Cut-Through Switching, Pipelining, and Scheduling for Network Evacuation

Leandros Tassiulas

Abstract— **A general model of a virtual circuit network consisting of a number of servers and a number of traffic classes is considered. A traffic class is identified by the sequence of servers that should be visited and the corresponding service rates before a message (customer) of the class leaves the network. The following cases are distinguished: 1) the messages need nonpreemptive service; 2) the service of a message can be preempted at any time; 3) pipelining of the service in a sequence of servers is allowed; and 4) pipelining is not allowed. All of these cases arise in different transmission switching techniques and scheduling schemes. A fluid model that emerges when both preemption and pipelining are allowed is considered. Scheduling schemes in the fluid model are compared with corresponding ones in the network with nonpreemptive service and no pipelining. The problem of evacuating the network from an initial backlog without further arrival is identified in the fluid model. Based on that, a policy with nearly optimal evacuation time is identified for the store-and-forward case. Finally, scheduling with deadlines is considered and it is shown that in the fluid model, the evacuation problem is equivalent to a linear programming problem. The evacuation times under different work-conserving policies are considered in specific examples.**

I. Introduction

ONE of the goals of this work is to distinguish and classify the different transmission techniques in communication networks and to identify models that capture the specific characteristics of each technique. Store-and-forward (SF) service is typical in a classical packet-switched network. Nonpreemptive service is provided to the messages since only one message at a time can be transmitted through a link and the transmission cannot be interrupted after it starts. Furthermore, the transmission of a message through a link cannot start until the whole message is residing at the origin node of the link. That is, the message cannot undergo transmission through more than one link at a time.

This is not the case if cut-through switching is employed [8]. In such architecture, a message may start its transmission through a link as soon as the beginning of the message arrives at the origin node of the link and while the rest of the message is under transmission through the upstream link. That is, pipelining of the transmissions is possible. The worm-hole routing technique [2], considered for the interconnection networks in parallel computer architectures, is based on this principle. Another issue is whether a message transmission should proceed uninterrupted, or if the message consists of smaller units, the transmission of which can be interleaved with other messages. In the former case, nonpreemptive service is required, while the latter is the case of preemptive service. In the limit, where infinitesimal portions of the messages can be interleaved, the system can be viewed as if different messages are served simultaneously by the server using fractions of its service capacity. The assumption of divisible messages becomes more plausible in networks with small packet sizes (i.e., asynchronous transfer mode (ATM) networks). The consideration of preemptive scheduling is useful sometimes, even when preemptive scheduling is not allowed, since it may enhance the understanding of the operation of the system and facilitate the design of efficient schemes that can be translated then to nonpreemptive analogs. This is the case of fair queueing [1] and the generalized processor sharing schemes [9], [10], considered for network congestion control. If both cut-through switching and preemptive service are employed, then a fluid model of the network naturally emerges. In this paper, such a fluid model is defined and a comparison is done between scheduling in the fluid model and scheduling when neither preemption nor pipelining is allowed. The evacuation problem is studied then under the different assumptions of preemptive or nonpreemptive service and pipelining or SF service. In the evacuation problem, there are a number of messages originally stored in the network, no exogenous arrivals exist, and the objective is to complete the service of the messages. This problem arises naturally in applications like file transfers and batch processing. In addition, it is an important step toward the study of the system in continual operation with exogenous arrivals. The minimum evacuation time problem is considered first on the fluid model and the optimal policy is specified. Based on that, a policy for the SF case with asymptotically optimal evacuation time is obtained. The evacuation time of the SF policy may exceed the optimal by at most a constant, independent of the evacuation time. The evacuation is studied next for the case where different parts of the traffic have different target times to complete their service. In the fluid model, it is shown that this problem can be formulated as a linear programming problem. Special topologies are considered where the problem has a more direct solution.

Manuscript received December 19, 1995; revised July 14, 1997; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Mitra. This work was supported in part by National Science Foundation under Grant NCR-9406415.

The author is with the Electrical Engineering Department and the Institute For Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail leandros@eng.umd.edu).

