Dynamic scheduling in multiclass queueing networks: Stability under discrete-review policies

Constantinos Maglaras [a]

[a] *Columbia Business School,*
Uris Hall 409, 3022 Broadway,
New York, NY 10027-6902
E-mail: c.maglaras@columbia.edu

Received August 1997; revised November 1998

This paper describes a family of discrete-review policies for scheduling open multiclass queueing networks. Each of the policies in the family is derived from what we call a dynamic reward function: such a function associates with each queue length vector q and each job class k a positive value $r_k(q)$, which is treated as a reward rate for time devoted to processing class k jobs. Assuming that each station has traffic intensity parameter less than one, all policies in the family considered are shown to be stable. In such a policy, system status is reviewed at discrete points in time, and at each such point the controller formulates a processing plan for the next review period, based on the queue length vector observed. Stability is proved by combining elementary large deviations theory with an analysis of an associated fluid control problem. These results are extended to systems with class dependent setup times as well as systems with alternate routing and admission control capabilities.

Keywords: Scheduling, open multiclass queueing networks, discrete-review policies, fluid models, stability.

AMS Subject classification: Primary: 90B35, 60K25; Secondary: 93E15, 93D30.

1. Introduction

This paper is concerned with dynamic scheduling for open multiclass queueing networks: each customer class has its own general service time distribution, and there is Markovian routing among classes. In these networks there are many job classes that may differ in their arrival processes, service requirements, and routes through the network, and there is a many-to-one relation between job classes and servers. Decisions are to be made as to the sequence in which jobs of various classes are served at each station, and a scheduling policy is a rule according to which these sequencing decisions are made.

These models arise in many application areas such as manufacturing networks, service operations, packet switched communication networks, and multiprocessor computer systems. The main purpose of this paper is to propose a family of policies that are simple to implement and analyze, and to establish that each policy in this family guarantees stability for the controlled stochastic system. This family of stable policies provides a rich framework within which one can then pursue more interesting questions like performance optimization, guaranteed bounds on performance, and system level design and optimization.

Our focus on stability is justified by the fact that this is the most basic and fundamental property of a scheduling policy for network control. Examples developed lately by Lu and Kumar [27], Rybko and Stolyar [32], Bramson [4] and others show that networks can exhibit unanticipated instability phenomena even though the traffic intensity parameter is less than one at each station. Thus, guaranteeing stability for a network operating under a specified policy is not a simple issue. This observation has stimulated a lot of work on the stability of multiclass networks, most of which has been focused in the complete characterization of the stability region of these networks; this is the region in which the network will be stable under any non-idling scheduling policy. For example, see Dai [7], Bertsimas et al. [2], and Kumar and Meyn [23]. In a few other cases, specific policies or classes of policies have been proved stable, mainly for the class of networks referred to as re-entrant lines; examples can be found in the work of Kumar [22], Kumar and Kumar [25] and Dai [7].

This paper exploits the recent idea of a discrete-review structure introduced by Harrison in his BIGSTEP approach to dynamic flow management in multiclass networks [18], in order to define a broad family of policies that has the desired stability property. Discrete-review policies, and specifically, policies that step through time in large intervals within which a deterministic planning logic is employed, have been proposed by other researchers in the areas of applied probability and network control. Some examples that are closer to our work can be found in the work by Bertsimas and Van Ryzin [3], Bambos and Walrand [1], Tassiulas and Papavassiliou [37], and Gans and Van Ryzin [16], but other related papers can be found as well.

Each discrete-review policy in the family to be investigated is derived from what we call a dynamic reward function: such a function associates with each queue length vector q and each job class k a positive value $r_k(q)$, which is treated as a reward rate for time devoted to processing class k jobs. A constant reward function, where $r_k(q)$ does not actually depend on q, induces a discrete-review static priority scheme, whereas a general reward function induces a discrete-review dynamic priority scheme. In such a policy, system status is reviewed at discrete points in time, and at each such point the controller formulates a processing plan for the next review period based on the queue length vector observed. Formulation of the plan requires solution of a linear program whose objective function involves the dynamic reward function with which one starts. Implementation of the processing plan involves enforcement of certain safety stock requirements in order to avoid unplanned server idleness. In our variation of the Harrison's BIGSTEP method [18], both the durations of review periods and the magnitudes of safety stocks are dynamically adjusted: review periods get longer and safety stocks increase as queues lengthen, but both grow less-than-linearly as functions of queue length and hence are negligible under fluid scaling. During each review period the system is only allowed to process jobs that were present in the beginning of that period, which makes the implementation of the associated processing plans very simple.

The aim of this paper is to provide a detailed description of this family of network control policies, to analyze their behavior, and establish their stability. The method of analysis relies on the use of fluid models. Fluid models are deterministic continuous dynamical systems that nominally describe the large scale (or averaged) behavior of the associated queueing networks. Dai established in [7] that stability of the associated fluid model guarantees positive Harris recurrence for the controlled network; several refinements of this statement can be found in the work of Dai [7], Chen [6], Stolyar [35], and more recently in the work of Bramson [5]. This strong connection, together with the fact that fluid models are much simpler to analyze than the original stochastic processing networks, has been used extensively to study stability phenomena in multiclass networks. For example, see Dai [7,9], Chen [6], Meyn [30], and Dai and Weiss [11]. Similalry, fluid models will play a central role in our analysis. Moreover, it will become apparent that fluid models (or fluid approximations) are also closely related to the definition and behavior of the proposed family of control policies.

The main contributions of this paper are the following:

1. For any discrete-review policy in this family its limiting behavior under fluid scaling is examined and its associated fluid limit model is derived. This result hinges on the discrete-review structure of these policies and on some elementary large deviations estimates. This will help highlight a close connection between this family of policies and the use of fluid approximations in network control.

2. The stability of each policy in this family is established by analyzing their associated fluid models. Roughly speaking, any non-idling dynamic priority rule implemented within a discrete-review framework is shown to be stable.

3. Finally, the network models under investigation are generalized in order to allow for class dependent setup (or switchover) delays as well as routing and admission control capabilities. These extensions illustrate how within the framework of discrete-review policies and fluid approximations, one can address seemingly different control problems with only minor modifications in the conceptual and implementation levels.

The remaining of the paper is structured as follows. Section 2 describes the basic open multiclass network models, Section 3 introduces the family of discrete-review policies under study and states the main result of this paper. In section 4 the fluid model associated with a policy under investigation is derived. In section 5 a non-idling condition for the associated fluid models is established and then stability is proved for the case of a constant reward vector. Subsequently these results are extended to general reward functions. Finally, section 6 outlines the extension of these results in order to allow for routing and admission control capability and setup delays at each server, and section 7 contains some concluding remarks.

2. Open multiclass network models

In the description of a multiclass queueing network we adopt the setup introduced by Harrison [17]. Consider a queueing network of single server stations indexed by $i = 1, \ldots, S$. (The terms station and server will be used interchangeably.) The network is populated by job classes indexed by $k = 1, \ldots, K$ and infinite capacity buffers are associated with each class of jobs. Class k jobs are served by a unique station $s(k)$ and their service times are $\{\eta_k(n), n \geq 1\}$. That is, the n^{th} class k job requires $\eta_k(n)$ time units of service from station $s(k)$. Jobs within a class are served on First-In-First-Out (FIFO) basis. Upon completion of service at station $s(k)$, a class k job becomes a job of class m with probability P_{km} and exits the network with probability $1 - \sum_m P_{km}$, independent of all previous history. Assume that the general routing matrix $P = [P_{km}]$ is transient (that is, $I + P + P^2 + \ldots$ is convergent). Let $\{\phi^k(n)\}$ denote the sequence of K-dimensional IID Bernoulli random vectors such that $\phi^k_j(n) = 1$ if upon service completion the n^{th} class k job becomes a class j job and is zero otherwise, and let $\Phi^k(n) = \sum_{j=1}^{n} \phi^k(j)$. Every job class k can have its own exogenous arrival process with interarrival times $\{\xi_k(n), n \geq 1\}$. The set of classes that have a non-null exogenous arrival process will be denoted by $\mathcal{E}$ and the notation $E(t)$ will denote the K-dimensional vector of exogenous arrivals in the time interval $[0, t]$. It is assumed that $\mathcal{E} \neq \emptyset$.

We make the following assumptions on the distributional characteristics of the arrival and service time processes:

(A1) $\xi_1, \ldots, \xi_K$ and $\eta_1, \ldots, \eta_K$ are mutually independent, positive, IID sequences;

(A2) $\mathbf{E}[\eta_k(1)] \neq 0$ for $k = 1, \ldots, K$. For some $\theta > 0$, $\mathbf{E}[e^{\theta \eta_k(1)}] < \infty$ for $k = 1, \ldots, K$ and $\mathbf{E}[e^{\theta \xi_k(1)}] < \infty$ for $k \in \mathcal{E}$;

(A3) For any $x > 0$, $k \in \mathcal{E}$, $\mathbf{P}\{\xi_k(1) \geq x\} > 0$. Also, for some positive function $p(x)$ on $\mathbf{R}_+$ with $\int_0^\infty p(x)dx > 0$, and some integer j_0, $\mathbf{P}\left\{\sum_{i=1}^{j_0} \xi_k(i) \in dx\right\} \geq p(x)dx$.

Condition **(A1)** can be relaxed; see the remark after Proposition 2.1 of Dai [7]. **(A2)** is stronger than the finite first moment condition usually imposed (see, for example, Dai [7]), and it is needed in the derivation of large deviation bounds required in our analysis. This condition is satisfied for $\{\phi^k(n)\}$. The technical regularity conditions in **(A3)** are imposed so that we can make use of the general stability theory of Dai [7] (see also Dai and Meyn [10]); these conditions are never invoked in propositions that are actually proved in this work.

For future reference, let $\lambda_k = 1/\mathbf{E}[\xi_k(1)]$ and $\mu_k = 1/\mathbf{E}[\eta_k(1)] = 1/m_k$ be the arrival and service rates respectively for class k jobs, let $\lambda = (\lambda_1, \ldots, \lambda_K)'$, and let $M = \mathbf{diag}\{m_1, \ldots, m_K\}$. The set $\{k : s(k) = i\}$, denoted C_i, is called the constituency of the server i, while the $S \times K$ constituency matrix C will be the following incidence matrix:

$$C_{ik} = \begin{cases} 1, & \text{if } s(k) = i \\ 0, & \text{otherwise.} \end{cases}$$

Given the Markovian routing structure of these networks and the transience of P, one can compute the vector of effective arrival rates, $\alpha = (I - P')^{-1}\lambda$, and the vector of traffic intensities $\rho = CR^{-1}\lambda$, where ρ_i denotes the nominal load (or utilization level) for server i. Hereafter, it will be assumed that $\alpha > 0$; this restriction is imposed in order to simplify the policy description of section 3, and can easily be relaxed (some necessary modifications will be outlined later). Moreover, we assume that the traffic intensity at every station is less than one, or equivalently that there is enough processing capacity to cope with the incoming traffic; that is,

(A4) $$\rho = CR^{-1}\lambda < \mathbf{1},$$

where $\mathbf{1}$ denotes the vector of ones of appropriate dimension and inequalities are to be interpreted componentwise.

Denote by $Q_k(t)$ the total number of class k jobs in the system at time t, and by $Q(t)$ the corresponding K-vector of "queue lengths." A generic value of $Q(t)$ will be denoted by q, and the size of this vector is defined as $|q| = \sum_k q_k$. (To avoid confusion the reader should note that $|A|$ will also denote the cardinality of a set A, but the intended use of the notation will always be clear.) A scheduling policy is a rule according to which resource allocation decisions are made over time starting from an initial condition y (apart from the initial queue length configuration, y may include additional information regarding the initial state of the system It takes the form of a K-dimensional cumulative allocation process $\{T^y(t),\ t \geq 0;\ T^y(0) = 0\}$, where $T^y_k(t)$ denotes the time allocated by server $s(k)$ into serving class k jobs up to time t, and the superscript "y" denotes the dependence on the initial condition. Since the process T^y is Lipschitz, one can define its derivative $\dot{T^y}(t) \triangleq dT^y(t)/dt$, where $\dot{T^y}_k(t)$ will be the fraction of effort allocated into processing class k jobs by server $s(k)$ at time t. For an admissible policy, $\dot{T}^y(t)$ is non-negative, it satisfies the capacity constraints $C\dot{T}^y(t) \leq \mathbf{1}$, and also $\dot{T}^y_k(t)$ can only be positive if $Q_k(t) > 0$. In addition, an admissible policy needs to be *non-anticipating* (or causal), which, roughly speaking, ensures that $\dot{T^y}(t)$ only depends on information available up to time t and does not require information regarding the future. For purposes of this paper we will avoid a precise statement of this condition, since it will not be required towards the development of our results and it would otherwise involve a fairly subtle and technical exposition. Each server can only process one job at a time and thus, $\dot{T}^y_k(t)$ is equal to 1 if a class k job is being processed at time t, and 0 otherwise. For concreteness we assume non-preemptive type of service. Let $I^y(t)$ be the S-dimensional cumulative idleness process defined by $I^y(t) = \mathbf{1}t - CT^y(t)$, where $I^y_i(t)$ is the total time that server i has been idled up to time t. The process $I^y(t)$ is non-decreasing.

Given any admissible scheduling policy, a Markovian state descriptor can be constructed and an underlying Markov chain can be identified for the controlled network. The Markovian state is based on the queue length vector, as well as other auxiliary quantities that depend on the distributional characteristics of the interarrival and service processes and on the scheduling rule used. The Markovian state at time t will be denoted by $Y(t)$ and the corresponding normed state space will be $(\mathbf{Y}, \|\cdot\|)$; see the comments by Dai and Meyn [10, section IIb] or Bramson [5, section 3] regarding the choice of $\|\cdot\|$. Examples of such Markov chain constructions can be found in Dai [7].

