Adaptive Back-Pressure Congestion Control Based on Local Information

Leandros Tassiulas, *Member, IEEE*

Abstract—**The problem of distributed congestion control as it arises in communication networks as well as in manufacturing systems is studied in this paper. In particular, a multistage queueing system that models virtual circuit and datagram communication networks and a class of manufacturing systems are considered. The topology may be arbitrary, there are multiple traffic classes, and the routing can be class dependent, with routes that may form direct or indirect loops. The model incorporates the functions of transmission scheduling, flow control, and routing, through which congestion control is performed in the network. A policy is given that performs these functions jointly. According to the policy, heavily loaded queues are given higher priority in service. A congested node may reduce the flow from upstream nodes through a flow control mechanism. Whenever routing is required, it is performed in such a manner that the lightly loaded queues receive most of the traffic. For arrival processes with bounded burstiness, the policy guarantees bounded backlogs in the network, as long as the load of each server is less than one. The actions of each server are based on the state of its own queues and of the queues one hop away. Therefore, they are implementable in a distributed fashion. An adaptive version of the policy is also provided which makes it independent of the arrival rates.**

I. Introduction

DYNAMIC control achieves a better utilization of the transmission and switching resources of a communication network over static schemes. Dynamic schemes have been adopted for packet routing, for sharing the transmission capacity of a link among several competing information streams, or for controlling the flow of traffic in a packet switching network. In large networks though, where the control functions are distributed, dynamic control policies are prone to misbehavior, and the network may enter intriguing instability modes. The same phenomenon has been observed in manufacturing systems with distributed scheduling [10], [6]. In this work, we present a new distributed dynamic control policy in a multistage queueing system that models virtual circuit and datagram communication networks, as well as a class of manufacturing systems. The policy, which we call the adaptive back-pressure (ABP) congestion control policy, schedules the servers, routes the served traffic, and controls the flow based on local information, and therefore is amenable to distributed implementation. Each server is allocated dynamically based on the state of the queues that it serves, as well as on the state of downstream queues, which are one hop away. When there are routing options, the decision is taken again in a similar distributed fashion. It is shown that under the ABP policy, the backlog in the network remains bounded as long as the utilization of each server is less than one.

A central problem in the design of virtual circuit (VC) communication networks is the sharing of the transmission capacity of a link among the virtual circuits that go through it. Several schemes have been proposed, including round robin, weighted round robin, golden ratio scheduling, virtual clock, and processor sharing [4], [9], [14], [8], [5]. In some of the above schemes [4], [5], the link transmission scheduling is combined with window flow control. One of the main problems arising in this context is how to evaluate the throughput of a virtual circuit network, that is, the session rates that can be sustained by the network for certain transmission control policies, without experiencing instabilities. Cruz [2], [3] has considered this problem in virtual circuit networks by employing several different link transmission disciplines including FIFO, LIFO, and strict priority. He obtained bounds for the backlog in the nodes of the network. These bounds depend on the session arrival rates and burstiness characteristics, and guarantee stable operation of the system for the traffic parameters for which they remain finite. In certain cases though, in networks with virtual circuits that may form cycles, these bounds explode for arrival rates which give utilization strictly less than one at all network links. In this case, no conclusion can be drawn for the stability of the network. Chang [1] extended these results, obtaining backlog bounds in multiclass networks with routing. The stability problem yet remained open for certain cases in networks with cycles. Yaron and Sidi [13] addressed the same question with processes satisfying constraints on the tails of the backlog distribution. Hahne [4] has shown that with round robin scheduling in each link and hop-by-hop window flow control, there exist window sizes to stabilize the network as long as the link utilization is less than one. The selection of the window sizes depends on the arrival rates. By the ABP policy, stabilization of the network is achieved as long as the utilization of every link is less than one, and no knowledge of the arrival rates is required for its implementation. When the routes of the packets are not prespecified but only their final destinations are given, as is the case with datagram communication networks, then the ABP policy combines routing with the scheduling mechanisms described in the VC case to preserve bounded queue lengths.

A problem of stability, similar to the one discussed above, arises in manufacturing systems operated under distributed

Manuscript received December 22, 1992; revised May 15, 1994. Recommended by Associate Editor, P. Nain. This work was supported in part by the Center for Advanced Technology in Telecommunications, Polytechnic University, and by the NSF under Grant NCR-9211417.

The author is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY 11201 USA.

IEEE Log Number 9407563.

scheduling policies. Perkins and Kumar [10] and Kumar and Seidman [6] have studied the problem of stability in a flexible manufacturing system with distributed scheduling. While simple distributed policies are shown to stabilize acyclic manufacturing networks in [10], a simple two-stage queueing network is presented in [6]. In this model instabilities occur in situations in which all servers of the system are strictly underloaded and the system is operated under a work-conserving policy. This example demonstrates how instabilities may occur in multistage manufacturing systems due to the phenomena of starvation and overloading. The queueing network that corresponds to the manufacturing system considered falls within the scope of the networks presented in our study. The ABP policy stabilizes the manufacturing system as long as the utilization of each machine is less than one. The scheduling of each machine i is determined by the size of the backlog of the different part types in i, as well as the size of the backlogs of those part types in downstream machines one hop away from i. Occasionally, the machines are forced to idle again, depending on the local state. The queueing system presented here extends the above manufacturing systems model to include the case where a part type may have the option of several alternative manufacturing scenarios and routing decisions are made. In Section II-C, this is discussed in more detail. The ABP policy in that case combines scheduling and routing decisions to achieve the same goal. In [11], a single-class network was considered, and a routing policy of the same nature was studied for Poisson arrivals.

For the arrival process, we assume that the burstiness is bounded by a deterministic bound [2], [3]. This traffic assumption has been used widely lately since the outputs of the traffic regulators that shape the traffic before it enters a high-speed network satisfy this type of constraint. Lu and Kumar [7] have used a similar type of traffic in the context of a manufacturing system.

The rest of the paper is organized as follows. In Section II, the queueing network is presented, and the correspondence with virtual circuit and datagrams networks as well as manufacturing systems is demonstrated. In Section III, the issue of stability is discussed, and the sufficient stability condition is shown. In Section IV, a parametric class of policies is specified, and their stability is studied. In Section V, the ABP policy is specified and investigated. Finally, in Section VI, the results are discussed and some open problems are presented.

II. The Network Model

We consider a network consisting of N servers and B buffers (Fig. 1). Each server i can serve any buffer from the set of buffers $\mathcal{B}_i$, $i = 1, \cdots, N$— it is allocated to the buffers according to some scheduling discipline. The sets $\mathcal{B}_i$ may be overlapping, that is, a buffer may be served by more than one server simultaneously. The served traffic from buffer j can be directed to any buffer of the set $\mathcal{R}_j$. A routing policy determines to which buffer in $\mathcal{R}_j$ the traffic from j is routed. From certain buffers, the traffic can be directed out of the system; this is indicated by including a 0 in the set $\mathcal{R}_j$. We make the natural assumption that, from every buffer, the work can be forwarded out of the system if it is routed through an appropriate sequence of buffers. We consider a fluid model for the work coming into the system. The time is continuous and the instantaneous rate with which work comes to buffer i from outside is $a_i(t)$. For simplicity, we assume that the arrival stream can contain no impulses, that is, the arrival rate $a_i(t)$ for all buffers i is bounded uniformly over time by a bound λ. In the time interval (t_1, t_2), an amount of work equal to

$$\int_{t_1}^{t_2} a_i(t)\,dt$$

enters buffer i from outside. We assume that the arrival streams satisfy some burstiness constraints; that is, there are nonnegative numbers a_i, b_i, $i = 1, \cdots, B$ such that, for all $0 \le t_1 < t_2$, we have

$$\int_{t_1}^{t_2} a_i(t)\,dt \le a_i(t_2 - t_1) + b_i. \tag{2.1}$$

We will assume that the long-run average rate with which work enters the system in buffer i exists

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t a_i(s)\,ds = a_i, \qquad i = 1, \cdots, B$$

and a_i will be referred to as the arrival rate to buffer i. The latter assumption is not needed for the validity of the results. It is introduced only because it is conceptually simpler to think of the a_i as being the arrival rate to buffer i. When server i provides service to buffer $j \in \mathcal{B}_i$ which is nonempty, work leaves the buffer with a constant rate μ_{ij}, the service rate of server i at buffer j. If server i is assigned to buffer j continuously for an amount of time T and the traffic is directed to buffer l in $\mathcal{R}_j$, then an amount of work $T\mu_{ij}$ is transferred from j to l. The selection of buffer $j \in \mathcal{B}_i$ served by i and of buffer $l \in \mathcal{R}_j$ to where the traffic is directed is done by the control policy. This work is routed to some buffer in $\mathcal{R}_j$. We allow more than one server to serve the same buffer; in that case, the service rate is assumed to be equal to the sum of the rates of the servers that serve the buffer. When server i that serves buffer j and directs the traffic to buffer k switches to buffers l and m, respectively, a switchover time δ^i_{jklm} is involved during which the server idles. The results in the paper hold for arbitrary and distinct switchover times. For notational simplification but no loss of generality, however, we will assume that the switchover time denoted by δ and be the same for all servers and buffers. In the following, we discuss how certain networks fall within the scope of the above model.