Publisher Item Identifier S 1063-6692(99)01819-1.

II. The Network Model

A network of the following type is considered. There are M servers $\{1, 2, \cdots, M\}$. A customer i needs to receive service

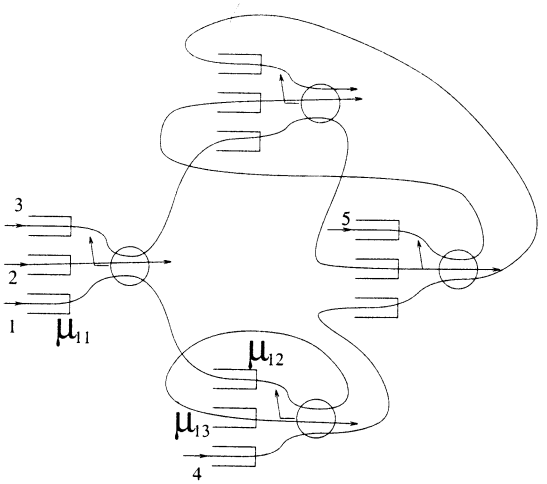

Fig. 1. A network with four servers and five customer classes is depicted.

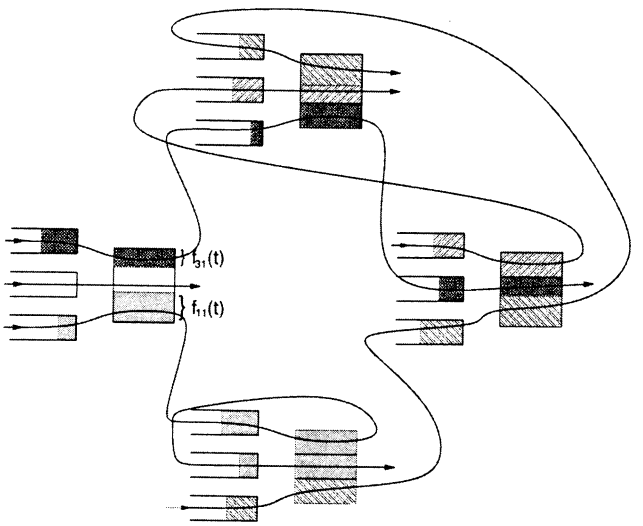

Fig. 2. The fluid model of the network of Fig. 1 is depicted.

by the servers in a certain sequence. The service sequence and the service time may change from customer to customer.

L traffic classes are distinguished. All customers of class l are visiting the servers in the sequence $s_1^l, s_2^l, \cdots, s_{M_l}^l$, where M_l is the number of service stages for class l. Among customers of the same class, the service discipline is first-come-first-serve. An example of the network is depicted in Fig. 1. The service times are determined as follows. Each customer j is associated with a certain workload w_j. A service rate μ_{lm} in units of work per time unit, is associated with the mth service provided to a class l customer. The service time of the jth customer of class l in stage m is equal to $w_j(\mu_{lm})^{-1}$. Hence the class of the customer and its workload specify the service stages it has to go through and the service time at each stage, an arbitrary number of times in any arbitrary sequence. Also, it is possible that the same customer has different service times at different visits to the same server, since different service rates may be associated with each visit.

Another way to view the distinction of different customer classes is to consider that a class l consists of all customers that follow the same service sequence, and the service times $\sigma_m^i, \sigma_n^i, \sigma_m^j, \sigma_n^j$ of any two customers i and j of class l in stages m and n are such that $\sigma_m^i/\sigma_n^i = \sigma_m^j/\sigma_n^j$. This assumption is natural in communication networks where the customers are messages, the workload is the amount of information, and the service times are the transmission times. In this case, the ratio of the transmission times of message in two links is equal to the ratio of the capacities of the links, and independent of the information length of the message. A communication network model where traffic streams undergo several multiplexing stages as they are routed through the network is included in the formulation above. The service stages correspond to different multiplexing stages and the servers to transmission links. In this case, a customer will be served once at most by a server, and each server is associated with a specific service stage. Furthermore, the service rate is associated with the server and is the same for all classes served by the server. Hence, a class in this case is associated only with the route of the customer, while in the general case we may have classes, the customers of which follow the same route, and differ only in the service rates at the different stages. Several different assumptions can be made about the type of service. This is referred as the SF service. Either assumption can be relaxed in certain cases. If both assumptions are relaxed, that is pipelining is allowed as well as preemption, we end up with a fluid model of the network, to be specified in detail next.