Finally, a few brief comments on fluid scaling and fluid limit models. Consider a sequence of initial conditions $\{y^n\} \subset \mathbf{Y}$ such that $\|y^n\| \to \infty$ as $n \to \infty$. For any real valued process $\{f^y(t), t \geq 0\}$ define its fluid scaled counterpart by

$$\bar{f}^n(t) = \frac{1}{\|y^n\|} f^{y^n}(\|y^n\|t). \tag{2.1}$$

Applying this type of scaling on the queue length and cumulative allocation processes and using the functional strong law of large numbers, one can show that almost surely the pair of queue length and cumulative allocation limit trajectories $(q(\cdot), \bar{T}(\cdot))$ satisfies the following set of equations

$$q(t) = q(0) + \lambda(t\mathbf{1} - \bar{R}_a)^+ - (I - P')M^{-1}(\bar{T}(t) - \bar{R}_s)^+, \tag{2.2}$$

$$q(t) \geq 0 \quad \text{for } t \geq 0, \tag{2.3}$$

$$\bar{I}(t) = \mathbf{1}t - C\bar{T}(t), \;\; \bar{T}(0) = 0, \tag{2.4}$$

$$\bar{T}(t), \;\; \bar{I}(t) \;\; \text{are non-decreasing for } t \geq 0, \tag{2.5}$$

together with some additional conditions on $(q(\cdot), \bar{T}(\cdot))$ that are specific to the scheduling policy employed. In the sequel, the overbar notation will signify fluid scaled quantities and appropriate superscripts will be used to signify the scaled processes corresponding to some initial condition along the sequence $\{y^n\}$. In order to avoid the use of double superscripts the dependence to the initial condition y^n will be denoted by a single superscript n. The use of the overbar notation without any superscript n will denote the fluid limit of the appropriate variable; for example $\bar{T}(\cdot)$ as the limit of $\bar{T}^n(n\cdot)$.

The above set of equations are referred to as the *delayed fluid model* associated with a multiclass queueing network under a specified scheduling policy. It is immediate from (2.2)-(2.5) that the limit processes $(q, \bar{T})$ are Lipschitz continuous. Hence, it follows that that they have a time derivative almost everywhere; see Lemma 2.1 of Dai and Weiss [11]. A path $q(\cdot)$ is called regular at t if it is differentiable at t, and its derivative at time t will be denoted by $\dot{q}(t)$. Let $v(t)$ denote the instantaneous *fluid* allocation vector at time t. Restricting attention to the case where $\bar{R}_a = \bar{R}_s = 0$ and using the a.e. differentiability of the limit processes, for almost all times $t \geq 0$ the fluid limit model can be expressed as a linear dynamical system with polytopic constraints in $v(t)$ of the following form:

$$\dot{q}(t) = \lambda - Rv(t), \quad q(0) = z, \tag{2.6}$$

$$q(t) \geq 0, \quad Cv(t) \leq \mathbf{1}, \quad v(t) \geq 0 \;\; \text{for } t \geq 0, \tag{2.7}$$

together with some policy specific conditions. The fluid limit model in (2.6)-(2.7) is called *undelayed*. We will say that that $(q, v) \in FM$ -or equivalently that it is a *fluid solution*- if this pair of state and input trajectories satisfies equations (2.6)-(2.7). Undelayed limits can be obtained if one restricts attention to exponential interarrival and service time processes, or in the case of general distributions, if one lets $\|y^n\| \to \infty$ while keeping $R_a^n(0)$ and $R_s^n(0)$ bounded.

An excellent exposition of fluid models, their derivation, and their properties can be found in Dai [7,8] and Bramson [5, section 4].

3. Discrete-Review Policies

The family of policies we propose and analyze is based on the recent idea of a discrete-review structure introduced by Harrison in his BIGSTEP approach to dynamic flow management in multiclass networks [18]. The main idea behind discrete-review policies, is that one steps through time in large intervals within which a deterministic planning logic is employed, and that the scheduling and execution steps in the corresponding systems becomes more efficient as the planning horizons become longer or equivalently, as the amount of work to be processed within each period increases.

3.1. Definition of a discrete-review control policy

A discrete-review policy is defined by or is derived from a function $l : \mathbf{R}_+ \to \mathbf{R}_+$, the function $r : \mathbf{R}_+^K \to \mathbf{R}^K$, plus a K-dimensional vector β that satisfy the following restrictions. First, $l(\cdot)$ is real valued, strictly positive, concave, and further satisfies

$$\frac{l(x)}{\log(x)} > c_0 \quad \text{and} \quad \frac{l(x)}{\log(x)} \to \infty \quad \text{as } x \to \infty, \tag{3.1}$$

and

$$\frac{l(x)}{x} \to 0 \quad \text{as } x \to \infty. \tag{3.2}$$

The significance of the growth condition (3.1) will become apparent in section 4. Second, $r(\cdot)$ defined on $\mathbf{R}_+^K$, is real valued, strictly positive, and continuous function, where each component of which satisfies the growth condition

$$c_1 \le r_k(q) \le c_2 + |q|^{c_3} \quad \text{for some} \quad c_1 > 0,\ c_2 > 0,\ c_3 > 0 \quad \text{and } k = 1, \ldots, K. \tag{3.3}$$

And third, β is a vector in $\mathbf{R}_+^K$ that satisfies

$$\beta > \mu, \tag{3.4}$$

where μ is the K-vector of service rates. Under any of the policies to be considered, system status will be observed at a sequence of times $0 = t_0 < t_1 < t_2 < \ldots$; we call t_j the j^{th} review point and the time interval between t_j and t_{j-1} the j^{th} planning period. Define $l_0 = al(|Q(0)|)$, where a is a small ($\ll 1$) positive constant, independent of $|Q(0)|$; the value of this constant is not crucial to the operation of the proposed discrete-review policy and for that reason it is not included as one of the defining quantities of these policies. Given that the queue length vector $q = Q(t_j)$ is observed at t_j, server activities over the next planning period are determined by solving a linear program, the data for which involve $l(\cdot)$, $r(\cdot)$, and β. To be specific, having observed q the controller sets $\tilde{q} = q/|Q(0)|$,

$$l = l_0 \vee l(|q|), \;\; r = r(\tilde{q}), \;\text{ and } \theta = l\beta, \tag{3.5}$$

and then solves the following linear program: choose a K-vector x of time allocations to

$$\text{maximize} \;\; r'x \tag{3.6}$$

$$\text{subject to} \;\; q + l\lambda - Rx \geq \theta, \quad x \geq 0, \quad Cx \leq l\mathbf{1}. \tag{3.7}$$

First, an interpretation of this planning logic will be provided assuming that this linear program is feasible; the case where (3.6)-(3.7) is infeasible will be dealt with shortly. Intuitively, the controller first computes the nominal length of the planning period $l(|q|)$, and a target safety stock θ to be maintained upon completion of this planning period, as a function of the observed queue length vector. In general, $l_0 \ll l(|q|)$, unless $|q|$ is very small in which case l_0 provides a lower bound on the planning horizon which is in the appropriate time scale. Then, the nominal time allocations for the ensuing planning period are computed using the linear program (3.6)-(3.7): the decision variable x_k represents the nominal amount of time that will be devoted to serving class k jobs over this planning period. The constraint $q+l\lambda-Rx \geq \theta$ implies that the target ending queue length vector will be above a specified threshold requirement, while $Cx \leq l\mathbf{1}$ states that the total time allocation for each of the servers cannot exceed its capacity. It is implicit in this formulation that the planning problem involves a deterministic "continuous variable" approximation.

The objective of the linear program (3.6) is defined using the function $r(\cdot)$. Hereafter, $r(\cdot)$ will be referred to as a *dynamic reward function*: it associates with each (appropriately normalized) queue length vector $\tilde{q}$ a corresponding K-vector $r(\tilde{q})$, where the k^{th} component $r_k(\tilde{q})$ is treated as a reward rate for time devoted to processing class k jobs. In the planning problem (3.6)-(3.7) one seeks to determine a vector x of time allocations over the planning period that maximizes total reward subject to the constraints explained above. This is essentially a transient optimization procedure. If one interprets $r(\cdot)$ as a reward function according to which this transient optimization process should be performed, then the transformation of q to $\tilde{q}$ simply reduces the planning phase for each review period to a *normalized* optimization problem.

Given the vector of nominal time allocations x, a plan expressed in units of jobs of each class to be processed over the ensuing period, and a nominal idleness plan expressed in units of time for each server to remain idle over the same period are formed as follows:

$$p(k) = \left\lfloor \frac{x_k}{m_k} \right\rfloor \wedge q_k \quad \text{for } k = 1, \ldots, K, \;\; \text{and} \;\; u_i = l - (Cx)_i \quad \text{for } i = 1, \ldots, S. \tag{3.8}$$

The execution of these decisions is as follows. First, the plan p is implemented in open-loop fashion; that is, each server i processes sequentially $p(k)$ jobs for each class $k \in C_i$. The construction of the processing plan p using equation (3.8) ensures that it will be implementable from jobs present at the beginning of this review period. Let d_i denote the time taken to complete processing of these jobs at server i and let $\delta_i = (l - d_i)^+$ be the nominal time remaining until the end of the ensuing period. In the second phase of execution, each server i will idle for $u_i \wedge \delta_i$ time units. The completion of both execution phases signals the beginning of the $(j+1)^{st}$ review period. The total duration of execution of the j^{th} review period will be denoted T^{exe}.

When the planning linear program (3.6)-(3.7) is infeasible, a alternative logic is employed to steer the state above the desired threshold levels. The first step of this infeasible planning algorithm is summarized in the following linear program: find a scalar $\hat{l}$ and a K-vector $\hat{x}$ to

$$\text{minimize} \quad \hat{l} \tag{3.9}$$

$$\text{subject to} \quad \hat{l}\lambda - R\hat{x} > \beta + \mathbf{1}, \quad \hat{x} \geq 0, \quad \hat{l} \geq 0, \quad C\hat{x} \leq \hat{l}\mathbf{1}. \tag{3.10}$$

Given the solution of the linear program (3.9)-(3.10), which is always feasible, a processing plan $\hat{p}(k) = \lfloor \hat{x}_k / m_k \rfloor$ and an idleness plan $\hat{u} = \hat{l}\mathbf{1} - C\hat{x}$ are formed. Let $N = \lceil l \rceil$, where l was defined in (3.5). The controller then, "attempts" to sequentially execute the plan $(\hat{p}, \hat{u})$ N times. The wording used is indicative of the fact that this processing plan cannot be implemented from jobs that are all present at the review point, and as a result a more careful execution methodology should be employed. As N gets large, intuition suggests that the ending state after this execution cycle will be close to the state predicted using the fluid approximation, which is above the required threshold requirements. The details of this *infeasible planning logic* are provided in section 4.1 and in the Appendix.

Hereafter, the notation $\mathbf{DR}(r, l, \beta)$ will denote the discrete-review policy derived from the functions $r(\cdot)$, $l(\cdot)$, and the vector β that satisfy (3.1)-(3.2), (3.3), and (3.4). In the sequel, we will use a subscript to differentiate between different review periods. An algorithmic description of a discrete-review policy is shown in Figure 1. (For clarity, the infeasible logic has been suppressed.)

Figure 1: Algorithmic description of $\mathbf{DR}(r, l, \beta)$

For a multiclass network under any policy in the proposed family the underlying continuous time Markov chain is defined as follows. Assume that $t_j \leq t < t_{j+1}$ and define the parameter $N(t)$ to be equal to 1 if the linear program (3.6)-(3.7) is feasible or otherwise set it equal to the number of remaining executions of the processing plan $\hat{p}$ derived from (3.9)-(3.10). Let $p(t)$ be a K-vector, where $p_k(t)$ is the number of class k jobs that remain to be processed at time t according to the processing plan p_j or $\hat{p}_j$, depending on whether the planning LP was feasible. Let $u(t)$ be the S-vector of remaining idling times for each of the servers for the ensuing planning period. Finally, let $R_a(t)$ be the $|\mathcal{E}|$-vector and $R_s(t)$ be the K-vector associated with the residual arrival and service time information. The Markovian state descriptor will then be

$$Y(t) = [Q(t);\ N(t);\ p(t);\ u(t);\ R_a(t);\ R_s(t);\ |Q(0)|], \tag{3.11}$$

and $\mathbf{Y}$ will represent the underlying state space. Imitating Dai's argument [7] and using the strong Markov property for piecewise deterministic processes of Davis [13], it is easy to show that the process $\{Y(t),\ t \geq 0\}$ is a strong Markov process with state space $\mathbf{Y}$. The associated norm will be

$$\|Y(t)\| = |Q(t)| + N(t) + |p(t)| + |u(t)| + |R_s(t)| + |R_a(t)|.$$

Before we proceed with a statement of the main results of this paper, we list a few remarks regarding the family of discrete-review policies under investigation. First, the actual execution time of the plan (p, u), denoted T^{exe}, is in general different than the nominal duration of l time units, and therefore, a distinction needs to be made between the nominal and actual cumulative allocation processes.