A. Virtual Circuit Networks

A VC network is characterized by its topology graph $G = (V, E)$, the transmission rates of the links and the established VC's. The topology graph contains one node for every network node and one directed link $e = (v, w)$ for every communication link from node v to node w. The transmission rate of link e is denoted by μ_e. A virtual circuit i is specified by the sequence of links $e^i_1, \cdots, e^i_N$ that traverses as it goes through the network. The traffic of VC i enters the network at the origin node of e^i_1 and leaves it at the destination node

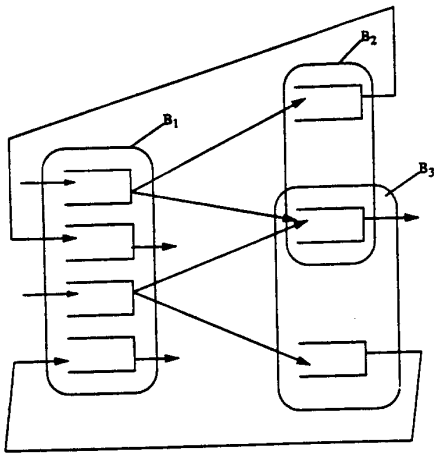

Fig. 1. An example of the network specified in Section II is depicted. There are three servers 1, 2, 3 (not depicted) which serve the queues in the sets B_1, B_2, and B_3, respectively. The traffic from each queue i may be routed to any queue j to which there is an arrow from i.

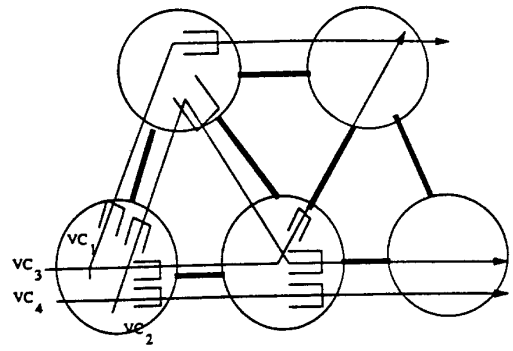

Fig. 2. A virtual circuit network.

of link e^i_N. A scheduling policy schedules for each link the transmissions of the virtual circuits going through it. Also, at certain times, the link may be forced to idle due to flow control actions.

A virtual circuit network is modeled by the above queueing system as follows (Fig. 2). Each link corresponds to a server with transmission capacity equal to its service rate. Clearly, in this case, the service rate of server (link) j is the same for all buffers in $\mathcal{B}_j$. There is one buffer for every VC i and every link e^i_k is traversed by i. The buffers of virtual circuit i contain traffic of that VC waiting to be transmitted through the corresponding link. When a link is allocated to any of the buffers of the VC's going through it, that buffer empties with rate equal to the link transmission rate. The traffic from the buffer of virtual circuit i at link e^i_j is routed to the buffer of the same virtual circuit at link e^i_{j+1}, except if e^i_j is the last link traversed by VC i; in the latter case, the traffic leaves the system. Hence, the set $\mathcal{R}_j$ contains a single element for each j and no routing is needed. The traffic in the VC's consists of streams of packets, and clearly, the link cannot switch from VC to VC in a time period smaller than a packet transmission time. In the assumptions we made, though, in the previous section, the traffic is considered as a continuous flow. For the results that we obtain here, it is not important whether the traffic is continuous or in terms of packets since the policies we propose can be easily modified to work with packetized traffic.

B. Datagram Networks

In datagram networks, the traffic of a communication session does not have to follow a specific route, but can be routed arbitrarily, and different packets may follow different paths to the destination. The packets are differentiated only by their destination nodes. When a packet arrives at a node which is not its final destination, a decision is made as to which outgoing link the packet will follow next; then the packet is placed at the outgoing buffer of that link. There are no constraints on which outgoing link the packets of each specific destination will follow; this is determined by the routing policy. Packets of several different destinations are waiting in the buffer of each outgoing link to be transmitted. The link scheduling discipline determines which is the packet to be transmitted next. In the corresponding queueing model, the buffers are in one-to-one correspondence with the links. For each server i, there is one buffer in $\mathcal{B}_i$ for each destination node. The traffic with destination node k, which is transmitted through link i, is stored in the corresponding buffer of $\mathcal{B}_i$. These buffers do not necessarily correspond to physical buffers in the origin node of link i. Their introduction enables the differentiation of the traffic, based on the destination and the link the traffic will go through. If, from the topology of the network or from other constraints, it turns out that the traffic of a specific destination never transverses link i, then the corresponding buffer in $\mathcal{B}_i$ will be permanently empty.

C. Manufacturing Systems

In manufacturing systems, several different types of parts are fabricated. A part type, in order to be manufactured, needs to be processed by a number of machines in some prespecified order. In flexible manufacturing systems, a machine can process more than one type of part. In this case, the machine has to switch from one part type to another at certain times since the different parts are not processed simultaneously. The switching of the machine involves a period during which the machine idles. The switching should be done rarely enough such that the fraction of time that the machine is utilized is higher than the loading.

Perkins and Kumar [10] have obtained a distributed scheduling policy that stabilizes any acyclic network of manufacturing machines. When cycles are formed by the routes of the parts in the manufacturing system, then the behavior of the system is more difficult to characterize. The intrinsic complexity of the problem was demonstrated by a counterintuitive example of instability in a simple manufacturing system in [6]. The queueing network considered here is readily interpreted into a manufacturing system model. The servers correspond to the machines, the arrival streams to the different part types, and the buffers of each machine store the part types at intermediate processing stages. If a part type needs to be processed by a machine more than once in its manufacturing cycle, then it

waits in a different buffer each time. When the part types have unique routes through the manufacturing system, then the sets $\mathcal{R}_j$ have a unique element, the next buffer that the parts will join when they leave from j. Kumar and Seidman in [6] propose a scheduling policy with a supervisor mechanism that stabilizes the manufacturing system as long as the load of each machine is less than one. The supervisory mechanism, though, relies on knowledge of the arrival rates. The ABP policy stabilizes the system without the arrival rate information. The model considered here extends the one considered in [10], [6] to include cases where a part type has the option to follow different routes at certain points of its processing, in which case routing is performed.

III. Necessary and Sufficient Stabilizability Conditions

We are interested in the average rate with which work can be served. We consider that certain throughput rates can be achieved if there exists a policy under which the network is stable when the long-run average arrival rates are equal to the desired throughputs. Denote by $X_j(t)$ the amount of work in buffer j at the time t.

Definition: The system is stable under a policy π if

$$\limsup_{t\to\infty} \sum_{j=1}^{B} X_j(t) < D$$

where D may depend only on the arrival rates $\boldsymbol{a} = (a_1, \cdots, a_N)$, the burstiness coefficients $\boldsymbol{b} = (b_1, \cdots, b_N)$, and the service rates $\mu = (\mu_{ij} : i = 1, \cdots, N, j \in \mathcal{B}_i)$, but not on the initial condition.

The following condition on the arrival and service rates is necessary and sufficient for the existence of a policy under which the network is stable.

C1: There exist nonnegative numbers f_{jk}, $j = 1, \cdots, B$, $k \in R_j$ that satisfy the flow conservation equations

$$a_j + \sum_{k:j\in R_k} f_{kj} = \sum_{k\in R_j} f_{jk}, \qquad j = 1, \cdots, B. \tag{3.1}$$

Also, there exist nonnegative numbers u_j^i, $i = 1, \cdots, N$, $j = 1, \cdots, B$ such that

$$\sum_{j\in\mathcal{B}_i} u_j^i < 1, \qquad i = 1, \cdots, N, \tag{3.2a}$$

$$\sum_{l\in R_j} f_{jl} \le \sum_{i:j\in\mathcal{B}_i} u_j^i \mu_{ij}, \qquad j = 1, \cdots, B. \tag{3.2b}$$

More specifically, it is shown that C1 is sufficient for stability if the burstiness condition (2.1) holds, while C1 is necessary for stability when the arrival process satisfies a "bounded idleness" condition, to be specified later. No existence of the arrival rate is needed. An intuitive justification of C1 follows. The number f_{jk} in the condition C1 represents the long-run average rate with which work is transferred from buffer j to buffer k. Hence, (3.1) is indeed a flow conservation equation. A collection of nonnegative numbers f_{jk}, $j = 1, \cdots, B$, $k \in R_j$ which satisfy (3.1) will be referred to as flow in the following. The number u_j^i represents the fraction of time that server i spends serving buffer j. Inequality (3.2b) expresses the fact that the rate with which work leaves buffer j [the sum on the left side of (3.2b)] cannot be greater than the server capacity [the sum on the right of (3.2b)] that is allocated to that buffer.

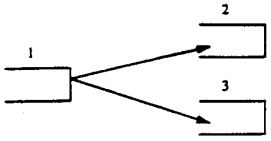

Fig. 3. A network with routing control.

In the case of a virtual circuit network, condition C1 is equivalent to the condition that, for every link, the sum of the average rates of all virtual circuits that go through the link is strictly less than its transmission capacity. This can be easily seen by observing the following. Since there is no routing involved, $\mathcal{R}_j$ contains a single element for all j's and f_{jk} is equal to the arrival rate of the virtual circuit that buffer j corresponds to (recall that in the VC network case, each buffer corresponds to one virtual circuit). Each buffer can be served by one server (link) only; therefore, the sums at the second inequality in (3.2) are reduced to single terms; these inequalities are equivalent to the fact that the traffic load of every link is less than one. The proof of the necessity of C1 for stabilizability follows, while the sufficiency will be proved in Section IV where the parametric back-pressure policy, the predecessor of the ABP, will be specified.

A. Necessity

When the long-run average rate with which work goes from buffer j to k as well as the long-run average fraction of time spent by server i serving buffer j exist, the intuitive interpretation of C1 given earlier turns readily into a proof of the necessity of C1. These long-run averages, though, do not exist for every policy, even if the system is stable, as illustrated in the following counterexample; therefore, a different approach should be taken to show the necessity of C1.