III. The Fluid Model

Some notation is introduced, to be used in the description of the fluid model. Let $a_n^l, w_{ln}, n = 1, 2, \cdots$ be the arrival times and workload lengths, respectively, of the nth customer of class l. Note that a_n^l is the time that the customer has completely arrived to the system. The system starts at time zero and $a_n^l \geq 0$. If there are customers in the system at time $t = 0$, they are incorporated in the arrival sequence with arrival times equal to zero. It is assumed that there are no customers initially in the system at intermediate service stages. This is without loss of generality, since if in actuality there are such customers, it can be assumed that they are of a different class that starts service at that stage. Also, it is natural to assume that the message lengths cannot be arbitrarily small but they are lower bounded by a positive number. Let $A_l(t)$ be the total workload of stream l that arrived in the system by time t. Hence at time a_n^l

$$A_l(a_n^l) = \sum_{j=1}^{n} w_{lj}.$$

The customers may arrive in the system either instantaneously, in which case $A_l(t)$ will have jumps at the arrival times, or continuously, in which case $A_l(t)$ is considered to be differentiable during the arrival period of a customer, with derivative equal to the instantaneous arrival rate. Let $D_{lm}(t)$ be the total class l workload that has gone through the mth stage of service by time t. This includes the served part of the customer who is under service at stage m at time t. Let $w_{lm}(t)$ be the total workload of class l customers who are in the system, but have not gone through service at stage m at time t. It includes the remaining workload of the customer receiving service at stage m at time t. The workload of class l that has gone through stage $m-1$ but not through stage m by

time t is denoted by $x_{lm}(t)$ and will be called the backlog of class l at stage m. Note that both the workload and the backlog of a class are defined in terms of the remaining service time (normalized by the corresponding service rate of the server) and not in terms of the remaining number of customers. Clearly it holds

$$x_{lm}(t) = w_{lm}(t) - w_{l(m-1)}(t) \tag{1}$$

and

$$D_{lm}(t) = A_l(t) - w_{lm}(t).$$

The study of some scheduling problems is facilitated by the consideration of the following fluid model. A server i can allocate portions of its capacity to different customers simultaneously, and in general can allocate some fraction of its capacity to the mth stage of class l customers if $s_m^l = i$. At time instant t this fraction is denoted by $f_{lm}(t)$. It holds

$$\sum_{l,m:\ s_m^l = i} f_{lm}(t) \leq 1. \tag{2}$$

The inequality in (2) is strict whenever some portion of the server capacity is not used, something that corresponds to partial idling. The service rate at time t for class l at stage m is equal to $\mu_{lm} f_{lm}(t)$. If the backlog $x_{lm}(t)$ is greater than zero, then the departure rate $\dot{D}_{lm}(t) = dD_{lm}(t)/dt$ is equal to $\mu_{lm} f_{lm}(t)$. Otherwise it may be less. An open-loop schedule $\{\{f_{lm}(t), t \geq 0\}, l = 1, \cdots, L, m = 1, \cdots, M_l\}$ specifies the service fractions at each stage at all times. The workload under some schedule evolves according to the following equation:

$$\dot{w}_{lm}(t) = \begin{cases} \dot{A}_l(t) - f_{lm}(t)\mu_{lm}, \\ \quad \text{if } x_{lm}(t) > 0 \\ \dot{A}_l(t) - \min\{f_{lm}(t)\mu_{lm}, \dot{A}_l(t) - \dot{w}_{l(m-1)}(t)\}, \\ \quad \text{if } x_{lm}(t) = 0, m > 1 \\ \dot{A}_l(t) - \min\{f_{lm}(t)\mu_{lm}, \dot{A}_l(t)\}, \\ \quad \text{if } x_{lm}(t) = 0, m = 1. \end{cases} \tag{3}$$