Second, the scheduling complexity for the proposed policies during each period is low. This is due to the fact that the execution of a discrete-review policy is insensitive to the precise processing sequence followed and thus, the overall complexity is that of a linear program of size equal to K, the number of classes in the network; this scales very gracefully with the size of the network. That is, the computational effort required in each planning phase is constant as a function of the review period length, the load in the network, and the amount of work to be scheduled. This is an important feature, for if the scheduling complexity had a superlinear growth rate as a function of $|q|$, then the associated computational delay would become significant relative to the time allocated to processing jobs, which would degrade performance and could affect the stability of the controlled network; this issue was addressed for a related class of policies by Bambos and Walrand [1]. Within a discrete-review setting, scheduling complexity could become significant in the more aggressive scenario where there is no implicit requirement that processing plans should be fulfilled from work present at the beginning of each review period. In this case, the sequencing of jobs within each period would be crucial in the execution of the policy and this issue will significantly increase the complexity of the planning phase. Such an example is discussed in [29, Appendix 2]. There, a variant of the discrete-review policies discussed here is considered, where safety stock levels are no longer enforced at the expense of a more complicated execution methodology and the requirement for longer planning horizons; these should be at least of order $\log^2(|Q(t)|)$, which is a more stringent condition than (3.1). That is, by increasing the length of the planning horizons the controller has enough flexibility to implement any processing plan by cleverly shifting work around the various stations, without incurring significant amounts of unplanned idleness. Nevertheless, the implementation methodology suggested for this "leaner" discrete-review policy is significantly more complex and is likely to perform poorly (in a non-asymptotic sense) unless some thresholds -of very moderate size- are still enforced.

The choice of the function $l(\cdot)$ to be an increasing function of the size of the state with sublinear growth is intuitive: as the size of the state increases and review periods lengthen, the approximation to system dynamics embodied in (3.7) becomes more accurate, while relative to the overall system evolution it appears as if we are reviewing system status at an ever increasing rate. That is, there are three relevant time scales: the first, is the natural time scale of the system that is proportional to the total backlog at any point in time, which is equal to $|q|$; the second, is the time scale within which the discrete-review structure is being executed that is proportional to $l(|q|)$; and the third, is the time scale in which individual events occur in the network, which is of the order of magnitude of mean service or interarrival times. Condition (3.1) ensures that the time scale of the discrete-review structure is increasing as a function of $|q|$ and together with (3.2) ensures the separation of the three time scales when $|q|$ is large. The constant l_0 is a lower bound on the planning horizon of the appropriate magnitude, which only becomes relevant if $|q|$ becomes very small. Some further comments on (3.1)-(3.2) can be found in Harrison [18], Tassiulas and Papavassiliou [37], and Gans and Van Ryzin [16]. The conditions in [18] are almost identical to ours, whereas the authors in [37] state only the sublinearity condition (3.2). In [16] the corresponding condition is of the form $l \sim (1-\rho)^{-a}$, where $a \in (0,1)$. Note that $(1-\rho)^{-1}$ is of the same order of magnitude as the average backlog in the system and thus their condition is more restrictive than (3.1).

In the case where some of the job classes have zero effective arrival rates, the planning and infeasible LPs need to be appropriately modified. This was first recognized by Sethuraman and Berstimas [33]. Specifically, the safety stock requirement for any class k such that $\alpha_k = 0$, is set to be $\theta_k = \min(l(|Q(t)|)\beta_k,\ Q_k(t))$, which implies that when $Q_k(t)$ becomes sufficiently small, it effectively drops out from the planning and execution phases associated with both LPs.

Finally, an example of an alternative implementation logic would be to allocate to each job class k the corresponding planned server usage x_k over the ensuing period, without translating it to number of class k jobs. Provided that preemptive-resume type of service is allowed, the main properties established in this paper would still be valid subject to minor changes in their derivation.

3.2. The stability theorem

The main result of this paper is establishing the stability for the family of discrete-review policies described above. Stability for a multiclass network is defined as follows:

Definition 3.1. A multiclass network under a specific scheduling policy is stable if the underlying Markov chain is positive Harris recurrent.

The main theorem to be proved is the following:

Theorem 3.1. Let $l : \mathbf{R}_+ \to \mathbf{R}_+$ be a strictly positive, concave function that satisfies (3.1)-(3.2), and $r : \mathbf{R}_+^K \to \mathbf{R}_+^K$ be a continuous function that satisfies (3.3), and $\beta \in \mathbf{R}_+^K$ be a vector that satisfies (3.4). Then a multiclass network is stable under the discrete-review policy $\mathbf{DR}(r, l, \beta)$.

The proof of Theorem 3.1, is divided in two major steps: (1) the derivation of the fluid model associated with $\mathbf{DR}(r, l, \beta)$ (this is Theorem 4.1); and (2) the stability analysis of the fluid model which will be used in order to establish the stability of the underlying stochastic network.

From Theorem 3.1 it follows that any multiclass network under a policy in the family considered here will be stable provided that there is enough processing capacity at each station. This is one of the first families of policies to be proved stable for this general class of networks and its stability verifies the conjecture that for multiclass networks with Markovian routing there always exists a stabilizing policy. This remark generalizes similar results for acyclic, feedforward, or re-entrant line networks which can be found in the work of Perkins et al. [31], and Kumar [22]. This observation although anticipated it is not entirely obvious. For example, although the *Last-Buffer-First-Serve* (*LBFS*) policy is known to be stable for re-entrant lines for $\rho < \mathbf{1}$, it leads to instability in the network considered by Rybko and Stolyar [32], which is a minor modification of a re-entrant line.

4. The fluid model associated with $\mathbf{DR}(r, l, \beta)$

As it will be shown in section 5, it is sufficient for stability analysis to restrict attention to sequences $\{y^n\}$ in $\mathbf{Y}$ such that $\|y^n\| \to \infty$ as $n \to \infty$, where every converging subsequence will yield undelayed limits, that is, $\bar{R}_a(0) = \bar{R}_s(0) = 0$. The main theorem proved in this section is the following:

Theorem 4.1. Consider a multiclass open queueing network under the discrete-review policy $\mathbf{DR}(r, l, \beta)$. For almost all sample paths ω and any sequence of initial conditions $\{y^n\} \subset \mathbf{Y}$ such that $\|y^n\| \to \infty$ as $n \to \infty$, there is subsequence $\{y^{n_j}(\omega)\}$ with $\|y^{n_j}(\omega)\| \to \infty$ such that

$$\bar{Q}^{n_j}(0, \omega) \to q(0, \omega) \tag{4.1}$$

$$(\bar{Q}^{n_j}(\cdot, \omega), \bar{T}^{n_j}(\cdot, \omega)) \to (q(\cdot, \omega), \bar{T}(\cdot, \omega)) \text{ u.o.c.,} \tag{4.2}$$

$$(\bar{N}^{n_j}(\cdot, \omega), \bar{p}^{n_j}(\cdot, \omega), \bar{u}^{n_j}(\cdot, \omega)) \to (0, 0, 0) \text{ u.o.c.,} \tag{4.3}$$

and the cumulative allocation process can be expressed in the form

$$\bar{T}(t, \omega) = \int_0^t v(\tau, \omega) d\tau \quad \text{for } t \geq 0. \tag{4.4}$$

The pair (q, v) satisfies equations (2.6)-(2.7) and the policy specific equation

$$v(t) \;\in\; \operatorname*{argmax}_{v \in \mathcal{V}(q(t))} \; r(q(t))' v, \tag{4.5}$$

where $\mathcal{V}(q(t)) = \{v : v \geq 0,\ Cv \leq \mathbf{1},\ (Rv)_k \leq \lambda_k \text{ for all } k \text{ such that } q_k(t) = 0\}$.

Equations (2.6)-(2.7) and (4.5) are the conditions that describe the desired behavior we are (a) trying to mimic through the discrete-review structure, and (b) trying to achieve asymptotically under fluid scaling. The proof of the theorem relies on a series of propositions that establish condition (4.5). First, several large deviations bounds are derived that essentially describe the asymptotic behavior of these policies under the proposed discrete-review structure. In specific, we prove that asymptotically the conditions of Lemma 4.1 are satisfied for every review period with probability one, which implies that asymptotically the LP in (3.6)-(3.7) is feasible w.p. 1. Second, the difference between the nominal and actual allocation processes is bounded using the FSLLN, and finally, the fluid equations for the nominal allocation process are derived. Without loss of generality, in the sequel we work directly with the converging subsequence, thus avoiding the use of double subscripts, and we assume that $\|y^n\| = n$ for all $n \geq 1$.

4.1. Preliminaries: Large deviations bounds

The following basic result from large deviations theory, often referred to as Chernoff bound, is a direct application of Markov's inequality, and it be used repeatedly in our analysis. It can be found, along with an illustrative exposure of the subject of large deviations, in several books such as Shwartz and Weiss [34, Theorem 1.5], Dembo and Zeitouni [14], or in the introductory paper by Weiss [39].

Fact 4.1. For a sequence $\{x_i\}$ of IID random variables such that $\mathbf{E}(e^{\theta x_1}) < \infty$ for some $\theta > 0$, for every $a > \mathbf{E}(x_1)$ and any positive integer n,

$$\mathbf{P}\left(x_1 + \cdots + x_n \geq na\right) \leq e^{-nf(a)},$$

where $f(a) = \sup_\theta(\theta a - \log(\mathbf{E}(e^{\theta x_1})))$ is a convex function and $f(a) > 0$.

An elementary analysis of the behavior under the proposed discrete-review structure is required in order to derive the fluid models associated with these policies. Figure 2 depicts a possible trajectory of one of the queue length variables over a single review period, where various quantities of interest are defined. Hereafter, the planned ending state upon completion of the j^{th} review period will be denoted by $z_{j+1} = q_j + l_j\lambda - Rx_j$, where q_j is the queue length vector observed at time t_j.

Figure 2: Discrete-review policy: a schematic representation

Lemma 4.1 proves that if q_j is above the level $(1-\Delta)\theta_j$, where $0 < \Delta < 1$ is a specified constant, then the linear program (3.6)-(3.7) is feasible and also the nominal processing plan for the j^{th} planning period can be implemented from jobs present in the system upon the review point. In Figure 2, this requirement is shown to be satisfied. The difference between the predicted ending state, z_{j+1}, and the actual state observed at the next review point, q_{j+1}, is denoted b_1. The difference between q_{j+1} and the lower threshold requirement mentioned above is b_2, and the time difference between the expected and actual duration of the review period is b_3. Large deviations bounds for the quantities b_1, b_2, b_3 will be obtained in Lemma 4.3. Similar bounds for the case where the linear program (3.6)-(3.7) is not feasible and the infeasible planning logic of equations (3.9)-(3.10) is used, are derived in Lemma 4.2. The focus of this large deviations analysis is on the behavior of the proposed policies when the state is large, or, equivalently, when the planning horizons get long.

The first lemma we prove states an algebraic result regarding the feasibility of the linear program (3.6)-(3.7).

Lemma 4.1. Let

$$\delta_1 = \min_k \frac{(R^{-1}\lambda)_k}{(R^{-1}\beta)_k}, \quad \delta_2 = \min_k(\beta_k - \mu_k), \text{ and } \quad \Delta = \min(1, \delta_1, \delta_2).$$

If at a review point $q_j > (1-\Delta)\theta_j$, then the linear program (3.6)-(3.7) is feasible and also $p_j(k) < q_j(k)$.

Proof. Let $v = R^{-1}\lambda - \delta R^{-1}\beta$, for some $\delta > 0$. We now choose δ such that $v \geq 0$ and $Cv \leq \mathbf{1}$. The first constraint is equivalent to

$$R^{-1}\lambda - \delta R^{-1}\beta \geq 0 \;\Rightarrow\; \delta \leq \min_k \frac{(R^{-1}\lambda)_k}{(R^{-1}\beta)_k}.$$

The second constraint states that $\rho - \delta C R^{-1}\beta \leq \mathbf{1}$ which is true for all $\delta > 0$. Let $x_j = l_j v$. From the definition of v we have that $x_j \geq 0$, $Cx_j \leq l_j\mathbf{1}$, and also $z_{j+1} = q_j + l_j(\lambda - Rv) \geq q_j + l_j(\lambda - Rv) \geq q_j + \delta l_j \beta$. Hence, the linear program (3.6)-(3.7) is feasible. Moreover, if $q_j > (1-\delta_2)\theta_j$ then, $q_j > l_j\mu$ and $p_j(k) < q_j(k)$. Setting $\Delta = \min(1, \delta, \delta_2)$ completes the proof. □

The following two lemmas summarize the large deviation bounds required for the derivation of the fluid limit model associated with a discrete-review policy. Their proofs are given in the Appendix.

Lemma 4.2. Consider any review point where (3.6)-(3.7) is infeasible. Under the processing plan derived from the infeasible planning algorithm (3.9)-(3.10), for any $\epsilon > 0$ there exists a constant N_1, such that if $|q_j| > N_1$ and for some $h(\epsilon) > 0$,

$$\mathbf{P}\left(q_{j+1} \not\geq (1-\epsilon)\theta_{j+1}\right) \leq e^{-h(\epsilon)l_j}. \tag{4.6}$$

Furthermore, for some constant $L_\epsilon > 0$ independent of the state q_j and for some $d(L_\epsilon) > 0$ we have that

$$\mathbf{P}\left(t_{j+1} - t_j > L_\epsilon l_j\right) \leq e^{-d(L_\epsilon)l_j}. \tag{4.7}$$

The second lemma concerns the system behavior when the planning LP is feasible.