Counterexample: Consider the system in Fig. 3 that consists of three buffers. Buffer 1 receives work with constant rate 0.8, and is served by a server of higher service rate; therefore, it leaves the buffer instantaneously. The served work is routed either to buffer 2 or to 3, from where it leaves the system instantaneously. The switchover time is equal to 0. Clearly, as long as the server does not idle, the system will be stable. The routing is specified by the variable $r(t)$ which represents the buffer that is fed from buffer 1 at time t. Let

$$r(t) = \begin{cases} 2, & 3^{2k} \le t \le 3^{2k+1},\ k = 0, 1, \cdots \\ 3, & 3^{2k+1} \le t \le 3^{2k+2},\ k = 0, 1, \cdots. \end{cases}$$

If f_{12}^i is the average flow from buffer 1 to buffer 2 at the ith routing transition, then we can see that f_{12}^i fluctuates between values which are greater than $\frac{8}{15}$ and less than $\frac{4}{15}$; therefore, it cannot converge, and the long-run average flow does not exist. ◊

Next, we show the necessity of condition C1. The necessity follows when the arrival streams satisfy the following condition.

S1: There exist nonnegative numbers $\hat{b}_i$, $i = 1, \cdots, B$ such that

$$\int_{t_1}^{t_2} a_i(\tau)\, d\tau \geq a_i(t_2 - t_1) - \hat{b}_i.$$

Condition S1 can be viewed as a constraint on the idling of the arrival streams.

Theorem 3.1: If the system is stable and condition S1 holds, then condition C1 holds as well.

The proof of the theorem will follow after a lemma. The following definition is needed. A collection of nonnegative numbers f_{jk}, $j = 1, \cdots, B, k \in \mathcal{R}_j$ is called a *superflow* if

$$a_j + \sum_{k:j\in\mathcal{R}_k} f_{kj} \leq \sum_{k\in\mathcal{R}_j} f_{jk}, \qquad j = 1, \cdots, B \tag{3.3}$$

and a strict superflow if the inequality in (3.3) is strict whenever $\sum_{k\in\mathcal{R}_j} f_{jk} > 0$. The notions of superflow and strict superflow do not correspond to any physical quantities in the system, and they are introduced only to be used in the proofs.

Lemma 3.1: If $f' = (f'_{jk}, j = 1, \cdots, B, k \in \mathcal{R}_j)$ is a superflow, then there exists a flow $f = (f_{jk}, j = 1, \cdots, B, k \in \mathcal{R}_j)$ such that

$$f \leq f' \tag{3.4}$$

where the inequality (3.4) holds componentwise. If f' is a strict superflow, then there exists a flow f such that (3.4) holds with strict inequality, for the nonzero elements of f and f'.

Proof: Consider the transformation T defined by $f^{i+1} = T(f^i)$where

$$f_{jk}^{i+1} = \begin{cases} \dfrac{a_j + \sum_{l:j\in\mathcal{R}_l} f_{lj}^i}{\sum_{l\in\mathcal{R}_j} f_{jl}^i} f_{jk}^i, & \text{if } \sum_{l\in\mathcal{R}_j} f_{jl}^i > 0 \\ f_{jk}^i, & \text{otherwise.} \end{cases}$$

It is claimed that if f^i is a superflow, then f^{i+1} is a superflow as well and

$$f^{i+1} \leq f^i \tag{3.5}$$

while if f^i is a strict superflow, then f^{i+1} is a strict superflow as well, and (3.5) holds with strict inequality. Notice that (3.5) follows readily since the fact that f^i is a superflow (strict superflow) implies readily that

$$\frac{a_j + \sum_{l:j\in\mathcal{R}_l} f_{lj}^i}{\sum_{l\in\mathcal{R}_j} f_{lj}^i}$$

is less (strictly less) than one. From the definition of $T(\,)$, we have

$$\sum_{k\in\mathcal{R}_j} f_{jk}^{i+1} = a_j + \sum_{l:j\in\mathcal{R}_l} f_{jl}^i. \tag{3.5a}$$

From (3.5) and (3.5a), it follows that

$$\sum_{k\in\mathcal{R}_j} f_{jk}^{i+1} \geq a_j + \sum_{l:j\in\mathcal{R}_j} f_{jl}^{i+1} \tag{3.5b}$$

if f^i is a superflow, while (3.5b) holds with strict inequality if f^i is a strict superflow. Relation (3.5b) shows that f^{i+1} is a superflow and (3.5b), with strict inequality, that it is a strict superflow.

Obviously, if f^i is a flow, then $f^{i+1} = f^i$.Consider the sequence of superflows f^i, $i = 0, 1, \cdots$ defined as $f^0 = f'$, $f^{i+1} = T(f^i)$, $i = 0, 1, \cdots$. This sequence is nonincreasing, and as a consequence, it converges to a fixed point of $T(\,)$ denoted by f^∞. Note, though, that all fixed points of T are flows; hence, f^∞ is a flow which is less than or equal to f' in general, and strictly less than f' if f' is a strict superflow. ◊

Remark: If f' in the above lemma contains no cycles, that is, there is no sequence of buffers $l_1, l_2, \cdots, l_k$ such that $l_1 = l_k$, $l_{j+1} \in \mathcal{R}_j$, $j = 1, \cdots, k-1$, $f_{l_j l_{j+1}} > 0$, $j = 1, \cdots, k-1$, then the sequence of the superflows f^i, $i = 0, 1, \cdots$ in the proof of Lemma 3.1 will converge after a finite number of steps. If there is a cycle in f', then the convergence takes an infinite number of steps.

Now, we can proceed to the proof of the theorem.

Proof of Theorem 3.1: Assume that the system is stable. From the definition of stability, there exists D such that for every $X(0)$, there exists T, which may depend on $X(0)$, for which

$$\sum_{j=1}^{B} X_j(t) \leq D, \qquad t \geq T. \tag{3.6}$$

Assume that initially each buffer has a backlog equal to $2D$, and T is such that (3.6) holds for this initial condition. Let Q_{jk} be the total amount of work that has been transferred from buffer j to buffer $k \in \mathcal{R}_j$ in the time interval $(0, T')$, where $T' > T$ is a time to be selected appropriately, as we will see in the following. Clearly, $X_j(T') \leq D$, $j = 1, \cdots, B$ and

$$X_j(T') = 2D + \sum_{k:j\in\mathcal{R}_k} Q_{kj} - \sum_{k\in\mathcal{R}_j} Q_{jk} + \int_0^{T'} a_j(t)\, dt \leq D, \qquad j = 1, \cdots, B. \tag{3.6a}$$

Consider the nonnegative vector $f = (f_{jk} : j = 1, \cdots, B, k \in \mathcal{R}_j)$ where $f_{jk} = Q_{jk}/T'$. If we divide (3.6a) by T', and using S1, we get

$$\begin{aligned} a_j + \sum_{k:j\in\mathcal{R}_k} f_{kj} &\leq \sum_{k\in\mathcal{R}_j} f_{jk} + a_j - \frac{1}{T'}\int_0^{T'} a_j(t)\, dt - \frac{D}{T'}, \quad j = 1, \cdots, B \\ &\leq \sum_{k\in\mathcal{R}_j} f_{jk} + \frac{\hat{b}_j}{T'} - \frac{D}{T'}. \end{aligned} \tag{3.7}$$

If we select $D > \max_{j=1,\cdots,B} \hat{b}_j$ and condition S1 holds, then (3.7) implies that f is a strict superflow. Let T_{jk}^i be the amount of time that server i serves buffer j and directs the traffic to buffer k during the time period from 0 to T'. Define

$$\hat{u}_j^i = \frac{\sum_{k\in\mathcal{R}_j} T_{jk}^i}{T'}.$$

Since each server may serve at most one buffer at a time, we have

$$\sum_{j\in\mathcal{B}_i}\hat{u}_j^i \le 1. \tag{3.8}$$

We also have

$$Q_{jk} = \sum_{i:j\in\mathcal{B}_i} T_{jk}^i \mu_{ij}, \qquad k\in\mathcal{R}_j,$$

which yields

$$\sum_{k\in\mathcal{R}_j} Q_{jk} = \sum_{k\in\mathcal{R}_j}\sum_{i:j\in\mathcal{B}_i} T_{jk}^i\mu_{ij}, \qquad k\in\mathcal{R}_j$$

and therefore

$$\sum_{k\in\mathcal{R}_j} f_{jk} = \sum_{i:j\in\mathcal{B}_i}\hat{u}_j^i\mu_{ij}, \qquad k\in\mathcal{R}_j. \tag{3.9}$$

From Lemma 3.1, since $\boldsymbol{f}$ is a strict superflow, there exists a flow f' such that $f'_{jk} < f_{jk}$ if $f_{jk} > 0$. Let

$$\epsilon = \max\left\{\frac{f'_{jk}}{f_{jk}} : j = 1,\cdots,B,\ k\in\mathcal{R}_j,\ f_{jk}>0\right\}.$$

Clearly, $\epsilon < 1$. Define

$$u_j^i = \epsilon\hat{u}_j^i, \qquad i = 1,\cdots,n,\ j\in\mathcal{B}_i.$$

It holds that

$$\sum_{j\in\mathcal{B}_i} u_j^i < 1.$$

By multiplying each part of (3.9) by ϵ, and given the fact that $f'_{jk} \le \epsilon f_{jk}$, it follows that

$$\sum_{k\in\mathcal{R}_j} f'_{jk} \le \sum_{i:j\in\mathcal{B}_i} u_j^i\mu_{ij}, \qquad j = 1, 2,\cdots,B. \tag{3.10}$$

◊

The necessity of strict inequality in relation (3.2a) of condition C1 is due to the definition of stability that we have, and more specifically to the requirement that the asymptotic bound of the backlog is independent of the initial conditions. It is possible that bounded backlog is guaranteed without strict inequality in (3.2a). This bound, though, cannot be independent of the initial condition. This is illustrated in the following counterexample. Consider the case of a single-server queue with constant instantaneous arrival rate equal to a and service rate μ. Condition C1, with strict inequality in (3.2a), boils down to the condition $a < \mu$. If equality is allowed in (3.2a), we may have $a = \mu$. In the latter case, and if the instantaneous arrival rate is constant and equal to the service rate μ, then the backlog at all times will be equal to the backlog at the time instant 0. Therefore, the network will be unstable according to the definition of stability considered in the paper, since there is no asymptotic bound in the backlog independent of the initial condition.