At the arrival time t of an instantaneously arriving customer, $\dot{A}_l(t)$ is a delta function with mass equal to the workload of the arriving customer. Equation (3) implies that given an open-loop schedule, the evolution of the workload $w_{lm}(t)$ of stream l in stage m is independent of the other arrival streams sharing server s_m^l. In open-loop schedules, the sample paths of service fractions

$$\{\{f_{lm}(t), t \geq 0\}, l = 1, \cdots, L, m = 1, \cdots, M_l\}$$

are prespecified and remain the same for all arrival processes. In closed-loop schedules, the service fractions are determined based on the state of the network. Therefore, a closed-loop schedule can be viewed as a mapping which maps an open-loop schedule to each arrival sample path. Given an arrival sample path and an open-loop schedule, (3) determines the evolution of the network. Clearly, the fluid model provides higher flexibility in scheduling than the SF one, since any SF schedule is a fluid schedule as well, while the opposite is not true. In fact, the SF schedules can be defined formally as those where $f_{ln}(t) \in \{0, 1\}$ and the allocation of a specific server may only change at the service completion times. In the Sections IV and V it is shown that for every fluid schedule, there is a schedule in the SF system that follows it closely. The results hold for every arrival and service fraction sample paths. For the validity of these results, it does not matter whether the service schedule is open-loop and remains fixed for every arrival process or not. Note that the construction of the store and forward schedule from the fluid one is nonanticipative in the sense that the allocation at t depends only on the evolution of the system up to time t. Hence the construction can go through both for open- and closed-loop schedules.

IV. SF Schedules From Fluid Schedules: The Single Server Case

Consider a single server with L arrival streams. After service the work leaves the system. We keep the notation introduced earlier for the network, without the indices that correspond to the stages though, since there is only one service stage. Recall that in SF service, pipelining is not allowed, that is a packet may be selected for service only if it has completely arrived. Let $S = \{\{f_l(t), t \geq 0\}, l = 1, \cdots, L\}$ be an arbitrary fluid schedule. Consider the SF schedule $\hat{S}$ that is derived from S as follows. The server is tracking the fluid model, the departure processes $D_l(t), l = 1, \cdots, L$ of which evolve as follows:

$$\dot{D}_l(t) = \begin{cases} \mu_l f_l(t), & \text{if } D_l(t) < A_l(t) \\ \min\{\mu_l f_l(t), \dot{A}_l(t)\}, & \text{if } D_l(t) = A_l(t). \end{cases} \tag{4}$$

At a service completion instant ρ of the nonpreemptive discipline, the next customer to be served is determined as follows. Let $\hat{D}_l(t)$ be the departure process of class l under the SF server. It includes the workload of the stream l customers that completed service, and if a stream l customer is receiving service at time t, the portion of the workload of that customer that has already being served. The classes l for which the fluid server is running ahead of the SF server, that is the class l such that

$$D_l(\rho) - \hat{D}_l(\rho) > 0 \tag{5}$$

are identified. If there is no such class, then the SF server may serve arbitrarily. Otherwise, for every class l that (5) holds, consider all the messages that have not been served by the SF server yet have been served or have started service in the fluid server at ρ. Let ρ_l be the service initiation time of the first class l message that started service before ρ under the fluid model. Then the SF server selects for service the class for which ρ_l is smallest. Note that in the case of a single server, it is superfluous to consider different service rates for each class since these can be incorporated in appropriate normalization of the service times. In the multiserver case though, if the service rate for a class changes from server-to-server, it is necessary to consider explicitly the service rates. Since the result for the single server needs to be applied in the network case, it is stated with the consideration of the service rates.

Let W_l be the maximum message size of class l. Let $\hat{t}_i^l, t_i^l$ be the departure times of the ith message of the lth class under schedules $\hat{S}$ and S, respectively. The SF schedule $\hat{S}$ tracks the fluid schedule S in the sense stated in the following theorem.