Lemma 4.3. If the planning linear program (3.6)-(3.7) is feasible, then for any $\epsilon > 0$ there exists a positive constant N_2, such that if $|q_j| > N_2$ and for some $f(\epsilon) > 0$,

$$\mathbf{P}\left(q_{j+1} \not\geq (1-\epsilon)\theta_{j+1}\right) \leq e^{-f(\epsilon)l_j}. \tag{4.8}$$

Setting $\epsilon = \Delta$ in Lemmas 4.2 and 4.3, one obtains bounds on the probability that the condition of Lemma 4.1 is satisfied upon the next review period. The exponential form of these bounds explains the specific choice of the functions $l(\cdot)$ and $\theta(\cdot)$, since for n sufficiently large we have that $l \gtrapprox \kappa \log(n)$, which in turn implies that the derived bounds will decay as $1/n^{\kappa}$; this is sufficient in order to establish asymptotic properties in an almost sure sense.

The derived bounds could be computed explicitly if the distributional characteristics of the arrival and service processes were specified. Moreover, using the derived expressions one could calculate other quantities of interest such as, for example, the minimum review period length that would ensure a certain bound on the probability that during any planning period the controlled network would exhibit "significantly" different behavior than nominally planned.

4.2. Derivation of the associated fluid models

Given the sequence of review points $t_0, t_1, \ldots$ and any time $t \geq 0$, let $j_{max} = \min\{j : t_j \geq nt\}$.

Proposition 4.1. Define the sequence of events $\{A_n\}$, where $A_n = \{\omega : \exists j \leq j_{max},$ such that $q_j^n < (1-\epsilon)\theta_j^n\}$. Then for any $\epsilon > 0$, $\mathbf{P}\left(\limsup_n A_n\right) = 0$.

Proof. Recall that $\|y^n\| = n$. Given the definition of l_0 and the growth condition (3.1), for any $\epsilon > 0$ and any constant $\kappa > 0$, there exists an $N(\epsilon, \kappa) > 0$ such that for any $n \geq N(\epsilon, \kappa)$ we have that

$$(h(\epsilon) \vee f(\epsilon)) l_0^n > \kappa \log(n).$$

Set $N(\epsilon, \kappa) = \max(N(\epsilon, \kappa), N_1, N_2)$. Using the bounds derived in Lemmas 4.2 and 4.3 one gets that

$$\begin{aligned}
\mathbf{P}(A_n) &= \mathbf{P}\left(q_1^n \not\geq (1-\epsilon)\theta_1^n\right) + \\
&\quad + \sum_{j=1}^{j_{max}} \mathbf{P}\left(q_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n,\ q_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq e^{-\kappa \log(n)} + \\
&\quad + \sum_{j=1}^{j_{max}} \mathbf{P}\left(q_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n \mid q_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \times \\
&\qquad\qquad \mathbf{P}\left(q_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq \frac{1}{n^{\kappa}} + \sum_{j=1}^{j_{max}} \mathbf{P}\left(q_{j+1}^n \not\geq (1-\epsilon)\theta_{j+1}^n \mid q_i^n \geq (1-\epsilon)\theta_i^n,\ i \leq j\right) \\
&\leq \frac{j_{max}}{n^{\kappa}}.
\end{aligned}$$

For $\kappa \geq 3$,

$$\sum_n \mathbf{P}(A_n) = \leq \sum_{n \leq N(\epsilon,\kappa)} \mathbf{P}(A_n) + \sum_{n > N(\epsilon,\kappa)} \frac{j_{max}}{n^\kappa}$$
$$\leq N(\epsilon,\kappa) + c \sum_{n > N(\epsilon,\kappa)} \frac{1}{n^{\kappa-1}}$$
$$< \infty.$$

The desired result follows from the Borel-Cantelli Lemma. □

Remark 4.1. Set $\epsilon = \Delta$ and evaluate this probability bound for $q_j^n \geq (1-\Delta)\theta_j^n$ for all $j \leq j_{max}$. From Proposition 4.1 it follows that asymptotically under fluid scaling the condition that $q_j^n \geq (1-\Delta)\theta_j^n$ of Lemma 4.1 is almost always satisfied and thus, the infeasible planning logic is almost never used.

For any sample path ω, let $\bar{X}^n(t,\omega)$ be the (scaled) nominal allocation process, which is equal to the sum of all planned allocation times over all review periods up to time t, along this specific sample path ω. Similarly, let $\bar{T}^n(t,\omega)$ be the (scaled) actual allocation process, which is equal to the sum of all actual time allocations observed during execution of the processing plans for all review periods up to time t.

Proposition 4.2. $|\bar{T}^n(t) - \bar{X}^n(t)| \to 0$ a.s., as $n \to \infty$.

Proof. For any fixed time $t \geq 0$, let $t_j \leq nt < t_{j+1}$. Given the planned allocation over the j^{th} review period x_j^n, and the corresponding processing plan p_j^n, let $\delta_j^n(k) = x_j^n(k) - p_j^n(k)m_k$. That is, $\delta_j^n(k)$ is the remaining time allocated in processing class k jobs not included in the corresponding processing plan p_j^n. From Proposition 4.1, we have that for large enough n and for a.e. ω, the condition of Lemma 4.1 is satisfied at every review period. Then,

$$|\bar{T}^n(t,\omega) - \bar{X}^n(t,\omega)| = \frac{1}{n}|T^n(nt,\omega) - X^n(nt,\omega)|$$
$$\leq \frac{1}{n} Ll(|q_0^n|) + \frac{1}{n}\sum_{j=1}^{j_{max}}\sum_{k=1}^{K}\left[\sum_{i=1}^{p_j^n(k)}(\eta_k(i) - m_k) + \delta_j^n(k)\right]$$
$$\leq \frac{1}{n} Ll(|q_0^n|) + \sum_{k=1}^{K}\left[\frac{1}{n}\sum_{i=1}^{p^n(k)}(\eta_k(i) - m_k) + \frac{1}{n}\sum_{j=1}^{j_{max}}\delta_j^n(k)\right],$$

where $p^n(k) = \sum_j p_j^n(k) \leq nt/m_k$. The asymptotic behavior of the first term is clear. The second term of the equation satisfies the strong law

$$\sum_{k=1}^{K} \left(\frac{1}{n} \sum_{i=1}^{p^n(k)} (\eta_k(i) - m_k) \right) \to 0 \quad \text{a.s..} \tag{4.9}$$

This is a consequence of the functional strong law of large numbers for the service time processes. It remains to show that $\sum_j \delta_j^n(k)/n \to 0$. From assumption (3.1) and the fact that $l_j^n \geq al(|Q^n(0)|)$, it follows that $l_j^n \to \infty$ as $n \to \infty$ for all $j \leq j_{max}$.

Since $\delta_j^n(k) \leq m_k$ for all $j \geq 1$ it follows that

$$\frac{1}{n} \sum_{j=1}^{j_{max}} \delta_j^n(k) \leq \frac{1}{n} \frac{nt \cdot m_k}{\min_j l_j^n} \to 0, \quad \text{a.s..} \tag{4.10}$$

Combining (4.9) and (4.10) it follows that $|\bar{T}^n(t) - \bar{X}^n(t)| \to 0$ a.s.. $\square$

Once again, fix again time at some $t \geq 0$. Choose j such that $t_j \leq nt < t_{j+1}$ and let x_j^n denote the nominal allocation over the j^{th} planning period, which is of length $l^n(|q_j^n|)$. From Theorem 4.1 in Dai [7] we have that for a.e. ω, $\bar{Q}^n(t,\omega) \to q(t,\omega)$. Furthermore, from the absolute continuity of $q(\cdot,\omega)$ and condition (3.2) it follows that $\bar{l}_j^n = l_j^n/n \to 0$ as $n \to \infty$. Let $\bar{x}^n(t,\omega) = x_j^n/l_j^n$ and observe that

$$\bar{X}^n(t,\omega) = \sum_j \bar{x}^n(t,\omega)\bar{l}_j^n \to \int_0^t v(\tau,\omega)d\tau,$$

for some $v(\cdot,\omega)$ not yet specified. That is, the Riemann sum converges to the appropriate integral as $n \to \infty$. On the same time, from the previous proposition and Theorem 4.1 in Dai [7] we know that the nominal allocation process $\bar{X}^n(t,\omega)$ converges to a limit, denoted $\bar{X}(t,\omega)$, which is absolutely continuous. It follows that

$$v(t,\omega) = \frac{d\bar{X}(t,\omega)}{dt}$$

almost everywhere on the real line and thus, it is sufficient to study the limit of $\bar{x}^n(t,\omega)$ along the sequence $\{y^n\}$ in order to establish the fluid limit of the nominal allocation process.

Proposition 4.3. For a.e. ω, $\bar{x}^n(t,\omega) \to v(t,\omega) \in \text{argmax}\{r(q(t,\omega))'v : v \in \mathcal{V}(q(t,\omega))\}$.

Proof. From Theorem 4.1 in [7] it follows that for a.e. ω and any sequence of initial conditions $\{y^n\} \subset \mathbf{Y}$ such that $\|y^n\| \to \infty$ as $n \to \infty$, there exists a converging subsequence $\{y^{n_j}(\omega)\}$ with $\|y^{n_j}(\omega)\| \to \infty$ along which the fluid limits of the scaled queue length and allocation processes exist. Furthermore, from Proposition 4.1 it follows that as $n \to \infty$ for a.e. ω the planning LP of (3.6)-(3.7) is feasible.

Pick one such ω satisfying Theorem 4.1 from Dai [7] and Proposition 4.1, fix time at t and choose j such that $t_j(\omega) \leq nt < t_{j+1}(\omega)$. (Recall that to simplify notation we are working directly with the converging subsequence.) We have that $q_j^n(\omega) = Q^n(t_j(\omega)) = Q^n(n(t-\epsilon_n),\omega)$, for some ϵ_n such that where $0 \leq \epsilon_n \leq \bar{l}_j^n(\omega)$. Note that $\epsilon_n \to 0$ as $n \to \infty$. Also, let $\tilde{q}_j^n(\omega) = q_j^n(\omega)/|Q^n(0)|$. In the sequel, the variables x^n, v are dummy variables for the planning linear programs corresponding to $\bar{x}^n(t,\omega)$ and $v(t,\omega)$. The planning linear program of equations (3.6)-(3.7) states that

$$\bar{x}^n(t,\omega) \in \ \text{argmax} \ \ r(\tilde{q}_j^n(\omega))' x^n \tag{4.11}$$

$$\text{subject to}\, x^n \geq 0, \quad Cx^n \leq \mathbf{1}, \tag{4.12}$$

$$q_j^n(\omega) + (\lambda - Rx^n) l_j^n(\omega) \geq \theta_j^n(\omega). \tag{4.13}$$

Using strong duality of linear programming the optimality condition of equations (4.11)-(4.13) can be rewritten in the following form

$$r(\tilde{q}_j^n(\omega))' x^n = \mathbf{1}'\nu^n + (b^n(\omega))'\zeta^n, \tag{4.14}$$

$$x^n \geq 0, \quad Cx^n \leq \mathbf{1}, \tag{4.15}$$

$$Rx^n l_j^n(\omega) \leq b^n(\omega), \tag{4.16}$$

$$\nu^n, \ \zeta^n \geq 0, \tag{4.17}$$

$$C'\nu^n + R' l_j^n(\omega)\zeta^n \geq r(\tilde{q}_j^n(\omega))', \tag{4.18}$$

$$(\mathbf{1} - Cx^n)'\nu^n = 0, \quad (b^n(\omega) - Rx^n l_j^n(\omega))'\zeta^n = 0, \tag{4.19}$$

where $b^n(\omega) = q_j^n(\omega) + \lambda l_j^n(\omega) - \theta_j^n(\omega)$ and $\nu^n, \ \zeta^n$ are the dual optimization variables.

The next step is to derive the limiting behavior of the above set of conditions as $n \to \infty$. Conditions (4.15) and (4.17) are clear. Equation (4.16) constrains the feasible allocation vectors in the *primal* planning problem of equations (3.6)-(3.7). Rewrite the constraint of (4.13) in the form

$$\lambda - Rx^n \geq \beta - \frac{Q^n(nt,\omega)}{l^n(|Q^n(n(t-\epsilon_n,\omega)|)}. \tag{4.20}$$

If $\lim_n \bar{Q}_k^n(t,\omega) > 0$, the limit in the right hand side of (4.20) is minus infinity, and thus the corresponding constraint for class k jobs is inactive. From Proposition 4.1, for any $\epsilon > 0$ and a.e. ω

$$\lim_n \frac{Q_k^n(nt,\omega)}{\beta_k l^n(|Q^n(n(t-\epsilon_n),\omega)|)} \geq 1 - \epsilon, \quad \text{for all } k = 1,\ldots,K.$$

This implies that the limit of the right hand side of (4.20) is bounded above by zero. Moreover, when $\lim_n \bar{Q}_k^n(t,\omega) = 0$, by Proposition 4.1 it follows that $q(t,\omega) \geq 0$ for all $t \geq 0$, and thus it must be that $(\lambda - Rx^n)_k \geq 0$. Hence, $(\lambda - Rx^n)_k \geq \max(0, \beta - \lim_n Q^n(nt,\omega) l^n(|Q^n(n(t-\epsilon_n),\omega)|)) = 0$, and thus condition (4.16) becomes

$$(Rv)_k \leq \lambda_k, \quad \text{for } k \in \{1,\ldots,K\} \text{ such that } q_k(t,\omega) = 0.$$

This condition can be expressed as a polytopic constraint of the form $R_{q(t,\omega)} v \leq b_{q(t,\omega)}$, for the appropriate state dependent matrix $R_{q(t,\omega)}$ and vector $b_{q(t,\omega)}$.