IV. The Parametric Back-Pressure Policy

In this section, we will present and study the parametric back-pressure policy PBP(α). This policy determines whether a server idles or not, and if it does not idle, which queue is served and where the traffic is directed. The frequency with which a server is switched from queue to queue depends on a parameter $\alpha > 1$. We include α as an argument in the name of the policy to emphasize that dependence

PBP(α) :While the server i serves a buffer j_0 and directs the traffic to a buffer k_0, it is constantly monitoring the quantity

$$A_i(t) = \max_{l\in\mathcal{B}_i}\left\{\mu_{il}\max_{n\in\mathcal{R}_l}\{X_l(t) - X_n(t)\}\right\}. \tag{4.1}$$

If

$$\alpha\mu_{ij_0}(X_{j_0}(t) - X_{k_0}(t)) > 0$$

and

$$A_i(t) \ge \alpha\mu_{ij_0}(X_{j_0}(t) - X_{k_0}(t))$$

then the server is rescheduled to serve the queue $l\in\mathcal{B}_i$, and the traffic is directed to queue $n\in\mathcal{R}_l$, which together realize the maximum in (4.1), with ties broken arbitrarily. Service is reassumed after the switchover time δ.

If

$$A_i(t) \le 0$$

then the server idles for a time period δ. If at the end of the idling period it is still $A_i(t) \le 0$, then the server restarts an idling period of the same duration. Otherwise, server i is rescheduled to serve the queue $l\in\mathcal{B}_i$, and the traffic is directed to queue $n\in\mathcal{R}_l$, which together realize the maximum in (4.1).

Some clarifications on the operation of PBP(α) follow. According to PBP(α), the served traffic of buffer l is routed to buffer n which achieves the maximum in

$$\max_{n\in\mathcal{R}_l}\{X_l(t) - X_n(t)\} \tag{4.1a}$$

whenever buffer l is served. Server i selects which buffer $l\in\mathcal{B}_i$ to serve based on the terms

$$\mu_{il}\max_{n\in\mathcal{R}_l}\{X_l(t) - X_n(t)\}, \qquad l\in\mathcal{B}_i.$$

It selects the one which achieves the maximum in (4.1). The policy is illustrated in Fig. 4. In order to avoid server i switching from queue to queue too often, the policy reschedules the server only if the quantity (4.1a) for the queue l under service becomes considerably smaller than $A_i(t)$. How much smaller it should become is determined by α, which regulates how often the server switches. This feature is reminiscent of the clear a fraction policy considered in [10]. If the backlog in all the downstream queues is larger than the backlog in the queues of $\mathcal{B}_i$, that is the quantity $A_i(t)$ is negative, then the server i idles.

Note that the scheduling of server i relies on information about the queue lengths of the buffers in $\mathcal{B}_i$ and in $\mathcal{R}_j$, $j\in\mathcal{B}_i$. This is local information for server i in several practical

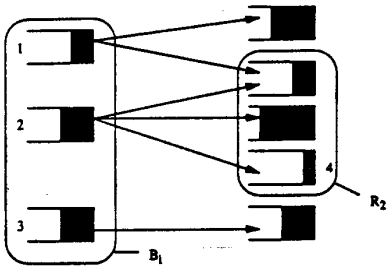

Fig. 4. Server i is allocated to one of the queues 1, 2, 3. When the backlogs areas illustrated in the picture, then queue 2 will be selected for service and the traffic will be directed to buffer 4.

systems. For example, in the case of the VC network, the buffers in $\mathcal{B}_i$ and in $\mathcal{R}_j$, $j \in \mathcal{B}_i$ reside in the origin and destination node of the server/link j; therefore, policy PBP(α) can be implemented in a distributed fashion.

The stabilizability properties of PBP(α) are stated in the following.

Theorem 4.1: If the arrival and service rates satisfy condition C1, then there exists an $\alpha > 1$ such that the system is stable under PBP(α).

Note that the policy PBP(α) needs only the service rates for its application. Furthermore, the stability holds for any value of the parameter δ in the definition of the policy. The proof of the theorem relies on the following lemma, which is a drift-type condition on the sum of the squared backlogs in the system.

Lemma 4.1: If the arrival and service rates satisfy condition C1, then there exists an α which depends only on the arrival and service rates and the burstiness coefficients such that when policy PBP(α) is employed, the following holds. There exist numbers D_0, T, and ϵ such that

$$\sum_{j=1}^{B} X_j^2(t+T) - \sum_{j=1}^{B} X_j^2(t) < -\epsilon \quad \text{if} \quad \sum_{j=1}^{B} X_j^2(t) \geq D_0.$$

Proof: The proof of the lemma is lengthy, and relies on some intermediate results. It is given in Appendix A. ◊

Proof of Theorem 4.1: We show that there exists a $\hat{T}$, which may depend on $\boldsymbol{X}(0)$, $\boldsymbol{a}$, μ, and δ, and a D', which may depend on $\boldsymbol{a}$, μ, and δ, such that

$$\sup_{t \geq \hat{T}} \sum_{j=1}^{B} X_j(t) \leq D'. \tag{4.1b}$$

Let T and ϵ be such that

$$\sum_{j=1}^{B} X_j^2(t+T) - \sum_{j=1}^{B} X_j^2(t) < -\epsilon \quad \text{if} \quad \sum_{j=1}^{B} X_j^2(t) \geq D_0. \tag{4.1c}$$

From Lemma 4.1, T and ϵ as above exist. Let $\hat{k}$ be the smallest integer k such that

$$\sum_{j=1}^{B} X_j^2(kT) \leq D_0.$$

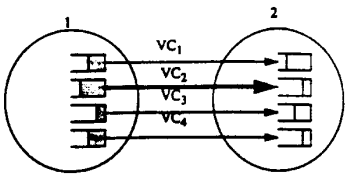

Fig. 5. The link from node 1 to node 2 that carries 4 VC's and the buffers in the origin and destination nodes of the link are illustrated. When the backlogs are as they appear in the picture, the link will transmit VC2.

From (4.1c), clearly such a $\hat{k}$ exists. Define $\hat{T} = \hat{k}T$ and

$$\hat{D} = \left(\sqrt{D_0} + \sum_{j=1}^{B} (a_j T + b_j) \right)^2.$$

We show by induction that we have

$$\sum_{j=1}^{B} X_j^2(lT) \leq B\hat{D}, \qquad l \geq \hat{k}. \tag{4.1d}$$

The induction step is as follows. If $\sum_{j=1}^{B} X_j^2(lT) \leq D_0$, then, clearly, $\sum_{j=1}^{B} X_j^2((l+1)T) \leq B\hat{D}$. If $D_0 \leq \sum_{j=1}^{B} X_j^2(lT) \leq B\hat{D}$, then $\sum_{j=1}^{B} X_j^2((l+1)T) \leq \sum_{j=1}^{B} X_j^2(lT) - \epsilon$, and therefore, $\sum_{j=1}^{B} X_j^2((l+1)T) \leq \hat{D}$. ◊

The actions taken by policy PBP(α) in the general network correspond to scheduling the server, routing of the traffic, and idling of the server. If routing is not involved, then the policy is simplified considerably. This is the case in a virtual circuit network where PBP(α) acts as follows. Every link is allocated to the virtual circuits that go through it based on the backlogs of the buffers of these VC's in its origin node and in its destination node. For each VC, the differences of the backlog of the buffer in the origin node of the link minus that of the buffer in the destination node of the link are formed (Fig. 5). If all differences are negative, then the link idles. Otherwise, the quantity

$$A_i(t) = \max_{l \in \mathcal{B}_i} \{X(t) - X_{\hat{l}}(t)\} \tag{4.1e}$$

is computed where l and $\hat{l}$ are the buffers that correspond to the same virtual circuit in the origin and destination node, respectively, of link i. If

$$A_i(t) > \alpha \mu_{ij} (X_j(t) - X_{\hat{j}}(t))$$

where j, $\hat{j}$ are the buffers that correspond to the VC currently transmitted, then the server switches to the VC that achieves the maximum in (4.1e).

Note that in the model, it is possible that work goes through the same buffer more than once. It appears that this possibility may lead to instabilities under a distributed policy. The following is an intuitive explanation of why this is not happening with PBP(α). The goal of the policy is to push all the traffic out of the network. It is possible that work may

visit a buffer more than once, following a cycle of buffers with small backlogs, instead of following a route out of the network, when the buffers that lead out of the network are congested. Eventually, though, the latter buffers will empty, and the work will find its way out of the network by selecting the route with the low-backlog buffers.

The PBP(α) policy achieves a bounded backlog in the network under the necessary and sufficient stabilizability condition C1, and is indeed distributed since the decisions at each node rely on the one hop away state information. Nevertheless, the arrival and service rates and the topology of the network need to be known in order to select the appropriate α. In the next section, the ABP policy (an adaptive version of the PBP) is obtained, which does not rely on knowledge of the arrival and service rates.

