that is when using norms  $\|\mathbf{L}(t)\|_{\beta}$  and  $\|\mathbf{L}(t)\|_{\beta}$ , we have seen in Sec. V that a modified (with safety stocks) TP policy has to be used to prove stability. This was essentially due to technical problems at the boundaries  $n_r(t) = 0$  while passing from the stochastic system to the fluid model. In the numerical results that follow we ignore this modification of the stochastic TP policies. The primary reason for doing so is ease of implementation. Since we will optimize over policy parameters, it is likely that the optimization process will keep us away from poor policies. For stable policies, the introduction of safety stocks will most likely hurt performance.

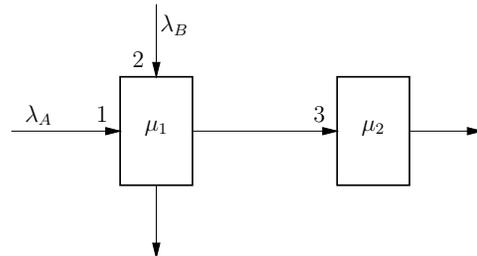


Fig. 2. Example 1: there are two types of jobs with Poisson arrival rates  $\lambda_A$  and  $\lambda_B$  (and 3 classes indicated on the figure). All jobs require an exponentially distributed service time with rate  $\mu_1$  and  $\mu_2$  at nodes 1 and 2, respectively.

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

TABLE I

The first example we consider is the two-node network of Figure 2. In Table I we compare several work-conserving scheduling policies. Scheduling here amounts to selecting which class to serve at node 1 at each point in time in order to minimize the objective of (4) with  $\mathbf{h} = \mathbf{e}$ . Let  $\rho = (\rho_1, \rho_2)$  the utilizations of nodes 1, 2, respectively. We use the following abbreviations for the various traffic scenarios we considered: I.L. (Imbalanced Light,  $\rho = (0.3, 0.2)$ ), B.L. (Balanced Light,  $\rho = (0.3, 0.3)$ ), I.M. (Imbalanced Medium,  $\rho =$

(0.6, 0.4)), B.M. (Balanced Medium,  $\rho = (0.6, 0.6)$ ), I.H. (Imbalanced Heavy,  $\rho = (0.9, 0.6)$ ), and B.H. (Balanced Heavy,  $\rho = (0.9, 0.9)$ ). The 2nd column (ALP) lists the lower bound on optimal performance obtained by solving the achievable region LP of [1] (see Sec. III). The 3rd column (DP) lists the optimal performance obtained via dynamic programming; the last row is missing because it was computationally intractable to obtain. The 4th column [ $TP_n(\mathbf{w}^*)$ ] reports the performance (obtained by simulation) of the TP policy using the L2 norm for  $\mathbf{n}(t)$  with target  $\theta$  equal to the optimal solution  $\mathbf{w}^*$  of the achievable region LP and norm weight vector  $\beta = \mathbf{e}$ . The 5th column (OTP<sub>n</sub>) reports the performance of the TP policy using the same norm but with optimized (as discussed in Sec. VI) policy parameters. For the first four rows we only optimized over  $\theta$  and used  $\beta = \mathbf{e}$ . For the last two (heavy traffic) rows we also optimized over  $\beta$  and report those results in brackets. The optimal  $\beta$  turned out to be (1, 3.4, 7.2) for I.H. and (1, 2.6, 11.2) for B.H., respectively. We do not report results for the TP policies using the L1 norm of  $\mathbf{L}(t)$  and the L2 norm of  $\mathbf{L}(t)$ ; in the former case the performance was inferior and the in latter case similar to the OTP<sub>n</sub> column. In the 6th column (Thr.) we list the performance of a threshold policy proposed in [4] based on heavy traffic Brownian approximations, 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 at station 1. The results listed in column 6 are for the best such policy (i.e., optimized over the threshold). Finally, in the last column 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,  $\text{Gap} = [(\text{Best Ours}) - (\text{Best Other})] \times 100\% / (\text{Best Other})$ . To facilitate the reader we use bold for these two values we compare.

A couple of remarks are in order. First, the TP policy using  $\theta = \mathbf{w}^*$  performs quite well from light to moderate traffic scenarios. This is quite appealing since obtaining  $\mathbf{w}^*$  can be done in polynomial-time by solving the associated 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  $\beta_3 \gg \beta_1, \beta_2$  is appropriate for those cases. This is to be expected since as  $\beta_3 \rightarrow \infty$  the TP policy approaches the threshold policy of [4] with threshold  $\theta_3$  and the latter policy is known to be effective in heavy-traffic.