Next, the limit of $\tilde{q}_j^n(\omega)$ is derived. Conditions (3.1)-(3.2) imply that $l_j^n(\omega) = l^n(|q_j^n(\omega)|) \to \infty$ and $\bar{l}_j^n(\omega) \to 0$ as $n \to \infty$. Also, by the construction of $Y(t)$, the state descriptor of the underlying Markov chain, that for any review period there exist positive constants a_1, a_2 such that $N(t) < a_1 l(|Q(t)|)$ and $p(t) < a_2 l(|Q(t)|)\mathbf{1}$, and also $u(t) < l(|Q(t)|)\mathbf{1}$. Hence,

$$\bar{N}^n(t,\omega) \to 0, \; \bar{p}^n(t,\omega) \to 0, \text{ and } \bar{u}^n(t,\omega) \to 0 \;\text{ u.o.c..} \tag{4.21}$$

Furthermore, for $\bar{R}_a(0) = \bar{R}_s(0) = 0$, it follows that $\|y^n\|/|Q^n(0)| \to 1$ and thus, $\tilde{q}_j^n(\omega) = \tilde{Q}^n(t-\epsilon_n) \to q(t,\omega)$. Using this result, it is now simple to derive the fluid limits of the other conditions using the continuity property of $r(\cdot)$, the continuous mapping theorem, and the methodology in Dai [7]. That is, these conditions can be expressed as integral constraints, that is pathwise complementarity conditions, which can be treated using Lemma 2.4 of Dai and Williams [12], in conjunction with the pointwise limit derived above for equation (4.16). First, one would derive the limits for (4.19) and then use the derived results for (4.14) and (4.18). The resulting set of conditions will be

$$\begin{aligned}
&r(q(t,\omega))'v = \mathbf{1}'\nu + b'_{q(t,\omega)}\zeta,\\
&v \geq 0, \quad Cv \leq \mathbf{1},\\
&R_{q(t,\omega)} v \leq b_{q(t,\omega)},\\
&\nu, \; \zeta \geq 0,\\
&C'\nu + R'_{q(t,\omega)}\zeta \geq r(q(t,\omega))',\\
&(\mathbf{1} - Cv)'\nu = 0, \quad (b_{q(t,\omega)} - R_{q(t,\omega)} v)'\zeta = 0.
\end{aligned}$$

One can immediately observe that these are the optimality conditions of the following linear program

$$v(t,\omega) \in \operatorname{argmax}\{r(q(t,\omega))'v : v \in \mathcal{V}(q(t,\omega))\}. \tag{4.22}$$

By Theorem 4.1 in [7] and Proposition 4.1, this limiting argument is true for almost all sample paths and thus the desired result is established. □

Remark 4.2. The limit $v(t,\omega)$ need not be unique; the above result simply describes the policy specific equation that the limit of the fluid scaled nominal allocation process should satisfy.

Remark 4.3. The definition of $\mathbf{DR}(r,l,\beta)$ uses information about the magnitude of the initial condition. This is required in order to get the appropriate limits for equations (4.11) and (4.14), and such a dependence is not necessary in the cases of static, linear or piecewise linear reward rate functions.

We can now complete proof of main theorem of this section.

PROOF OF THEOREM 4.1. Existence of a converging subsequence and the convergence of the queue length and allocation processes follow from Theorem 4.1 in Dai [7]. Together with the fluid limits of $p(t)$, $N(t)$ and $u(t)$ given in equation (4.21) this completes the proof of (4.3) and (4.3).

The representation of equation (4.4) follows again from the Lipschitz continuity of $\bar{T}(t,\omega)$ and equations (2.6) and (2.7) are restatements of the basic fluid equations derived in Theorem 4.1 in Dai [7], with respect to the instantaneous allocation process $v(\cdot,\omega)$.

Using Propositions 4.3 and 4.2 we have that for a.e. ω for any time $t \geq 0$

$$|\bar{X}^n(t,\omega) - \bar{T}(t,\omega)| \to 0 \quad \text{and} \quad |\bar{T}^n(t,\omega) - \bar{X}^n(t,\omega)| \to 0, \tag{4.23}$$

where $\bar{T}(t,\omega) = \int_0^t v(\tau,\omega)d\tau$ and $v(t,\omega)$ satisfies condition (4.5). Pick any sample path ω such that (4.23), is satisfied. Then for any time $t \geq 0$

$$|\bar{T}^n(t,\omega) - \bar{T}(t,\omega)| \leq |\bar{T}^n(t,\omega) - \bar{X}^n(t,\omega)| + |\bar{X}^n(t,\omega) - \bar{T}(t,\omega)| \to 0.$$

Since (4.23) is true for almost all sample paths, it follows that

$$|\bar{T}^n(t) - \bar{T}(t)| \to 0 \quad \text{a.s..} \tag{4.24}$$

By the Lipschitz continuity of the allocation processes it follows that all of the above conditions are true for any s such that $0 \leq s \leq t$, which implies that the convergence is uniform on compact sets for all $t \geq 0$. This completes the proof. □

5. Proof of stability

Stability of discrete-review policies will be established by analyzing their associated fluid models. This connection between the fluid models and the underlying stochastic networks was first established by Dai [7].

Definition 5.1. The fluid model associated with a scheduling policy is *stable* if there exists a time $T > 0$ such that for any solution $q(\cdot)$ of the fluid model equations with $|q(0)| = 1$, $q(t) = 0$, for $t \geq T$.

The following theorem is Dai's stability result [7] that relates the stability properties of the fluid model to that of the underlying queueing network.

Theorem 5.1. A multiclass open queueing network is *stable* under a scheduling policy if the associated fluid model is stable.

Thus, the verification of stability of a multiclass network under a specified policy is reduced to the much simpler task of checking stability of a fluid model with piecewise linear dynamics. For the case of *non-idling* or *work-conserving* fluid limits, these are fluid solutions (q, v) that satisfy the condition

$$(\mathbf{1} - Cv(t))'Cq(t) = 0, \quad \text{for all } t \geq 0, \tag{5.1}$$

Chen further refined this Theorem in [6, Theorem 5.2] by establishing the following condition: the fluid model equations (2.2)-(2.5) and (5.1) are stable if and only if (2.6)-(2.7) and (5.1) are also stable. Although this condition reduces the task of checking stability to an analysis of the simpler undelayed fluid model equations, in general one needs to establish stability under a whole family of policies that satisfy the non-idling constraint in (5.1), which is very conservative. The following modification of Chen's condition is sufficient in order to establish stability by analyzing a single policy of interest.

Proposition 5.1. Consider any scheduling policy such that its associated fluid limit model satisfies condition (5.1). Suppose that for any $B > 0$ there exists a time T_B such that for any solution $q(\cdot)$ of the undelayed fluid equations with $|q(0)| \leq B$, $q(t) = 0$, for all $t \geq T_B$. Then, the delayed fluid model is also stable.

Proof. Consider any fluid solution of the (delayed) fluid model equations. By the non-idling property, it follows that there exists a time t_0 such that $\bar{T}(t_0) \geq \bar{R}_s$ and $t_0\mathbf{1} \geq \bar{R}_a$. At time t_0 we have that

$$q(t_0) = \tilde{z} + \lambda t_0 - R\bar{T}(t_0),$$

where $\tilde{z} = z - \mathbf{diag}\{\lambda\}\bar{R}_a + R\bar{R}_s$. Let $B = |q(t_0)|$. By assumption it follows that $q(t) = 0$ for any $t \geq t_0 + T_B$, which completes the proof. □

The apparently stronger condition of establishing stability starting for any compact set of initial conditions is required in order to avoid potential pathological cases of dynamic control policies whose behavior changes drastically when the state is large. (For example, consider the Rybko-Stolyar network [32] which is known to be unstable under LBFS for some "bad parameter" choice. In this "bad parameter" regime, if the control policy used was LBFS if $|q(t)| > 1$ and any other strict priority rule if $|q(t)| \leq 1$, the undelayed fluid model will be stable if $|q(0)| \leq 1$ and unstable otherwise.) The condition of Proposition 5.1 is automatically satisfied in cases where a certain similarity property of the fluid limits applies (see Stolyar [35, section 6] or Chen [6, section 2]), and it is also true in all cases where this proposition will be invoked in the sequel.

We start by establishing that the positivity restriction on the reward rate functions implies the following non-idling property for the fluid models associated with $\mathbf{DR}(r, l, \beta)$:

Proposition 5.2. The fluid solutions of the set of equations (2.6), (2.7) and (4.5) associated with a strictly positive reward function $r(\cdot)$ are non-idling in the sense of (5.1).

Proof. For any $q(t) \neq 0$, the instantaneous allocation process in the fluid model $v(t)$, will satisfy equation (4.5). Suppose that there exists a server i such that $(Cq(t))_i > 0$ and $(Cv(t))_i < 1$. Let $k \in C_i$ be any job class for which $q_k(t) > 0$. Define the following instantaneous control $\hat{v} = v(t) + e_k(1-(Cv(t))_i)$, where e_k is the k^{th} unit vector. Clearly, $\hat{v} > 0$ and $C\hat{v} = Cv(t) + Ce_k(1-(Cv(t))_i) \leq \mathbf{1}$. Furthermore, for any job class $j \neq k$, $\lambda_j - (R\hat{v})_j \geq \lambda_j - (Rv(t))_j \geq 0$, which implies that $\hat{v}$ is a feasible instantaneous allocation. Moreover,

$$r(q(t))'\hat{v} = r(q(t))'v(t) + r_k(q(t))(1-(Cv(t))_i) > r(q(t))'v(t),$$

which contradicts the optimality of $v(t)$. Hence, there does not exist any server i such that $(Cq(t))_i > 0$ and $(Cv(t))_i < 1$, and thus the fluid solutions satisfying equation (4.5) will be non-idling. $\square$

Therefore, reward rate functions that satisfy (3.3) can be interpreted as non-idling dynamic priority rules; the latter is with respect to their fluid behavior. The specific form of the growth restriction in (3.3) is required for the proof of stability of these models; it could be relaxed at the expense of a more complicated proof.

Static reward rate vectors: in the case of a constant reward vector, (4.5) simplifies to

$$v(t) \in \operatorname*{argmax}_{v \in \mathcal{V}(q(t))} r'v. \tag{5.2}$$

The intuition is that a constant reward vector induces a static priority rule according to the relative magnitudes of the various reward rates in the sense described above. That is, for any two classes $k, j \in C_i$ for some server i, $r_k > r_j$ implies that class k jobs are given higher priority than class j jobs and so forth.

Stability is proved by identifying a Lyapunov function with the appropriate negative drift. (This follows from Lyapunov's direct method described, for example, in the book by LaSalle and Lefschetz [26].)

Proposition 5.3. The fluid model described by equations (2.6)-(2.7) and (5.2) is stable.

Proof. Define the function $V(q(t)) = r'R^{-1}q(t)$. First, observe that by the definition of the matrix R, it follows that R^{-1} is componentwise non-negative. Then, since $r > 0$ it follows that $r'R^{-1} > 0$ and thus, first, $V(q(t)) > 0$ for all $q(t) \neq 0$ and second, $V(q) = 0$ only when $q = 0$. Using $V(\cdot)$ as a candidate Lyapunov function, it is sufficient to prove that for all $q(t) \neq 0$ and for some $\epsilon > 0$,

$$\frac{dV(q(t))}{dt} = \min_{v \in \mathcal{V}(q(t))} r'R^{-1}(\lambda - Rv) < -\epsilon. \tag{5.3}$$

Let $\mathcal{I}(q(t)) = \{k : q_k(t) = 0\}$. The condition of equation (5.3) should be checked over all possible feasibility sets of the form $\mathcal{V}(q(t)) = \{v : v \geq 0,\ Cv \leq \mathbf{1},\ (Rv)_k \leq \lambda_k \text{ for all } k \in \mathcal{I}(q(t))\}$. Observe that if $\mathcal{I}(q_1) \subseteq \mathcal{I}(q_2)$, then $\mathcal{V}(q_1) \supseteq \mathcal{V}(q_2)$. This implies the drift condition need to be checked only for the extreme (most constrained) cases defined by $\mathcal{I}_k = \{1, \ldots, K\}/\{k\}$, for all job classes k. These sets correspond to the cases where all but the k^{th} job classes are empty, and the associated feasibility sets will be denoted by $\mathcal{V}_k = \{v : v \geq 0,\ Cv \leq \mathbf{1},\ R^k v \leq \lambda^k\}$, for the appropriate $(K-1) \times K$ matrix R^k and $(K-1)$-vector λ^k.

In order to check the drift condition for each $\mathcal{V}_k$, consider the input $\hat{v}^k = R^{-1}\lambda + \delta_k e_k$, where $\delta_k = 1 - \rho_{s(k)} > 0$. Clearly, $\hat{v}^k \geq 0$ and also $C\hat{v}^k = CR^{-1}\lambda + \delta_k Ce_k \leq \mathbf{1}$. Finally,

$$\lambda - R\hat{v}^k = -\delta_k Re_k = \delta_k(Pe_k - e_k) \Rightarrow R^k\hat{v}^k \leq \lambda^k, \tag{5.4}$$

which establishes the feasibility of the instantaneous allocation $\hat{v}^k$. Furthermore, a simple calculation yields that $r'R^{-1}(\lambda - R\hat{v}^k) = -\delta_k r_k < 0$. It follows from (5.2) that

$$\frac{dV(q(t))}{dt} < r'R^{-1}(\lambda - R\hat{v}^k). \tag{5.5}$$

Hence, the linear Lyapunov function $V(q(t)) = r'R^{-1}q(t)$ satisfies the drift condition of equation (5.3) with $\epsilon = \min_k \delta_k r_k$. One can easily obtain a bound on the time required to empty the fluid model starting from any bounded initial condition and hence, establish stability by invoking Proposition 5.1. □

One last observation in the context of constant reward rate vectors is that (5.2) can be rewritten in the form

$$v(t) \in \operatorname*{argmin}_{v \in \mathcal{V}(q(t))} \frac{dV(q(t))}{dt}. \tag{5.6}$$

Equation (5.6) provides a greedy interpretation of the associated fluid model, in the sense that resource usage is allocated in order to instantaneously minimize a linear objective defined by the linear Lyapunov function $V(q(t))$. The interpretation of the constant vector r as a static priority rule together with equation (5.6) illustrate a relation between linear Lyapunov functions, static priorities, and constant reward rate vectors.