V. The Adaptive Back-Pressure Policy

The adaptive back-pressure policy is identical to PBP(α), except that α is not preselected, but is computed at each node based on the local (one hop away) state. Each server i computes a parameter $\alpha_i(t)$ based on the lengths of the queues that it may serve and the queues that it may direct the traffic to, as follows. Consider a function $g: R^+ \to R^+$ which is nonincreasing, strictly greater than 1, and such that

$$\lim_{x\to\infty} g(x) = 1$$

$$\lim_{x\to\infty} x(g(x)-1) = \infty. \tag{5.1}$$

The function $g(x) = 1 + x^{-a}$ satisfies (5.1) for all $0 < a < 1$. Define $\alpha_i(t)$ as

$$\alpha_i(t) = g\left(\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\}\right).$$

The policy ABP is as follows

ABP: Each server acts as in the PBP(α) policy with the difference that server i uses the locally computed$\alpha_i(t)$ instead of α.

The following holds for ABP.

Theorem 5.1: The network is stable under ABP if condition C1 holds.

The proof of Theorem 5.1 is the same as that of Theorem 4.1, given the drift condition stated in the following lemma.

Lemma 5.1: If the arrival and service rates satisfy condition C1 and the network is operated under ABP, then there exist numbers D_0, T, and ϵ such that

$$\sum_{j=1}^{B} X_j^2(t+T) - \sum_{j=1}^{B} X_j^2(t) < -\epsilon \qquad \text{if} \sum_{j=1}^{B} X_j^2(t) \geq D_0.$$

Proof: The proof is given in Appendix B. ◊

Note that there is no parameter estimation taking place in the policy. In fact, the average arrival rates might not even exist. The logic of the adaptive policy is that it adjusts α such that it goes closer to 1 with a certain rate as the backlog increases. In this manner, it is achieved that the servers switch to the queues that have the heavier backlog difference frequently enough.

VI. Conclusions and Discussion

In the ABP policy, each server is scheduled based on the lengths of the queues that it serves and of the queues one hop away, which were assumed to be instantaneously available to the server in our study. This is not always the case, though. For example, in the virtual circuit network, the link scheduling was based on the queue lengths at the origin and the destination nodes of the link. Clearly, the queue lengths in the destination node could not be made available at the origin node without a delay greater than or equal to the propagation delay of the link. In high-speed networks, the link propagation time is enough for the state of the queues to change considerably. Therefore, there will be a discrepancy between the actual queue lengths and those available to the server. Nevertheless, as long as the difference between the actual queue lengths and those available to the server is bounded independently of the queue length values, then the results obtained in the paper remain unaffected. That is, the ABP policy stabilizes the network with delayed information about the queue lengths as well.

According to the ABP policy, server i idles if the quantity $A_i(t)$becomes less than or equal to 0 or, in other words, if for every queue $l \in \mathcal{B}_i$ the backlog of all queues in $\mathcal{R}_l$ is greater than or equal to that of queue l. The results remain unaffected when $A_i(t)$ is compared with an arbitrary negative number instead of 0 in the definition of ABP. That is, the server idles only if the backlog in the downstream queues becomes greater than that of the upstream queues by a certain amount.

The delay through the network is an issue left open for further research. Bounded backlogs imply bounded delay; therefore, under the ABP policy, the delay will be finite when the stability condition C1 holds. There are several questions which are left open, though. How does the delay vary when the function g that estimates the parameter α in each node changes? Which is a good choice of g as far as the delay is concerned? As we said earlier, the backlog in the network remains bounded even if the server i idles whenever $A_i(t) < h$ for an arbitrary h, and not necessarily for $h = 0$. What is a good choice of h for small delays? Also, what will be the effect of the delayed information about the state of the one hop away queues on the delays through the network, and how will the latter be affected from the propagation delay? The investigation of these questions might lead to refinements of ABP with improved performance with respect to delay.

Appendix A

Lemma 4.1 is proved here. Two lemmas precede its proof. The first quantifies the property of policy PBP(α) that the switching of the server becomes less frequent as the queue lengths increase and more frequent as α approaches 1. Note that the maximum rate with which the difference of two queues may vary (increase or decrease) is less than or equal to

$$B\lambda + \sum_{i=1}^{N}\sum_{j\in\mathcal{B}_i}\mu_{ij} = M.$$

Lemma A.1: If for a server i we have

$$\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\geq\frac{4\alpha}{\alpha-1}MT \tag{A.1}$$

then i will switch at most once in the time interval $(t, t+T)$ and

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j}Q_{jl}^i(X_j(t)-X_l(t))\\&\geq\frac{1}{\alpha}(T-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\\&\quad-4M(T+\delta)^2\end{aligned} \tag{A.2}$$

where Q_{jl}^i is the amount of work served by server i and transferred from buffer j to buffer l during the time interval $(t, t+T)$.

Proof: At time t, server i serves queue j_1 and directs the traffic to queue l_1. We distinguish two cases. Assume first that

$$\mu_{ij_1}(X_{j_1}(t)-X_{l_1}(t))=\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}. \tag{A.3}$$

Since the maximum rate with which any difference $\mu_{ij_1}(X_{j_1}(t)-X_{l_1}(t))$ may vary is M, we have

$$\begin{aligned}&\alpha\mu_{ij_1}(X_{j_1}(t')-X_{l_1}(t'))\\&\geq\alpha\left(\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}-2MT\right),\\&\quad t\in(t,t+T).\end{aligned} \tag{A.4}$$

Also, for any pair of queues $j\in\mathcal{B}_i$, $l\in R_j$

$$\begin{aligned}&\mu_{ij}(X_j(t')-X_l(t'))\\&\geq\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}+2MT,\\&\quad t'\in(t,t+T).\end{aligned} \tag{A.5}$$

From (A.4), (A.5) we can see that if (A.1) holds, then

$$\alpha\mu_{ij_1}(X_{j_1}(t')-X_{l_1}(t'))\geq\mu_{ij}(X_j(t')-X_l(t')),\quad j\in\mathcal{B}_i,\ j\neq j_1,\ l\in\mathcal{R}_j t'\in(t,t+T) \tag{A.5a}$$

and the server will not switch in the time interval $(t, t+T)$. Then, clearly

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j}Q_{jl}^i(X_j(t)-X_l(t))\\&=Q_{j_1l_1}^i(X_{j_1}(t)-X_{l_1}(t))\\&=T\mu_{ij_1}(X_{j_1}(t)-X_{l_1}(t))\\&=T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\end{aligned} \tag{A.6}$$

and (A.2) follows. Note that (A.6) assumes that the server is always busy during $(t, t+T)$, which is a valid assumption based on (A.1).

Assume now that (A.3) does not hold. From the definition of the policy at time t, we have

$$\alpha\mu_{ij_1}(X_{j_1}(t)-X_{l_1}(t))\geq\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}. \tag{A.7}$$

If server i will not switch queues during the interval $(t, t+T)$, then from (A.7), relation (A.2) follows easily. If the server will switch, then let t_1 be the first time after t at which the server switches. Let j_2 be the queue that is served after the switching, and l_2 the queue to which the traffic is routed. At time $t_1+\delta$, we have

$$\begin{aligned}&\mu_{ij_2}(X_{j_2}(t_1+\delta)-X_{l_2}(t_1+\delta))\\&=\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t_1+\delta)-X_l(t_1+\delta)\}\right\}\\&\geq\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}-2M(t_1-t+\delta)\end{aligned} \tag{A.8}$$

and arguing similarly as above

$$\begin{aligned}&\mu_{ij_2}(X_{j_2}(t')-X_{l_2}(t'))\\&\quad\geq\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}-2MT.\end{aligned} \tag{A.8a}$$

Equation (A.8a) together with (A.5) imply that if (A.1) holds, then the server will not switch again in the time interval $(t_1+\delta, t+T)$. Let $\hat{Q}_{jl}^i$ be the amount of work served by server i and transferred from buffer j to buffer l during the time interval (t, t_1); let $\tilde{Q}_{jl}^i$ be the same quantity for the time interval $(t_1+\delta, t+T)$. Then, clearly, $Q_{jl}^i=\hat{Q}_{jl}^i+\tilde{Q}_{jl}^i$ and

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j}\hat{Q}_{jl}^i(X_j(t)-X_l(t))\\&=(t_1-t)\mu_{ij_1}(X_{j_1}(t)-X_{l_1}(t))\\&\geq\frac{1}{\alpha}(t_1-t)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\end{aligned} \tag{A.9}$$

while

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j}\tilde{Q}_{jl}^i(X_j(t)-X_l(t))\\&=\tilde{Q}_{j_2l_2}^i(X_{j_2}(t)-X_{l_2}(t))\\&=(T+t-t_1-\delta)\mu_{ij_2}(X_{j_2}(t)-X_{l_2}(t)).\end{aligned} \tag{A.10}$$

Notice that

$$\mu_{ij_2}(X_{j_2}(t)-X_{l_2}(t))\geq\mu_{ij_2}(X_{j_2}(t_1+\delta)-X_{l_2}(t_1+\delta))-4M(t_1-t+\delta). \tag{A.11}$$

From (A.8), (A.10), and (A.11), we get

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j}\tilde{Q}_{jl}^i(X_j(t)-X_l(t))\\&\geq(T+t-t_1-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\\&\quad-4TM(t_1-t+\delta).\end{aligned} \tag{A.12}$$

From (A.9) and (A.12), we have

$$\begin{aligned}
&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j} Q_{jl}^i(X_j(t)-X_l(t))\\
&\geq \frac{1}{\alpha}(t_1-t)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\\
&\quad+(T+t-t_1-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\\
&\quad-4TM(t_1-t+\delta)\\
&\geq \frac{1}{\alpha}(T-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}\\
&\quad-4M(T+\delta)^2. \qquad \text{(A.12a)}
\end{aligned}$$

◊

In the following, we show that one of the differences $X_j(t)-X_l(t)$, $j=1,\cdots,B$, $l\in\mathcal{R}_j$ is on the order of $\sqrt{\sum_{j=1}^B X_j^2(t)}$. A result similar to the following lemma has been shown in [11]. The lemma is included here for completeness. Notice that it holds independently of the policy and is a characteristic of the network.