	ALP	DP	$TP_n(\mathbf{w}^*)$	OTP <sub>n</sub>	$TP_L(\mathbf{w}^*)$	OTPL	BPP	Gap
I.L.	0.62	<b>0.663</b>	0.684	<b>0.671</b>	0.678	0.675	0.743	1.2%
B.L.	0.71	<b>0.798</b>	0.844	0.803	0.809	<b>0.8</b>	0.916	0.3%
I.M.	1.76	<b>1.966</b>	2.15	2.01	2.068	<b>2.005</b>	2.31	2%
B.M.	1.94	<b>2.56</b>	2.81	2.59	2.69	<b>2.58</b>	3.07	0.8%
I.H.	7.63	-	9.41	<b>8.45</b>	9.77	8.50	<b>9.21</b>	-8.3%
B.H.	8.21	-	16	13.8 [ <b>13.6</b> ]	15.3	13.8	<b>15.1</b>	-9.9%

TABLE II

The second example we consider is the six-class network of

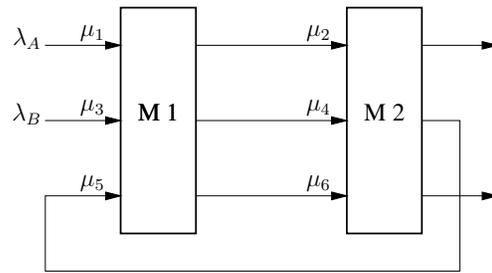


Fig. 3. Example 2: there are two types of jobs with Poisson arrival rates  $\lambda_A$  and  $\lambda_B$ . All jobs require exponentially distributed service times with rate  $\mu_i$  for class  $i = 1, \dots, 6$ . We set  $\mathbf{h} = \mathbf{e}$ .

Figure 3. The results are reported in Table II, where we use the same notation and abbreviations as in Table I. Here we also report results for the TP policy using the L2 norm for  $\mathbf{L}(t)$  with target either derived from  $\mathbf{w}^*$  (6th Column) or optimized (7th Column). In the 5th and 7th Columns optimization was done over  $\theta$  keeping  $\beta = \mathbf{e}$ . In the last row of the 5th Column we also optimized over  $\beta$  and report the result in brackets. The 8th 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.

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 ( $\theta, \beta$ ). Overall, we are within 2% of the optimal (when possible to compute) or we outperform by more than 8% the best other policy found.

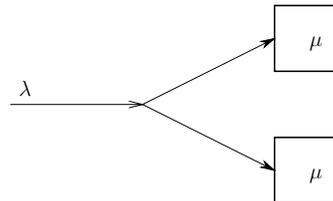


Fig. 4. Example 3.

Load	ALP	SQ	OTP <sub>n</sub>	Gap
Light	1.22	1.69	1.69	0%
Medium	2.33	2.94	2.94	0%
Heavy	9.00	9.56	9.56	0%

TABLE III

Our third and final example is the system of Figure 4. Jobs arrive according to a Poisson process of rate  $\lambda$  and are to be routed either at the top or bottom node (e.g., this could model a simple load-balancing mechanism directing requests to two Web servers). Service times are exponentially distributed with rate  $\mu$  at both nodes. We need to decide where to route each job in order to minimize the objective of (4) with  $\mathbf{h} = \mathbf{e}$ . Table III reports our results for three traffic scenarios corresponding to

$\rho = \lambda/(2\mu) = 0.55, 0.7, 0.9$ , respectively. Again we use the notation and abbreviations used in previous tables. The third column (SQ) lists the performance of the policy that sends jobs to the shortest queue, which is known to be optimal [19]. The 4th Column lists the performance of the optimized, over  $\theta$  and with  $\beta = \mathbf{e}$ , 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 (LP2) and (20) it can be easily verified that any TP policy with target  $\theta$  such that  $\theta_1 = \theta_2$  and  $\beta = \mathbf{e}$  makes routing decisions identical to the SQ policy.

## IX. CONCLUSIONS

We proposed a new class of what we called target-pursuing policies for scheduling and routing in Markovian MQNETs. These networks can model a variety of systems, including clusters of computing servers, multiprocessor computer systems, wireless sensor networks, and manufacturing systems. External arrivals were assumed to be Poisson with class-dependent rates and service times exponentially distributed with class-dependent rates. These assumptions, although restrictive, can even accommodate heavy-tailed service distributions encountered in Web servers by using a hyperexponential approximation of the heavy-tailed distribution.