As an example of a specific choice of a constant reward rate function consider an optimal network control problem under linear holding cost criterion. In this case, a sensible starting point in choosing a control policy for these networks, is to try to enforce the priority ranking that at any point in time strives to maximize cost draining out of the system; this is a generalization of the celebrated "$c\mu$" rule (see [38] for a discussion of related results). Assuming that the linear holding costs are denoted by h_k for each job class k, this greedy policy strives to minimize $h'(\lambda - Rv)$ of the admissible controls $v \in \mathcal{V}(q(t))$. The corresponding static reward rate function is given by $r = R'h$ and the appropriate structure of this policy is precisely the greedy control of (5.2).

Dynamic reward rate functions: stability for the fluid model described by (2.6)-(2.7) and (4.5) cannot be established by extending the method used for the case of constant reward vectors. The reason is that the vector valued function $r(\cdot)$ need not correspond to the gradient of some well defined potential function and as a result, no obvious Lyapunov function can be associated with $r(\cdot)$ in order to establish the stability of the fluid model. However, given the non-idling property of the dynamic reward rate functions satisfying (3.3), the negative drift condition in (5.3) still carries through, and it will be exploited in establishing the desired result.

Proposition 5.4. The fluid model described by equations (2.6)-(2.7) and (4.5) is stable.

Proof. Let $g(q(t)) = R^{-T} r(q(t))$. Since, R^{-1} is componentwise non-negative it is clear that $g(q(t)) > 0$, for all $q(t) \neq 0$. Furthermore, without loss of generality we can assume that the following normalization condition is true: $1 \leq g(q) \leq b + |q|^\gamma$ for some constant $b > 0$. Define the functional

$$V(q(t)) = -\int_t^\infty e^{-\zeta(\tau - t)} g(q(\tau))' \dot{q}(\tau) d\tau, \tag{5.7}$$

where $\zeta > 0$ is a discount factor. The functional $V(\cdot)$ can be interpreted as an exponentially weighted energy function for the fluid model. The drift of $V(\cdot)$ is given by

$$\frac{dV(q(t))}{dt} = g(q(t))'\dot{q}(t). \tag{5.8}$$

Imitating the proof of Proposition 5.3, an upper bound on the drift of $V(\cdot)$ is first derived as follows

$$\min_{v \in \mathcal{V}_i} \; g(q(t))'(\lambda - Rv) \;\leq\; g(q(t))'[\lambda - R(R^{-1}\lambda + \delta_i e_i)] \;\leq\; -\,\delta_i. \tag{5.9}$$

A lower bound of the drift of (5.8) can also be computed using $G = -\min_k\{(\lambda - Rv)_k : v \geq 0,\ Cv \leq \mathbf{1}\}$ Then for any $q(t) \neq 0$ we have that

$$-GK(b + |q(t)|^\gamma) < \frac{dV(q(t))}{dt} < -\epsilon = -(1 - \max_i \rho_i). \tag{5.10}$$

Hence, $V(q(t)) > 0$ for all $q(t) \neq 0$, and $V(q(t)) = 0$ implies that $q(t) = 0$. Given (5.10) the following upper bound on $V(q(t))$ is obtained

$$\begin{aligned} V(q(t)) &= -\int_t^\infty e^{-\zeta(\tau - t)} g(q(\tau))'\dot{q}(\tau)d\tau \\ &\leq GK \int_0^\infty e^{-\zeta\tau}(b + |q(t+\tau)|^\gamma)d\tau \\ &\leq GK \int_0^\infty e^{-\zeta\tau}(|q(t)| + \kappa\tau)^\gamma d\tau + \frac{GK}{\zeta}, \end{aligned}$$

where the last inequality is derived using the Lipschitz continuity of the fluid limit of the queue length process and $\kappa > 0$ is the appropriate Lipschitz constant. As a result, for any initial condition such that $|q(0)| = 1$,

$$V(q(0)) \leq GK \int_0^\infty e^{-\zeta\tau}(1 + \kappa\tau)^\gamma d\tau + \frac{GK}{\zeta} \leq \hat{G}, \tag{5.11}$$

where $\hat{G}$ is a constant that depends on the parameters $\gamma,\ \kappa,\ \zeta$ and G. Using the upper bounds of equations (5.10) and (5.11), we get that the time required to empty the fluid model is bounded above by $\hat{T} = \hat{G}/\epsilon$. Stability follows from Proposition 5.1. □

Finally, stability of the corresponding discrete-review policy follows from Theorem 5.1 (due to Dai) and Proposition 5.1. This completes the proof of Theorem 3.1.

Roughly speaking, the results of this section state that greedily optimizing over any reward rate function that satisfies a non-idling constraint (with respect to their fluid model behavior in the sense explained in Proposition 5.2) within a discrete-review structure will result in a stable scheduling policy; this is true even for priority rules that are known to have unstable fluid limit models when they are directly implemented. This property of the proposed discrete-review structure can be viewed as a regulator, or a stabilization method, for dynamic (non-idling) priority rules. This property follows from the fact that the priority ranking is not strictly enforced, but the controller is allowed extra flexibility in serving lower priority classes in order to maintain feasibility of the nominal ending state after each planning period, and thus prevent future idleness. Inherently, the last remark describes how the linear programming planning structure and the safety stock requirements of discrete-review policies address the fundamental tradeoff between myopic cost minimization and long-term resource utilization. In contrast, in a multiclass network under a given priority rule, resources are allocated in order to minimize some associated holding cost without any consideration for future resource utilization, which can lead to instability.

The idea of a stabilization method is not novel; Kumar and Seidman [24] proposed a threshold heuristic that essentially controls the large state behavior of any multiclass network with deterministic routing and deterministic processing and setup times, and Humes [20] proposed the use of regulators, a variant of the *leaky buckets* idea from the area of communications systems and traffic shaping. In this setting, they proved that their mechanisms stabilizes almost any scheduling policy. The differences between these two techniques and the policies presented here are first, on the stochastic versus deterministic nature of the controlled network and second, on the behavior (or performance) of the underlying networks when they are many jobs in the system. That is, although the essential mechanics of discrete-review policies reduce to a period by period deterministic reasoning, the overall system dynamics are stochastic. It was the choice of the length of each planning period and the corresponding magnitude of safety stocks that allow for this separation of the relevant time scales and yield the deterministic nature of the planning problem in (3.6)-(3.7). On the same time, a discrete-review policy explicitly involves optimization with respect to a some desired specification, such as designated class priorities, or some other performance criterion of interest, while the mechanisms proposed in [24,20] do not incorporate such performance considerations.

The non-idling assumption has proved necessary in order to guarantee the global drift condition of equation (5.9). In the case of idling-allowing priority rules, stability cannot be established using the techniques used in the proof of Theorem 3.1, and additional requirements need to be imposed on the reward function in order to still guarantee a negative drift for some other candidate Lyapunov function for the associated fluid model.

6. Extensions

In this section we describe several extensions to the network models under investigation that can be readily incorporated within the proposed framework with only minor modifications both in the conceptual and implementation levels. This is possible because both fluid approximations and the proposed family of control policies are largely insensitive to many subtle modeling details that lie below their level of resolution. In contrast, within mainstream queueing network theory these extensions would normally venture into radically different domains of application and research, and would not be able to be treated in a unified framework.

6.1. Adding more control capability

Alternate routing capability arises either when a job completes service at a station and has a choice as to which buffer to join next, or upon an exogenous arrival of a job in the system that again has a choice between different buffers that it can join. We will assume that these external arrival streams or input processes can be turned off, or equivalently, that such jobs can be rejected upon arrival depending on whether such an action would be advantageous for the overall system performance. An incentive structure for accepting arriving jobs will be introduced shortly. In contrast, the models considered so far assumed that routing decisions were made in a Markovian fashion according to the transition matrix P; this corresponds to a a randomized routing policy that is *a priori* specified with no admission control capability.

In extending the model formulation of section 2, it is convenient to introduce the designation of a "type" in order to identify each exogenous arrival stream that now could be routed to (or be split into) different job classes; this follows the modeling approach of Harrison in [18]. Types of arrivals will be indexed by j and with slight abuse of notation, the set of exogenous arrival streams will still be denoted by $\mathcal{E}$. Furthermore, it is easiest to model these input streams as being "created" or "generated" by fictitious "input servers" associated with each type of arrival. In this framework, a service time of the j^{th} input server corresponds to an interarrival time of the type j input stream drawn from the IID sequence $\{\xi_j(n), n \geq 1\}$, defined in section 2. We denote by R_j^a the set of classes k where a type j job can be routed to upon its arrival, and by R_k^d the set of classes l where a class k job can be routed to upon its service completion. The superscripts "a" and "d" are mnemonic for arrivals and departures respectively. If $\lambda_j = 0$, that is, if there are no arrivals of type j, we set $R_j^a = \{j\}$; this is consistent with our treatment so far.

To simplify notation it will be assumed that routing decisions are made upon the beginning of service of a job by a server or creation of a job by a fictitious input server. (Note that this is not a restrictive assumption, at least in the context of our approach, and could easily be relaxed.) In this case, a class k (type j) job beginning service (creation) is already "tagged" with the destination class $l \in R_k^d$ ($l \in R_j^a$), where it will be routed upon completion of service. Let $T_{k,l}(t)$ be the cumulative time allocated up to time t in processing class k jobs that are routed into class l jobs upon their service completion ($l \in R_k^d$), $Y_{j,l}(t)$ be the cumulative time allocated up to time t in creating type j jobs routed into class l jobs ($l \in R_j^a$), and $E_{j,l}(t)$ be the cumulative number of type j jobs routed into class l jobs up to time t given the cumulative allocation processes $Y_{j,l}(t)$. Extending our earlier formulation, a control policy now takes the form of a pair of cumulative allocation processes $\{(Y(t), T(t)), t \geq 0\}$. For all classes k and any $t \geq 0$,

$$T_k(t) = \sum_{l \in R_k^d} T_{k,l}(t) \quad \text{and} \quad E_k(t) = \sum_{j:k \in R_j^a} E_{j,k}(t), \tag{6.1}$$

and furthermore for each $j \in \mathcal{E}$ we have that $\sum_{l \in R_j^a} Y_{j,l}(t) \leq t$, for all $t \geq 0$; the last inequality is a consequence of the input control actions up to time t.

The following notation will be useful. Let $y_{j,l}(t)$ denote the fractional effort of the fictitious input server j allocated in creating type j jobs that are routed to the class l buffer at time t, and $\nu_{k,l}(t)$ be the fractional effort of server $s(k)$ allocated in processing class k jobs that are routed to the class l buffer at time t. The notation $v_k(t)$ still denotes the fraction of effort of server $s(k)$ devoted to processing class k jobs at time t.

An incentive structure for accepting externally arriving jobs is introduced in the form of a reward rate function $r^y : \mathbf{R}_+^K \to \mathbf{R}^{(\sum_{j \in \mathcal{E}} |R_j^a|)}$, that assigns a reward rate $r_{j,l}^y(q)$ to the activity of creating (or accepting) type j jobs that are routed to the class l buffer ($l \in R_j^a$) when the state of the system is equal to q. The resulting instantaneous reward will thus be $r_{j,l}^y y_{j,l}$. Similarly, the reward rate function $r^\nu : \mathbf{R}_+^K \to \mathbf{R}^{(\sum_k |R_k^d|)}$, assigns a reward rate $r_{k,l}^\nu(q)$ into processing class k jobs to be routed upon service completion to the class l buffer ($l \in R_k^d$) when the state of the system is q.

For appropriate choices of control vectors y, ν and matrices $\tilde{A}, \tilde{F}, \tilde{R}, \tilde{C}$, a discrete-review policy can be defined by replacing (3.6)-(3.7) by the following planning LP at each review period

$$\begin{aligned} (y, \nu) \;\in\; & \operatorname{argmax} \;\; (r^y)'y + (r^\nu)'\nu \\ & \text{subject to}\, y \geq 0,\; \nu \geq 0,\; \tilde{C}\nu \leq \mathbf{1}l,\; \tilde{A}y \leq l\mathbf{1},\; q + \tilde{F}y - \tilde{R}\nu \geq \theta. \end{aligned}$$

The execution of the processing plan derived by this LP is as follows:

- for each type j accept $\lceil (\tilde{A}y)_j \lambda_j \rceil$ jobs and the turn arrival stream off

- route $\lfloor y_{j,l}\lambda_j \rfloor$ type j jobs to class l, for each $j \in \mathcal{E}$ and $l \in R_j^a$
- process $\left(\sum_{l \in R_k^d} \lfloor \nu_{k,l}/m_k \rfloor\right) \wedge q_k$ class k jobs, for $k = 1, \ldots, K$
- route $\lfloor \nu_{k,l}/m_k \rfloor$ class k jobs to class l, for $k = 1, \ldots, K$ and $l \in R_k^d$
- implement remaining of idleness budget $u = l\mathbf{1} - \tilde{C}\nu$ as in section 3

The alternative logic of (3.9)-(3.10) and Lemma 4.2, that is applied if the planning LP above is infeasible can be extended in a similar way.