Lemma A.2: There is a constant $c>0$ that depends only on the topology of the network so that

$$\max_{j=1,\cdots,B,l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\geq c\sqrt{\sum_{j=1}^B X_j^2(t)}.$$

Proof: Consider the queue j_0 with the maximum length. Clearly

$$X_{j_0}(t)\geq\sqrt{\sum_{j=1}^B X_j^2(t)/B}.$$

There exists a sequence of queues through which the work can be routed out of the network; that is, there exist a sequence $i_1,\cdots,i_n$ such that $n\leq B$, $i_1=j_0$, $0\in\mathcal{R}_{i_n}$, $i_{k+1}\in\mathcal{R}_{i_k}$, $k=1,\cdots,n-1$. Then we have

$$\sum_{k=1}^{n-1}(X_{i_k}(t)-X_{i_{k+1}}(t))=X_{i_1}(t)=X_{j_0}(t)$$

$$\Rightarrow \max_{k=1,\cdots,n-1}(X_{i_k}-X_{i_{k+1}})\geq\frac{X_{j_0}(t)}{B}\geq\sqrt{\frac{\sum_{j=1}^B X_j^2(t)}{B^3}}.$$

◊

Proof of Lemma 4.1: With simple calculations, we get for every T

$$\begin{aligned}
\sum_{j=1}^B X_j^2(t+T)-\sum_{j=1}^B X_j^2(t)&=2\sum_{j=1}^B(X_j(t+T)-X_j(t))X_j(t)\\
&\quad+\sum_{j=1}^B(X_j(t+T)-X_j(t))^2. \qquad \text{(A.13)}
\end{aligned}$$

Notice that

$$(X_j(t+T)-X_j(t))^2\leq M^2T^2,\qquad j=1,\cdots,B$$

where M is the maximum rate with which any queue may vary, and it has been defined before Lemma A.1. Hence,

$$\sum_{j=1}^B(X_j(t+T)-X_j(t))^2\leq BM^2T^2. \qquad \text{(A.14)}$$

In the following, we proceed to bound the first term of the right-hand side of (A.13). Let Q_{jl}^i be the amount of work served by server i and transferred from buffer j to buffer l during the time interval $(t, t+T)$. Let Q_j be the amount of work transferred to buffer j from outside, and Q_{j0}^i the amount of work served by server i and transferred from buffer j to outside during $(t, t+T)$. Clearly[0]

$$X_j(t+T)=X_j(t)+Q_j+\sum_{l:j\in\mathcal{R}_l}\sum_{i=1}^N Q_{lj}^i-\sum_{l\in\mathcal{R}_j}\sum_{i=1}^N Q_{jl}^i.$$

Hence, for the first term on the right side of (A.13), we h

$$\begin{aligned}
&\sum_{j=1}^B(X_j(t+T)-X_j(t))X_j(t)\\
&=\sum_{j=1}^B Q_jX_j(t)+\sum_{j=1}^B\sum_{l:j\in\mathcal{R}_l}\sum_{i=1}^N Q_{lj}^iX_j(t)\\
&\quad-\sum_{j=1}^B\sum_{l\in\mathcal{R}_j}\sum_{i=1}^N Q_{jl}^iX_j(t)\\
&=\sum_{j=1}^B Q_jX_j(t)+\sum_{j=1}^B\sum_{l\in\mathcal{R}_j}\sum_{i=1}^N Q_{jl}^i(X_l(t)-X_j(t)).
\end{aligned} \qquad \text{(A.15)}$$

From the burstiness constraints on the arrival streams, we have

$$Q_j\leq a_jT+b_j,\qquad j=1,\cdots,B$$

which, together with condition C1, gives

$$\begin{aligned}
\sum_{j=1}^B Q_jX_j(t)&\leq T\sum_{j=1}^B a_jX_j(t)+\sum_{j=1}^B X_j(t)b_j\\
&=T\sum_{j=1}^B\left(\sum_{l\in\mathcal{R}_j}f_{jl}-\sum_{l:j\in\mathcal{R}_l}f_{lj}\right)X_j(t)\\
&\quad+\sum_{j=1}^B X_j(t)b_j\\
&=T\sum_{j=1}^B\sum_{l\in\mathcal{R}_j}f_{jl}(X_j(t)-X_l(t))+\sum_{j=1}^B X_j(t)b_j\\
&\leq T\sum_{j=1}^B\sum_{l\in\mathcal{R}_j}f_{jl}\max_{k\in\mathcal{R}_j}(X_j(t)-X_k(t))\\
&\quad+\sum_{j=1}^B X_j(t)b_j. \qquad \text{(A.16)}
\end{aligned}$$

In the following, we upper bound the second term on the right side of (A.15). Note that

$$Q_{jl}^i(X_j(t)-X_l(t))\geq -M^2T^2,\qquad i=1,\cdots,N,\quad j,l=1,\cdots,B. \qquad \text{(A.17)}$$

This is so because, first

$$0 \le Q_{jl}^i \le MT, \qquad i = 1, \cdots, N, \quad j, l = 1, \cdots, B$$

and, second, if $X_j(t) - X_l(t) < -MT$, then $X_j(t') - X_l(t') \le 0$ for all $t' \in (t, t+T)$, and Q_{jl}^i will be equal to 0 since no server will serve queue j in the time interval $(t, t+T)$. By interchanging the order of the summation, we have

$$\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q_{jl}^i(X_l(t) - X_j(t)) = -\sum_{i=1}^{N}\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j} Q_{jl}^i(X_j(t) - X_l(t)). \quad \text{(A.18)}$$

Define $\hat{D} = \sum_{j=1}^{B} X_j^2(t)$. From Lemma A.2, we get that for some i, say $i = i_0$, we have

$$\begin{aligned}
&\max_{j\in\mathcal{B}_{i_0}}\left\{\mu_{i_0 j}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&= \max_{i=1,\cdots,N}\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\ge \min_{j\in\mathcal{B}_i, 1=1,\cdots,N}\{\mu_{ij}\}\max_{j\in\mathcal{B}_i}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} \ge c\sqrt{\hat{D}}.
\end{aligned}$$

Hence, for $\hat{D}$ large enough, inequality (A.1) is satisfied for at least one server. Let $\mathcal{F}$ be the set of servers for which (A.1) holds. Replacing in (A.18) from (A.2) and (A.17), we get

$$\begin{aligned}
&\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q_{jl}^i(X_l(t) - X_j(t)) \\
&\le -\sum_{i\in\mathcal{F}}\frac{1}{\alpha}(T-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + 4M(T+\delta)^2 + NB^2M^2T^2 \\
&\le -\sum_{i\in\mathcal{F}}\left(1 - \sum_{j\in\mathcal{B}_i} u_j^i\right)\frac{1}{\alpha}T\max_{j\in\mathcal{B}_i} \\
&\quad \cdot\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + 4M(T+\delta)^2 + NB^2M^2T^2 \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right)T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + \sum_{i\in\mathcal{F}}\frac{\delta}{\alpha}\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right)T\left(\frac{1}{\alpha} - 1\right)\max_{j\in\mathcal{B}_i} \\
&\quad \cdot\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\}. \qquad \text{(A.19)}
\end{aligned}$$

Define

$$u = \min_{i=1,\cdots,N}\left\{1 - \sum_{j\in\mathcal{B}_i} u_j^i\right\}, \quad m = \min_{i=1,\cdots,N, j\in\mathcal{B}_i}\mu_{ij}. \qquad \text{(A.19a)}$$

Then, we can easily check that

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\left(1 - \sum_{j\in\mathcal{B}_i} u_j^i\right)\frac{1}{\alpha}T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\qquad \ge u\frac{1}{\alpha}Tm\max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}. \qquad \text{(A.20)}
\end{aligned}$$

Clearly

$$\sqrt{\hat{D}} \ge \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}. \qquad \text{(A.21)}$$

Let $\mu = \max_{i=1,\cdots,N, j\in\mathcal{B}_i}\mu_{ij}$. Then, from (A.21)

$$\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \le \mu\sqrt{\hat{D}}, \qquad i = 1, \cdots, N$$

and

$$\sum_{i\in\mathcal{F}}\frac{\delta}{\alpha}\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \le N\frac{\delta}{\alpha}\mu\sqrt{\hat{D}}. \qquad \text{(A.22)}$$

From (A.21), and since $1/\alpha < 1$, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right)T\left(\frac{1}{\alpha} - 1\right)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\ge \sum_{i\in\mathcal{F}}T\left(\frac{1}{\alpha} - 1\right)2\mu\sqrt{\hat{D}} \\
&\ge 2NT\left(\frac{1}{\alpha} - 1\right)\mu\sqrt{\hat{D}}. \qquad \text{(A.23)}
\end{aligned}$$

From (A.19), (A.20), (A.22), and (A.23), we have

$$\begin{aligned}
&\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q_{jl}^i(X_l(t) - X_j(t)) \\
&\le -u\frac{1}{\alpha}Tmc\sqrt{\hat{D}} + 4M(T+\delta)^2 + NB^2M^2T^2 \\
&\quad + N\frac{\delta}{\alpha}\mu\sqrt{\hat{D}} - 2NT\left(\frac{1}{\alpha} - 1\right)\mu\sqrt{\hat{D}} \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right)T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\}. \qquad \text{(A.24)}
\end{aligned}$$

Let $\hat{\mathcal{F}}$ be the set of all queues j for which

$$m\max_{l\in\mathcal{R}_j}(X_j(t) - X_l(t)) > \frac{4\alpha}{(\alpha-1)}MT. \qquad \text{(A.24a)}$$