The proposed policies “steer” the state of the system towards a fixed target  $\theta$ , where distance is measured using a weighted norm with weight vector  $\beta$ . We demonstrated that these policies are stable for any  $\theta$  under both an L1 and an L2 norm with weight vector  $\beta > 0$ . Hence, they are safe to implement even if the parameter vector  $(\theta, \beta)$  is not optimally selected (as long as  $\beta > 0$ ). Furthermore, they are amenable to distributed implementation using localized state information. Our numerical results suggest that the polyhedral relaxations of achievable performance obtained in [1] contain enough information to yield good targets  $\theta$ , especially in light to moderate load conditions. This might be sufficient in many practical situations involving clusters of servers, where performance considerations lead capacity planners to avoid heavy loads. Further performance improvements can be realized by optimizing over the parameter vector  $(\theta, \beta)$ ; we outlined how this can be done using simulation-based methods. As our numerical results indicate, overall and across all load conditions, we obtain near-optimal policies (when the optimal can be computed) and significantly outperform heuristic alternatives.

## REFERENCES

- [1] D. Bertsimas, I. Ch. Paschalidis, and J. N. Tsitsiklis, “Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance,” *The Annals of Applied Probability*, vol. 4, no. 1, pp. 43–75, 1994.
- [2] M.E. Crovella, R. Frangioso, and M. Harchol-Balter, “Connection scheduling in Web servers,” in *USENIX Symposium on Internet Technologies and Systems (USITS '99)*, Boulder, Colorado, October 1999, pp. 243–254.
- [3] C. H. Papadimitriou and J. N. Tsitsiklis, “The complexity of optimal queueing network control,” *Mathematics of Operations Research*, vol. 24, no. 2, pp. 293–305, 1999.
- [4] J.M. Harrison and L.M. Wein, “Scheduling networks of queues: Heavy traffic analysis of a simple open network,” *Queueing Systems Theory and Applications*, vol. 5, pp. 265–280, 1989.

- [5] S. Kumar and P.R. Kumar, “Performance bounds for queueing networks and scheduling policies,” *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1600–1611, 1994.
- [6] D. Bertsimas, I. Ch. Paschalidis, and J. N. Tsitsiklis, “Branching bandits and Klimov’s problem: Achievable region and side constraints,” *IEEE Transactions on Automatic Control*, vol. 40, no. 12, pp. 2063–2075, 1995.
- [7] P.R. Kumar and S.P. Meyn, “Stability of queueing networks and scheduling policies,” *IEEE Transactions on Automatic Control*, vol. 40, no. 2, pp. 251–260, 1995.
- [8] J.G. Dai, “On the positive Harris recurrence for multiclass queueing networks: A unified approach via fluid models,” *The Annals of Applied Probability*, vol. 5, pp. 49–77, 1995.
- [9] S.C.H. Lu, D. Ramaswamy, and P.R. Kumar, “Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374–385, 1994.
- [10] S.P. Meyn, “Sequencing and routing in multiclass queueing networks. Part I: Feedback regulation,” *SIAM J. Control and Optimization*, vol. 40, no. 3, pp. 741–776, 2001.
- [11] C. Maglaras, “Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality,” *The Annals of Applied Probability*, vol. 10, no. 3, pp. 897–929, 2000.
- [12] N. Baeuerle, “Asymptotic optimality of tracking-policies in stochastic networks,” *The Annals of Applied Probability*, , no. 10, pp. 1065–1083, 2000.
- [13] J.G. Dai, “Stability of fluid and stochastic processing networks,” Tech. Rep., Georgia Institute of Technology, 1998, Notes for a “Queueing Network Theory” course given at the University of Aarhus, Denmark.
- [14] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability*, Springer–Verlag, 1993.
- [15] P. Billingsley, *Probability and Measure*, Wiley, 3rd edition, 1995.
- [16] C. Maglaras, “Dynamic scheduling in multiclass queueing networks: Stability under discrete-review policies,” *Queueing Systems*, vol. 31, pp. 171–206, 1999.
- [17] P. Marbach and J.N. Tsitsiklis, “Simulation-based optimization of Markov reward processes,” *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 191–209, 2001.
- [18] V.R. Konda and J.N. Tsitsiklis, “Actor-critic algorithms,” Preprint, available at <http://web.mit.edu/jnt/www/home.html>, 2001, submitted to the *SIAM Journal on Control and Optimization*.
- [19] J. Walrand, *An Introduction to Queueing Networks*, Prentice Hall, 1988.