Using the standard procedure mentioned in section 2 and the derivations of section 4, the fluid limit model associated with this discrete-review policy will be

$$\dot{q}(t) = \tilde{F}y(t) - \tilde{R}\nu(t), \quad q(0) = z, \tag{6.2}$$

$$y(t) \geq 0, \quad \nu(t) \geq 0, \quad q(t) \geq 0, \tag{6.3}$$

$$\tilde{A}y(t) \leq \mathbf{1}, \quad \tilde{C}\nu(t) \leq \mathbf{1}, \tag{6.4}$$

together with the policy specific equation

$$(y(t), \nu(t)) \quad \in \quad \text{argmax} \quad r^y(q(t))'y + r^\nu(q(t))'\nu \tag{6.5}$$

The issue of stability is slightly more delicate now. Assuming that there is no input control (this is the natural case to consider), one needs to first define the appropriate notion of nominal load (or traffic intensity) for networks with alternate routing capability. (Without input control the constraint $\tilde{A}y(t) \leq \mathbf{1}$ needs to be replaced by $\tilde{A}y(t) = \mathbf{1}$.) As we have seen earlier, for a system to be stable it is necessary that the nominal load at each station is smaller than its processing capacity. Though, calculating the nominal utilization level at each station is no longer straight forward, since this depends on the routing strategy employed. The following linear program, adapted from Harrison [19], computes a worse case bound on the traffic intensity in the network:

$$\begin{aligned} &\text{minimize} \ \max_{1 \leq i \leq S} \rho_i \\ &\text{subject to} \ \tilde{A}y = \mathbf{1}, \ \tilde{R}\nu = \tilde{F}y, \ \tilde{C}\nu \leq \rho, \ \nu \geq 0, \ y \geq 0. \end{aligned}$$

The pair (y, ν) describes the average rates at which jobs are processed, created, and routed through the network, and ρ_i is an upper bound on the nominal utilization level at station i. A necessary condition for fluid model stability (according to Definition 5.1) is that $\rho < \mathbf{1}$. Provided that the solution of this LP satisfies this condition, all previous results regarding the stability of fluid models associated with discrete-review policies can be extended in a straight forward manner to the case of networks with routing control capability. It remains to establish the stability of the underlying stochastic networks, for which one first needs to extend the stability theory developed by Dai in [7] to this broader class of networks. In particular, a result equivalent to that of Theorem 4.1 in [7] is needed. Although, given the original work of Dai [7], such an extension would be mostly routine manipulation, it would be quite lengthy and is omitted. Assuming, nevertheless, the validity of this result, the stability of the discrete-review policies under investigation would easily be inferred.

6.2. Stations with setup delays

A practical feature of stochastic processing networks that is commonly omitted from mathematical models of these systems is that of setup delays (or switchover delays) that are incurred when a server switches between processing different classes of jobs. There is an extensive literature for these problems mostly under the rubric of polling systems that focuses on single server systems and often restricts attention to simple classes of policies, most often some variant of *round-robin* or *serve-to-exhaustion* policies; see for example [36,15,21].

In the context of discrete-review policies setup delays can be introduced at no extra cost in complexity or performance. This follows by exploiting the fact that planning horizons vary in a longer time-scale than that of setup delays incurred in switching between classes, and thus, the cumulative time spent in setups will be small compared to the cumulative time spent in the actual processing of jobs.

Specifically, let $d_{k,l}$ be the setup time required for server i to switch from processing class k jobs to processing class l jobs, where $k, l \in C_i$, and set $d_i = \max_{k,l \in C_i} d_{k,l}$. Then, the maximum time spent by server i in setups within every review period is bounded above by $|C_i| d_i$, since only one setup is needed per job class in executing the open-loop processing plans from jobs present at the beginning of each review period. This is constant as a function of $|Q(t)|$, and thus, as $|Q(t)|$ increases, the cumulative time spent in setups becomes negligible in comparison to the cumulative time spent in processing jobs at each station, and asymptotically under fluid scaling it vanishes. Using this last observation it should not be surprising that all the analysis and results provided so far will extend to the case of multiclass networks with setups.

7. Concluding remarks

In this paper we have described a new family of discrete-review policies for dynamic control of stochastic processing networks. These policies are characterized by the use of a dynamic reward rate function, of planning horizons that grow as a function of the magnitude of the queue length vector, and class level safety stocks. Under a continuity and a non-idleness assumption on the reward function and a growth condition on the review period length, every discrete-review policy in this family is proved to be stable. The loose assumptions imposed on the choice of the reward function provide significant design flexibility, which can be exploited in order to address a variety of applications with different performance metrics.

Several directions of research could be pursued. On the practical front, one would need to carry through extensive simulation experiments to validate the usefulness and applicability of these policies to practical situations. From a theoretical viewpoint, one would like to relate these ideas with optimal control of fluid and Brownian models. The first of the two has been addressed in part by the author in a companion paper [28], whereas the second is still an open problem.

Acknowledgements

I would like to thank Michael Harrison for providing the initial motivation for this work and for his continuing support along the various stages of this research. I am also grateful to Sunil Kumar for his continuing involvement with this work and his many insightful comments and suggestions. Finally, I am thankful to the anonymous referee for pointing out numerous helpful suggestions and changes.

Appendix

In the following two lemmas several large deviation bounds will be derived and thus, many auxiliary functions will have to be introduced (as exponents to these bounds). The convention we follow is that subscripts will designate class specific quantities whereas superscripts will designate different types of flows, such as arrivals or service completions.

PROOF – LEMMA 4.2. Suppose that at the j^{th} review period, the observed queue length vector is q_j, the nominal planning horizon length is l_j and (3.6)-(3.7) are infeasible. The controller proceeds with the infeasible planning logic of (3.9)-(3.10).

Since $\hat{l}$ is a slack parameter, the linear program (3.9)-(3.10) will always be feasible and $(\hat{x}, \hat{l})$ will be its corresponding minimizer. Define $\hat{p}(k) = \lfloor \hat{x}(k)/m_k \rfloor$ and $\hat{u} = \hat{l}\mathbf{1} - C\hat{x}$, where $\hat{u}$ is the vector of nominal idling times up to time $\hat{l}$. From equation (3.10) it follows that $\hat{l}\lambda - (I - P')\hat{p} > \beta$. Using the methodology of Lemma 4.1 the following bound on the review period length can be derived

$$\hat{l} \leq \frac{1}{\delta_1}\left(1 + \frac{1}{\min_k \beta_k}\right) \quad \text{and} \quad \hat{p}(k) \leq \frac{\hat{l}}{m_k}, \tag{.1}$$

where δ_1 is the constant defined in Lemma 4.1.

Let $N = \lceil l_j \rceil$ and define $p_j = N \cdot \hat{p}$ and $u_j = N \cdot \hat{u}$ to be the nominal processing plan and the nominal idling durations over the j^{th} planning period under this infeasible planning logic. To implement the proposed processing plan the following two step algorithm will be executed. First, the system will be idled for a sufficient period so as to accumulate all jobs associated with classes that have an exogenous arrival process, that are required to be processed according to the plan p_j. Second, the plan (p_j, u_j) will be divided and implemented as a sequence of N independent executions of $(\hat{p}, \hat{u})$.

Step 1: Let $\tau_I = \max\{p_j(k)/\lambda_k : \text{for all } k \in \mathcal{E}\}$. Using (.1) it is simple to see that

$$\tau_I \le \max_{k\in\mathcal{E}} \frac{\hat{l}\cdot N}{\lambda_k \cdot m_k} \le L_I \cdot l_j$$

for the appropriate constant L_I. As mentioned above, the first step is to idle the system for $(\tau_I + \epsilon l_j/2)$ time units. The following bound is a direct application of Fact 4.1 on the sequence of random variables $\{\xi_k(n),\ n \ge 1\}$ for each class $k \in \mathcal{E}$

$$\mathbf{P}\left(E_k(\tau_I + \epsilon l_j/2) < p_j(k)\right) \le e^{-h_k^a(\epsilon) l_j}, \tag{.2}$$

for some $h_k^a(\epsilon) > 0$. Let $h^a(\epsilon) = \min\{h_k^a(\epsilon) : \text{for all } k \in \mathcal{E}\}$ to get the bound

$$\mathbf{P}\left(E_k(\tau_I + \epsilon l_j/2) < p_j(k)\right) \le e^{-h^a(\epsilon) l_j} \quad \text{for all } k \in \mathcal{E}. \tag{.3}$$

Step 2: The plan (p_j, u_j) will be divided in N independent executions of $(\hat{p}, \hat{u})$. Each *inner execution* iteration of $(\hat{p}, \hat{u})$ could be performed as follows:

1. idle the system for $\max_i \hat{u}_i$ time units;
2. serve $\hat{p}(k)$ jobs for each class $k \in \mathcal{E}$;
3. sequentially serve jobs of each class $k \notin \mathcal{E}$ in any order until either $\hat{p}$ has been completed or there are no more jobs to serve for any class j for which the processing plan $\hat{p}(j)$ has not yet been fulfilled.

Let $\hat{y}(k) = \lfloor E_k(\tau_I + \epsilon l_j/2)/N \rfloor$ for each class $k \in \mathcal{E}$, and $\hat{y}(k) = 0$ otherwise. Let $\hat{w}^i$ be a random variable in $\mathbf{R}_+^K$ defined to be the ending state of the queueing network under study starting with initial condition $\hat{y}$, upon completion of the i^{th} *inner execution* step of $(\hat{p}, \hat{u})$ described above. That is, under this implementation mechanism all N of these inner execution steps are initialized at $\hat{y}$ and the system tries execute $(\hat{p}, \hat{u})$ as if the there were only $\hat{y}$ jobs in the queues at the beginning of the i^{th} step, independent of the ending state upon the completion of the previous $(i-1)$ cycles. Hence, $\hat{w}^1, \ldots, \hat{w}^N$ is a sequence of IID random variables. Since, each $\hat{w}^i$ can be expressed as a linear combination of service time and interarrival time random variables that satisfy **(A2)**, we have that $\mathbf{E}(E^{\theta \hat{w}^1}) < \infty$ for some $\theta > 0$. Let $T_{i.e.}$ be the execution time of a single iteration of this step. Clearly, $\mathbf{E}(T_{i.e.}) \ge \hat{l}$ and therefore $\mathbf{E}(\hat{w}^1) \ge \hat{l}\lambda - (I - P')\hat{p} > \beta$. Applying repeatedly Fact 4.1 one eventually gets the following bound

$$\mathbf{P}\left(\sum_{i=1}^{N} [\hat{w}^i - \mathbf{E}(\hat{w}^1)] > \frac{\epsilon}{2} N\beta\right) \le e^{-h^{i.e.}(\epsilon) N}, \tag{.4}$$

for some $h^{i.e.}(\epsilon) > 0$. For this execution plan, $q_{j+1} = q_j + \sum_{i=1}^N \hat{w}^i$. Hence, by combining the bounds in (.3) and (.4) and letting $h(\epsilon) = \min\{h^a(\epsilon),\ h^{i.e.}(\epsilon)\}$, the following bound is derived

$$\mathbf{P}\left(q_{j+1} - \theta_j \not\geq \frac{\epsilon}{2} l_j \beta\right) \le e^{-h(\epsilon) l_j}. \tag{.5}$$

Total Duration: The total duration of the proposed execution plan $t_{j+1} - t_j$ is given by $t_{j+1} - t_j = \tau_I + \epsilon l_j/2 + \sum_{i=1}^N T_{i.e.}(i)$. The sequence $T_{i.e.}(1), \ldots, T_{i.e.}(N)$ of IID random variables satisfies the following conditions:

$$\mathbf{E}(T_{i.e.}(1)) \ge \hat{l}, \quad \text{and} \quad \mathbf{E}(T_{i.e.}(1)) \le (1 + S)\hat{l}. \tag{.6}$$

Once again, it is easy to verify that $\mathbf{E}(E^{\theta T_{i.e.}(1)}) < \infty$ for some $\theta > 0$, and thus the following large deviations bound for $t_{j+1} - t_j$ can be derived using Fact 4.1 on the sequence $\{T_{i.e.}(i), i \leq N\}$ of IID random variables:

$$\mathbf{P}\left(t_{j+1} - t_j > L_\epsilon l_j\right) \leq e^{-d(L_\epsilon) l_j}, \tag{.7}$$

for some $L_\epsilon > L_I + \epsilon/2 + 2(1+S)\hat{l}$ and some $d(L_\epsilon) > 0$.