Then, from (A.16), we have

$$\begin{aligned}
&\sum_{j=1}^{B} Q_j X_j(t) \\
&\le \sum_{j=1}^{B} X_j(t)b_j + T\sum_{j\in\hat{\mathcal{F}}}\max_{k\in\mathcal{R}_j}\{X_j(t) - X_k(t)\}\sum_{l\in\mathcal{R}_j} f_{jl} \\
&\quad + B\sum_{i=1}^{B} a_i\frac{4\alpha}{\alpha-1}MT. \qquad \text{(A.25)}
\end{aligned}$$

Note that from (A.24a), if $j \in \hat{\mathcal{F}}$, then all i's such that $j \in \mathcal{B}_i$ belong to $\mathcal{F}$. Based on this fact, in the following, we verify that

$$\sum_{j\in\hat{\mathcal{F}}} \max_{k\in\mathcal{R}_j}\{X_j(t)-X_k(t)\}\sum_{l\in\mathcal{R}_j} f_{jl} \le \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right) T \max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}. \quad \text{(A.26)}$$

Note first that

$$\sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u_j^i\right) T \max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\} \ge T\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{B}_i} u_j^i\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\} \quad \text{(A.27)}$$

and from the fact that if $j \in \hat{\mathcal{F}}$, then all i's such that $j \in \mathcal{B}_i$ belong to $\mathcal{F}$, we have

$$T\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{B}_i} u_j^i\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\} \ge T\sum_{i\in\hat{\mathcal{F}}}\sum_{j\in\mathcal{B}_i} u_j^i\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}. \quad \text{(A.28)}$$

From (A.27), (A.28), and (3.2), the relation (A.26) follows. By adding the inequalities (A.24)–(A.26) and (A.15), and after the simplifications, we get

$$\begin{aligned}\sum_{j=1}^{B}(X_j(t+T)-X_j(t))X_j(t) &\le -u\frac{1}{\alpha}Tmc\sqrt{\hat{D}} + 4M(T+\delta)^2 + NB^2M^2T^2 \\ &+ N\frac{\delta}{\alpha}\mu\sqrt{\hat{D}} - 2NT\left(\frac{1}{\alpha}-1\right)\mu\sqrt{\hat{D}} \\ &+ \sqrt{\hat{D}}\sum_{j=1}^{B} b_j + B\sum_{i=1}^{B} a_i\frac{a\alpha}{\alpha-1}MT. \end{aligned} \quad \text{(A.28a)}$$

By replacing in (A.13) from (A.14) and (A.28a), we get

$$\begin{aligned}\sum_{j=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) &\le c_1T^2 - c_2\frac{1}{\alpha}T\sqrt{\hat{D}} + c_3\sqrt{\hat{D}} \\ &+ c_4T\sqrt{\hat{D}}\left(1-\frac{1}{\alpha}\right) \\ &+ c_5\frac{4\alpha}{\alpha-1}T \end{aligned} \quad \text{(A.29)}$$

where $c_1, \cdots, c_5$ are positive constants which depend only on the system topology and the arrival and service parameters. Select $\alpha > 1$ such that $c_4 - (c_2 + c_4/\alpha) = -\zeta < 0$, and let $T = (\hat{D})^{1/4}$. Then (A.29) becomes

$$\begin{aligned}\sum_{j=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) &\le c_1(\hat{D})^{1/2} - \zeta(\hat{D})^{3/4} + c_3(\hat{D})^{1/2} \\ &+ c_5\frac{4\alpha}{\alpha-1}(\hat{D})^{1/4}. \end{aligned} \quad \text{(A.30)}$$

It is clear that if $\hat{D}$ is large enough, the right side of (A.30) becomes strictly negative and the lemma follows. ◊

APPENDIX B

The proof of Lemma 5.1 is given in this Appendix after some intermediate results. Define

$$\alpha_i^T = g\left(\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\} + 2MT\right).$$

Lemma B.1: If for a server i, we have

$$\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\} > \frac{4\alpha_i^T}{(\alpha_i^T-1)}MT \quad \text{(B.1)}$$

then i will switch at most once in the time interval $(t, t+T)$ and

$$\begin{aligned}&\sum_{j\in\mathcal{B}_i}\sum_{l\in\mathcal{R}_j} Q_{jl}^i(X_j(t)-X_l(t)) \\ &\ge \frac{1}{\alpha_i(t)}(T-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\} \\ &\quad - 4M(T+\delta)^2 \end{aligned} \quad \text{(B.2)}$$

where Q_{jl}^i is the amount of work served by server i and transferred from buffer j to buffer l during the time interval $(t, t+T)$.

Proof: The proof follows the same steps as with Lemma A.1, except for the following differences. Inequality (A.4) holds with α replaced by $\alpha_i(t)$ on the left side and with α_i^T on the right side. Inequalities (A.5a), (A.7), (A.9), and (A.12a) hold with α replaced by $\alpha_i(t)$. ◊

Proof of Lemma 5.1: From (A.13)–(A.15), we get

$$\begin{aligned}\sum_{j=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) &= \sum_{j=1}^{B}Q_jX_j(t) \\ &+ \sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N}Q_{jl}^i(X_l(t)-X_j(t)) \\ &+ BM^2T^2. \end{aligned} \quad \text{(B.3)}$$

Also

$$\sum_{j=1}^{B}Q_jX_j(t) \le T\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j} f_{jl}\max_{k\in\mathcal{R}_j}(X_j(t)-X_k(t)) + \sum_{j=1}^{B}X_j(t)b_j. \quad \text{(B.4)}$$

In the following, we upper bound the second term on the right side of (B.3). Note first that

$$Q^i_{jl}(X_j(t) - X_l(t)) \geq -M^2T^2, \qquad i = 1, \cdots, N, \quad j, l = 1, \cdots, B. \tag{B.5}$$

By interchanging the order of the summation, we have

$$\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q^i_{jl}(X_l(t) - X_j(t)) = -\sum_{i=1}^{N}\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j} Q^i_{jl}(X_j(t) - X_l(t)). \tag{B.6}$$

Let $\mathcal{F}$ be the set of servers for which (B.1) holds. Replacing in (B.6) from (A.2) and (B.5), we get

$$\begin{aligned}
&\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q^i_{jl}(X_l(t) - X_j(t)) \\
&\leq -\sum_{i\in\mathcal{F}} \frac{1}{\alpha_i(t)}(T-\delta)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + 4M(T+\delta)^2 + NB^2M^2T^2 \\
&\leq -\sum_{i\in\mathcal{F}}\left(1 - \sum_{j\in\mathcal{B}_i} u^i_j\right)\frac{1}{\alpha_i(t)} T\max_{j\in\mathcal{B}_i} \\
&\quad \cdot\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + 4M(T+\delta)^2 + NB^2M^2T^2 \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u^i_j\right) T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad + \sum_{i\in\mathcal{F}}\frac{\delta}{\alpha_i(t)}\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u^i_j\right) T\left(\frac{1}{\alpha_i(t)} - 1\right)\max_{j\in\mathcal{B}_i} \\
&\quad \cdot\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\}.
\end{aligned} \tag{B.7}$$

To simplify the right side of inequality (B.7), we upper bound the terms appearing in it in the following. Based on the definition of u and m in (A.19a), the nonincreasingness of $g(\,)$, and the definition of $\alpha_i(t)$, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\left(1 - \sum_{j\in\mathcal{B}_i} u^i_j\right)\frac{1}{\alpha_i(t)} T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\geq u\frac{1}{g(m\max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\})} Tm \\
&\quad \times \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}.
\end{aligned} \tag{B.8}$$

Since $\alpha_i(t) > 1$, $i = 1, \cdots, N$, $t > 0$, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\frac{\delta}{\alpha_i(t)}\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\qquad \leq N\delta\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}.
\end{aligned} \tag{B.9}$$

Similarly, since $1/\alpha_i(t) < 1$, $i = 1, \cdots, N$, $t > 0$, $\sum_{j\in\mathcal{B}_i} u^i_j \leq 1$, and for μ as defined in (A.19a), we have (B.10), as found at the bottom of the page. From the definition of $\mathcal{F}$ and (B.1), (B.10) becomes

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u^i_j\right) T\left(\frac{1}{\alpha_i(t)} - 1\right)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_i}\{X_j(t) - X_l(t)\}\right\} \\
&\geq NT\left(\frac{1}{g((2\alpha_i^T/\alpha_i^T - 1)MT)} - 1\right)\mu \\
&\quad \cdot \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} \\
&\geq NT\left(\frac{1}{g(2MT)} - 1\right)\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}.
\end{aligned} \tag{B.11}$$

From (B.7)–(B.9) and (B.11), we have

$$\begin{aligned}
&\sum_{j=1}^{B}\sum_{l\in\mathcal{R}_j}\sum_{i=1}^{N} Q^i_{jl}(X_l(t) - X_j(t)) \\
&\leq -u\frac{1}{g(m\max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\})} Tm \\
&\quad \times \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} + 4M(T+\delta)^2 \\
&\quad + NB^2M^2T^2 + N\delta\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} \\
&\quad - NT\left(\frac{1}{g(2MT)} - 1\right)\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} \\
&\quad - \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u^i_j\right) T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\}.
\end{aligned} \tag{B.12}$$

$$\begin{aligned}
&\sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i} u^i_j\right) T\left(\frac{1}{\alpha_i(t)} - 1\right)\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\right\} \\
&\geq NT\left(\frac{1}{\max_{i\in\mathcal{F}}\alpha_i(t)} - 1\right)2\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\} \\
&\geq NT\left(\frac{1}{g(\min_{i\in\mathcal{F}}\max_{j\in\mathcal{B}_i}\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}\})} - 1\right)\mu \max_{j=1,\cdots,B, l\in\mathcal{R}_j}\{X_j(t) - X_l(t)\}.
\end{aligned} \tag{B.10}$$

Let $\hat{\mathcal{F}}$ be the set of all queues j for which

$$\max_{l\in\mathcal{R}_j}(X_j(t)-X_l(t)) > \frac{2\alpha_i^T}{(\alpha_i^T-1)}MT. \tag{B.13}$$

Then from (B.4), we have

$$\begin{aligned}\sum_{j=1}^{B}Q_jX_j(t) &\le \sum_{j=1}^{B}X_j(t)b_j \\ &\quad + T\sum_{j\in\mathcal{F}}\max_{k\in\mathcal{R}_j}\{X_j(t)-X_k(t)\}\sum_{l\in\mathcal{R}_j}f_{jl} \\ &\quad + B\sum_{i=1}^{B}a_i\frac{2\alpha_i^T}{\alpha_i^T-1}MT. \end{aligned}\tag{B.14}$$

Notice that

$$\begin{aligned}&B\sum_{i=1}^{B}a_i\frac{2\alpha_i^T}{\alpha_i^T-1}MT \\ &\le 2BMT\sum_{i=1}^{B}a_i + 2BMT\sum_{i=1}^{B}a_i\frac{1}{\min_{i=1,\cdots,B}\alpha_i^T-1} \\ &\le 2BMT\sum_{i=1}^{B}a_i + 2BMT\sum_{i=1}^{B}a_i \\ &\quad\cdot\frac{1}{g(\mu\max_{j=1,\cdots,B,l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\})-1}.\end{aligned}\tag{B.15}$$

Defining $\hat{D}=\sum_{j=1}^{B}X_j^2(t)$, we have

$$\sum_{j=1}^{B}X_j(t)b_j \le \sum_{j=1}^{B}B_j\sqrt{\hat{D}}. \tag{B.16}$$

From (B.15) and (B.16), relation (B.14) becomes

$$\begin{aligned}&\sum_{j=1}^{B}Q_jX_j(t) \\ &\le \sum_{j=1}^{B}b_j\sqrt{\hat{D}} + 2BMT\sum_{i=1}^{B}A_i + 2BMT\sum_{i=1}^{B}a_i \\ &\quad\cdot\frac{1}{g(\mu\max_{j=1,\cdots,B,l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}+2MT)-1} \\ &\quad + T\sum_{j\in\hat{\mathcal{F}}}\max_{k\in\mathcal{R}_j}\{X_j(t)-X_k(t)\}\sum_{l\in\mathcal{R}_j}f_{jl}.\end{aligned}\tag{B.17}$$

Note that from (B.13), if $j\in\hat{\mathcal{F}}$, then all i's such that $j\in\mathcal{B}_i$ belong to $\mathcal{F}$. Based on this fact, and arguing similarly as in the proof of Lemma 4.1 [relations (A.27), (A.28)], it easily follows that

$$\begin{aligned}&T\sum_{j\in\hat{\mathcal{F}}}\max_{k\in\mathcal{R}_j}\{X_j(t)-X_k(t)\}\sum_{l\in\mathcal{R}_j}f_{jl} \\ &\le \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{B}_i}u_j^i\right)T\max_{j\in\mathcal{B}_i}\left\{\mu_{ij}\max_{l\in\mathcal{R}_j}\{X_j(t)-X_l(t)\}\right\}.\end{aligned}\tag{B.18}$$

By replacing in (B.3) from (B.12), (B.17), and using (B.18) and Lemma B.1, we get

$$\begin{aligned}&\sum_{j=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) \\ &\le -u\frac{1}{g(mc\sqrt{\hat{D}})}Tmc\sqrt{\hat{D}} \\ &\quad + 4M(T+\delta)^2 + NB^2M^2T^2 \\ &\quad + N\delta\mu c\sqrt{\hat{D}} - NT\left(\frac{1}{g(2MT)}-1\right)\mu c\sqrt{\hat{D}} \\ &\quad + BM^2T^2 + \sum_{j=1}^{B}b_j\sqrt{\hat{D}} + 2BMT\sum_{i=1}^{B}a_i \\ &\quad + BMT\sum_{i=1}^{B}a_i\frac{1}{g(\mu c\sqrt{\hat{D}}+2MT)-1}.\end{aligned}\tag{B.19}$$

If we select $T=(\hat{D})^{1/4}$, then (B.19) becomes

$$\begin{aligned}&\sum_{J=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) \\ &\le c_1\hat{D}^{1/4} + c_2\sqrt{\hat{D}} - c_3\frac{1}{g(c_4\sqrt{\hat{D}})}\hat{D}^{1/4}\sqrt{\hat{D}} \\ &\quad - c_5\left(\frac{1}{g(2M\hat{D}^{1/2})}-1\right)\hat{D}^{1/4}\sqrt{\hat{D}} \\ &\quad + c_6\hat{D}^{1/4}\frac{\sqrt{\hat{D}}}{\sqrt{\hat{D}}(g(c_7\sqrt{\hat{D}}+c_8\hat{D}^{1/4})-1)}\end{aligned}\tag{B.20}$$

where $c_1,\cdots,c_8$ are constants that depend only on the system parameters. Since $\lim_{x\to\infty}g(x)=1$, there exists $c_9>0$ such that, for $\hat{D}$ sufficiently large,

$$-c_3\frac{1}{g(c_4\sqrt{\hat{D}})} - c_5\left(\frac{1}{g(2M\hat{D}^{1/2})}-1\right) \le -c_9 < 0. \tag{B.21}$$

Also, there exists a c_{10} such that, for $\hat{D}$ sufficiently large,

$$c_1\hat{D}^{1/4} + c_2\sqrt{\hat{D}} \le c_{10}\sqrt{\hat{D}}. \tag{B.22}$$

Hence, (B.20) becomes

$$\begin{aligned}&\sum_{j=1}^{B}X_j^2(t+T)-\sum_{j=1}^{B}X_j^2(t) \\ &\le c_{10}\sqrt{\hat{D}} - c_9\hat{D}^{3/4} + c_6\hat{D}^{1/4}\frac{\hat{D}^{1/2}}{\hat{D}^{1/2}(g(c_7\sqrt{\hat{D}}+c_8\hat{D}^{1/4})-1)} \\ &= c_{10}\sqrt{\hat{D}} + \hat{D}^{3/4}\left(c_6\frac{1}{\sqrt{\hat{D}}(g(c_7\sqrt{\hat{D}}+c_8D_0^{1/5})-1)} - c_{10}\right).\end{aligned}\tag{B.23}$$

Since $\lim_{x\to\infty}x(g(x)-1)=\infty$, we have that for $\hat{D}$ large enough, there exists $c_{11}>0$ such that

$$c_6\frac{1}{\sqrt{\hat{D}}(g(c_7\sqrt{\hat{D}}+c_8D_0^{1/5})-1)} - c_{10} < -c_{11}. \tag{B.24}$$

From (B.23) and (B.24), we have for large $\hat{D}$ that

$$\sum_{j=1}^{B} X_j^2(t) \le c_{10}\sqrt{\hat{D}} - c_{11}\hat{D}^{3/4}. \tag{B.25}$$

It is clear from (5.1) that if $\hat{D}$ is large enough, then the right side of (B.25) becomes strictly negative, and the lemma follows. ◊

ACKNOWLEDGMENT

The author would like to thank the reviewers for the thorough reviews that helped improve the final version of the paper.

REFERENCES

[1] C.-S. Chang, "Stability, queue length, and delay, Part I: Deterministic queueing networks," Tech. Rep. RC 17708, IBM T. J. Watson Res. Cen., Yorktown Heights, NY, 1992.
[2] R. L. Cruz, "A calculus of network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–131, Jan. 1991.
[3] ——, "A calculus of network delay, Part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132–141, Jan. 1991.
[4] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1024–1039, Sept. 1991.
[5] M. G. H. Katevenis, "Fast switching and fair control of congested flow in broadband networks," *IEEE J. Select Areas Commun.*, vol. SAC-5, pp. 1315–1326, Oct. 1987.
[6] P. R. Kumar and T. L. Seidman, "Distributed instabilities and stabilization methods in distributed real time scheduling of manufacturing systems," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 289–298, 1989.
[7] S. H. Lu and P. R. Kumar, "Distributed scheduling based on due dates and buffer priorities," *IEEE Trans. Automat. Contr.*, vol. 35, pp. 289–298, 1991.
[8] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, 1993.
[9] S. S. Panwar, T. K. Philips, and M. S. Chen, "Golden ratio scheduling for flow control with low buffer requirements," *IEEE Trans. Commun.*, vol. 40, pp. 765–772, Apr. 1992.
[10] J. Perkins and P. R. Kumar, "Stable, distributed, real-time scheduling of flexible manufacturing assembly/disassembly systems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 139–148, Feb. 1989.
[11] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1936–1948, Dec. 1992.
[12] ——, "Throughput properties of a queueing network with distributed dynamic routing and flow control," *Advances Applied Probability*, March, 1994.
[13] O. Yaron and M. Sidi, "Calculating performance bounds in communication networks," in *IEEE INFOCOM*, 1993, pp. 539–546.
[14] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switched networks," presented at SIGCOM'90, Philadelphia, PA, 1990.

Leandros Tassiulas (S'89–M'82) was born in 1965 in Katerini, Greece. He received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece in 1987 and the M.S. and Ph.D. degrees, also in electrical engineering, from the University of Maryland, College Park, in 1989 and 1991, respectively.

Since 1991, he has been an Assistant Professor in the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY. His research interests include computer and communication networks with an emphasis on wireless communications and high-speed networks, the control and optimization of stochastic systems, and parallel and distributed processing.

Dr. Tassiulas coauthored a paper that received the INFOCOM '94 Best Paper Award.