Feasibility: It remains to show that the state observed at the next review point q_{r+1} will satisfy the conditions of Lemma 4.1 with sufficiently high probability. Since, $|q_{j+1}| \leq |q_j| + |\sum_{i=1}^{N} \hat{w}^i|$ it follows that

$$l_{j+1} \leq l_j + l\left(|\sum_{i=1}^{N} \hat{w}^i|\right). \tag{.8}$$

Clearly, there exists a constant $a > 0$ such that $\beta < \mathbf{E}(\hat{w}^1) < a\beta$, and using (.4) the following bound is obtained

$$\mathbf{P}\left(|\sum_{i=1}^{N} \hat{w}^i| - Na|\beta| > \frac{\epsilon}{2} N|\beta|\right) \leq e^{-h^{i.e.}(\epsilon)N}.$$

Combining these two results with equation (.8) and for some constant $G > 0$ independent of q_j

$$\mathbf{P}\left(l_{j+1} - l_j > l(Gl_j)\right) \leq e^{-h(\epsilon) l_j}. \tag{.9}$$

Given (.5), in order to prove (4.6) it is sufficient to show that $(1-\epsilon/2)l_j > (1-\epsilon)l_{j+1}$. This is equivalent to the condition $l(Gl_j)/l_j < \epsilon/(2-2\epsilon)$, which using (3.2) can be rewritten in the form $l_j > N(\epsilon)$, where $N(\epsilon) > 0$ is some appropriate constant. The last condition is equivalent to $|q_j| > N_1 = e^{N(\epsilon)/c}$ and this completes the proof. $\square$

Proof – Lemma 4.3. Recall that $\{\phi^i(n)\}$ is the sequence of K-dimensional Bernoulli random vectors such that $\phi^k_j(n) = 1$ if upon service completion the n^{th} class k job becomes a class j job and is zero otherwise, and that $\Phi^k(n) = \sum_{j=1}^{n} \phi^k(j)$. Using these definitions the state observed at the end of the r^{th} planning period will be

$$q_{j+1} = q_j + E(t_{j+1} - t_j) - p_j + \sum_{k=1}^{K} \Phi^k(p_j(k)). \tag{.10}$$

The various terms in equation (.10) will be analyzed in order to obtain the desired bound for q_{j+1}.

External arrivals: The expected duration of the r^{th} execution period satisfies the condition

$$\mathbf{E}(t_{j+1} - t_j) = \|u_j + CMp_j\|_\infty \leq \|u_j + Cx_j\|_\infty \leq l_j. \tag{.11}$$

Using Wald's identity for random sums of random variables, one gets that the expected vector of external arrivals until the next review period t_{j+1} is such that $\mathbf{E}(E(t_{j+1} - t_j)) \leq l_j\lambda$. Let $S_j(k)$ denote the number of class k services completed in the first j review periods. Apply Fact 4.1 to the sequence $\{\eta_k(n),\ n \geq 1\}$ for each job class k to get that

$$\mathbf{P}\left(\sum_{i=S_j(k)+1}^{S_j(k)+p_j(k)} (\eta_k(i) - m_k) > \epsilon l_j\right) \leq \mathbf{P}\left(\sum_{i=1}^{\lceil l_j \cdot \mu_k \rceil} (\eta_k(i) - m_k) > \epsilon l_j\right) \leq e^{-f^s_k(\epsilon) l_j}. \tag{.12}$$

A similar argument will yield a lower bound on $\sum_{i=1}^{p_j(k)} (\eta_k(i) - m_k)$. (For this an alternative form of Fact 4.1 needs to be invoked that focuses on large exceedanses below the mean; such an extension is straight forward to obtain by considering the sequence of random variables $\{-x_1, \ldots, -x_n\}$ and any $a < \mathbf{E}(-x_1)$.) Overloading notation, let $f_k^s(\epsilon)$ be the exponent in the large deviations bound for $|\sum_{i=1}^{p_j(k)} (\eta_k(i) - m_k)|$. Combining the above results over all classes the following bound is obtained

$$\mathbf{P}(|t_{j+1} - t_j - l_j| > \epsilon l_j) \leq e^{-f^s(\epsilon) l_j}, \tag{.13}$$

where $f^s(\epsilon) = \min\{f_k^s(\epsilon) : \text{for all } k \text{ such that } p_j(k) > 0\} > 0$. Also, for any class $k \in \mathcal{E}$

$$\mathbf{P}(|E_k(t) - \lambda_k t| > \epsilon t \beta_k) \leq e^{-f_k^a(\epsilon_k) t}, \tag{.14}$$

where $\epsilon_k = \epsilon\beta_k$ and the last bound was obtained once again using Fact 4.1 in the sequence $\{\xi_i(n),\ n \geq 1\}$ of IID random variables for the events $\{\sum_{j=1}^{\lceil(\lambda_k+\epsilon)t\rceil} \xi_k(j) < t\}$ and $\{\sum_{j=1}^{\lfloor(\lambda_k-\epsilon)t\rfloor} \xi_k(j) > t\}$, and once again $f_k^a(\epsilon_k) > 0$. Let $f^a(\epsilon) = \min\{f_k^a(\epsilon_k) : \text{for all } k \in \mathcal{E}\}$. Then by combining (.13) and (.14) the following bound is derived

$$\mathbf{P}(|E(t_{j+1} - t_j) - l_j\lambda| > \epsilon l_j \beta) \leq e^{-f^{ext}(\epsilon) l_j}, \tag{.15}$$

where $f^{ext}(\epsilon) = \min\{f^s(\epsilon),\ f^a(\epsilon)\} > 0$.

Internal flows: Using Fact 4.1 on the appropriate sequence of Bernoulli random variables the following bound is obtained

$$\mathbf{P}(\Phi_i^k(p_j(k)) - P_{k,i} p_j(k) > \epsilon l_j \beta_i) \leq e^{-h_{ki}^{int}(\epsilon) l_j}.$$

Letting $h^{int}(\epsilon) = \min\{h_{ki}^{int}(\epsilon) : 1 \leq k, i \leq K\} > 0$ the following bound is established

$$\mathbf{P}\left(\sum_{k=1}^{K} \Phi^k(p_j(k)) - P' p_j > \epsilon l_j \beta\right) \leq e^{-h^{int}(\epsilon) l_j}. \tag{.16}$$

Similarly, one could obtain a lower bound for this internal flow process and then by combining (.15) and (.16) and letting $f(\epsilon) = \min\{f^{ext}(\epsilon/6),\ h^{int}(\epsilon/6)\} > 0$ the following bound is derived

$$\mathbf{P}(z_{j+1} - q_{j+1} > \mathbf{1} + \frac{1}{3}\epsilon l_j \beta) \leq e^{-f(\epsilon) l_j}. \tag{.17}$$

To prove the second result of this lemma observe that for l_j large enough, which is equivalent to the condition $q_j > N$ for some constant $N > 0$, equation (.17) becomes

$$\mathbf{P}(q_{j+1} \not\geq (1 - \epsilon/2)\theta_j) \leq e^{-f(\epsilon) l_j} \tag{.18}$$

Letting $\bar{\delta} = \text{argmax}\{|\lambda - Rv| : v \geq 0,\ Cv \leq \mathbf{1}\}$, the following is true

$$|q_{j+1}| \leq |q_j| + l_j(|\bar{\delta}| + \epsilon/2)|\beta|). \tag{.19}$$

As for Lemma 4.2, an appropriate constant N_2 can be computed to complete the proof. □

References

[1] N. Bambos and J. Walrand. Scheduling and stability aspects of a general class of parallel processing systems. *Adv. Appl. Prob.*, 25:176–202, 1993.

[2] D. Bertsimas, D. Gamarnik, and J. N. Tsitsiklis. Stability conditions for multiclass fluid queueing networks. *IEEE Trans. Aut. Control*, 41(11):1618–1631, Nov. 1996.

[3] D. Bertsimas and G. Van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Oper. Res.*, 39(4):601–615, July 1991.

[4] M. Bramson. Instability of FIFO queueing networks. *Ann. Appl. Prob.*, 4(2):414–431, 1994.

[5] M. Bramson. Stability of two families of queueing networks and a discussion of fluid limits. *Queueing Systems*, 1998. To appear.

[6] H. Chen. Fluid approximations and stability of multiclass queueing networks: work-conserving policies. *Ann. Appl. Prob.*, 5:637–655, 1995.

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

[8] J. G. Dai. Stability of open multiclass queueing networks via fluid models. In F. Kelly and R. Williams, editors, *Stochastic Networks*, volume 71, pages 71–90. Proceedings of the IMA, 1995.

[9] J. G. Dai. A fluid limit model criterion for instability of multiclass queueing networks. *Ann. Appl. Prob.*, 6:751–757, 1996.

[10] J. G. Dai and S. Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Trans. Aut. Control*, 40(11):1889–1904, Nov. 1995.

[11] J. G. Dai and G. Weiss. Stability and instability of fluid models for certain re-entrant lines. *Math. Oper. Res.*, 21:115–134, 1996.

[12] J. G. Dai and R. J. Williams. Existence and uniqueness of semimartingale reflecting brownian motions in convex polyhedrons. *Theory of Probability and its Applications (in Russian)*, 40:3–53, 1994. To appear in the SIAM translation of the journal with the same name.

[13] M. H. A. Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *J. R. Statist. Soc. Ser. B*, 46(3):353–388, 1984.

[14] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Jones and Bartlett Publishers, 1993.

[15] A. Federgruen and Z. Katalan. Approximating queue size and workload distributions in general polling systems. *Queueing Systems*, 18:353–386, 1994.

[16] N. Gans and G. Van Ryzin. Optimal control of a multiclass, flexible queueing system. *Oper. Res.*, 45(5):677–693, Sept. 1997.

[17] J. M. Harrison. Brownian models of queueing networks with heterogeneous customer populations. In W. Fleming and P. L. Lions, editors, *Stochastic Differential Systems, Stochastic Control Theory and Applications*, volume 10 of *Proceedings of the IMA*, pages 147–186. Springer-Verlag, New York, 1988.

[18] J. M. Harrison. The BIGSTEP approach to flow management in stochastic processing networks. In F. Kelly, S. Zachary, and I. Ziedins, editors, *Stochastic Networks: Theory and Applications*, pages 57–90. Oxford University Press, 1996.

[19] J. M. Harrison. Brownian models of open processing networks: Canonical representation of workload. *Ann. Appl. Prob.*, 1996. Submitted.

[20] C. Humes Jr. A regulator stabilization technique: Kumar-Seidman revisited. *IEEE Trans. Aut. Control*, 39(1):191–196, Jan. 1994.

[21] E. G. Coffman Jr., A. A. Puhalskii, and M. I. Reiman. Polling systems in heavy traffic: A Bessel process limit. *Math. Oper. Res.*, 23(2):257–304, 1998.

[22] P. R. Kumar. Re-entrant lines. *Queueing Systems*, 13:87–110, 1993.

[23] P. R. Kumar and S. P. Meyn. Duality and linear programs for stability and performance analysis of queuing networks and scheduling policies. *IEEE Trans. Aut. Control*, 41(1):4–17, Jan. 1996.

[24] P. R. Kumar and T. I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. Aut. Control*, 35(3):289–298, Mar. 1990.

[25] S. Kumar and P. R. Kumar. Fluctuation smoothing policies are stable for stochastic re-entrant lines. *Discrete Event Dynamic Systems*, 6:361–370, 1996.

[26] J. LaSalle and S. Lefschetz. *Stability by Liapunov's Direct Method*. Academic Press, New York, 1961.

[27] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Aut. Control*, 36(12):1406–1416, Dec. 1991.

[28] C. Maglaras. Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality. *Ann. Appl. Prob.*, 1997. Submitted.

[29] C. Maglaras. *Dynamic control for stochastic processing networks: A fluid model approach*. PhD thesis, Stanford University, 1998.

[30] S. P. Meyn. Transience of multiclass queueing networks via fluid limit models. *Ann. Appl. Prob.*, 5:946–957, 1995.

[31] J. R. Perkins, C. Humes, Jr., and P. R. Kumar. Distributed scheduling of flexible manufacturing systems: Stability and performance. *IEEE Trans. Robotics and Automation*, 10(2):133–141, Apr. 1994.

[32] A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operations of open queueing networks. *Problems of Information Transmission*, 28:199–220, 1992.

[33] J. Sethuraman and D. Bertsimas. Personal communication, 1998.

[34] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis: Queues, communications and computing*. Chapman & Hall, 1995.

[35] A. L. Stolyar. On the stability of multiclass queueing networks: A relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, 1:491–512, 1995.

[36] H. Takagi. Queueing analysis of polling models: an update. In H. Takagi, editor, *Stochastic Analysis of Computer and Communication Systems*. North Holland, Amsterdam, 1990.

[37] L. Tassiulas and S. Papavassiliou. Optimal anticipative scheduling with asynchronous transmission opprtunities. *IEEE Trans. Aut. Control*, 40(12):2052–2062, Dec. 1995.

[38] J. A. Van Mieghem. Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *Ann. Appl. Prob.*, 5:809–833, 1995.

[39] A. Weiss. An introduction to large deviations for communication networks. *IEEE Journal on Selected Areas of Communications*, 13(6):938–952, Aug. 1995.

given $r(\cdot), l(\cdot), \beta$;

$j = 0$; $t_0 = 0$; $l_0 = al(|Q(0)|)$;

repeat {

1. Compute length of review period length and safety stock levels. Set:
   $q := Q(t_j);\ \tilde{q} := q/|Q(0)|;\ l := l_0 \vee l(|q|);\ r := r(\tilde{q});\ \theta := \beta l;$
2. Compute K-vector of nominal allocations x
   $$\begin{array}{ll} \text{maximize} & r'x \\ \text{subject to} & q + l\lambda - Rx \geq \theta, \quad x \geq 0, \quad Cx \leq l\mathbf{1} \end{array}$$
3. Form processing plan p and idleness budget u
   $$p(k) := \left\lfloor \frac{x_k}{m_k} \right\rfloor \wedge q_k \ \text{ for } k = 1, \ldots, K, \quad u_i := l - (Cx)_i \text{ for } i = 1, \ldots, S$$
4. Execute (p, u) until completion - execution time T^{exe}.
5. Update: $t_{j+1} := t_j + T^{exe}$; $\ j := j + 1$;

}

Figure 1. Algorithmic description of $\mathbf{DR}(r, l, \beta)$

Figure 2. Discrete-review policy: a schematic representation