Theorem 1: Let S and $\hat{S}$ be a fluid schedule and the corresponding SF schedule. For the departure processes $D_l(t)$ and $\hat{D}_l(t)$ of the fluid and SF schedules, respectively, it holds

$$\hat{D}_l(t) - D_l(t) \geq -\mu_l \max_{j=1,\cdots,L} \left\{ \frac{W_j}{\mu_j} \right\} - 2\mu_l \sum_{j=1}^{L} \mu_j^{-1} W_j, \quad t \geq 0, l = 1, \cdots, L. \quad (6)$$

Furthermore, the departure times satisfy

$$\hat{t}_i^l - t_i^l \leq \max_{j=1,\cdots,L} \left\{ \frac{W_j}{\mu_j} \right\} + 2 \sum_{m=1}^{L} \mu_m^{-1} W_m + \mu_l^{-1} W_l, \quad l = 1, \cdots, L \quad (7)$$

Proof: Let $(t_n, u_n), n = 1, 2, \cdots$ be the collection of all intervals such that

$$\min_{l=1,\cdots,L} \{\hat{D}_l(t) - D_l(t) + W_l\} \geq 0, \qquad t \in [u_n, t_{n+1}] \quad (8)$$

and

$$\min_{l=1,\cdots,L} \{\hat{D}_l(t) - D_l(t) + W_l\} < 0, \qquad t \in (t_n, u_n). \quad (9)$$

If there is a time τ such that the left part of (8) and (9) is either negative for all $t > \tau$ or positive for all $t > \tau$, then the number of intervals (t_n, u_n) is finite. Note that between times t_n and u_{n+1}, at least one message is served completely, for instance a message of a class l_0 such that $\hat{D}_{l_0}(t) - D_{l_0}(t) < 0$ for some $t \in (t_n, u_n)$. Therefore $t_{n+1} - t_n$ is lower bounded by a positive number, since the length of the message is lower bounded. Hence, the sequence of $t_n s$ has no accumulation points. It is clear that (6) holds for $t \in [u_n, t_{n+1}]$. Therefore it is enough to prove it for $t \in (t_n, u_n)$.

Note that by the way $\hat{S}$ is defined, after a service completion at some time $t \in (t_n, u_n)$ the server is allocated to a message of a class l such that $\hat{D}_l(t) - D_l(t) < 0$. A consequence of this fact is that

$$\hat{D}_l(t) - D_l(t) \leq W_l, \qquad l \in C(t) \quad (10)$$

where $C(t)$ is the collection of classes l such that $\hat{D}_l(\tau) - D_l(\tau)$ is negative for some τ in the time interval (t_n, t). In other words, if the SF service of a class l is running behind the fluid service at any time $t \in (t_n, u_n)$, then it will never run ahead of the fluid service by more than W_l at any time instance during the interval (t, u_n) because the server is always allocated to classes for which $\hat{D}_l(t) - D_l(t) < 0$ for $t \in (t_n, u_n)$.

Let $T_l(t)$ be the time spent serving class l in the SF system during the time interval (t_n, t). Let t_n^0 be the time of the first server reallocation after t_n. Assume first that $t_n^0 < t$. During the period (t_n^0, t), only classes $l \in C(t)$ were served. Furthermore, there was always a customer to serve and no idling was required because of (9). Therefore, it follows

$$\sum_{l \in C(t)} T_l(t) = (t - t_n^0) \geq t - t_n - \tau \quad (11)$$

where τ is the maximum possible value of $t_n^0 - t_n$, that is, the maximum possible service time over all classes

$$\tau = \max_{l=1,\cdots,L} \{W_l/\mu_l\}.$$

If $t_n^0 \geq t$, then the same packet was served by the SF server in the interval (t_n, t), therefore, $t - t_n < \tau$ and (11) follows easily. Also it clearly holds

$$\sum_{l \in C(t)} \int_{t_n}^{t} f_l(\tau)\, d\tau \leq t - t_n. \quad (12)$$

Note that for any $t \in (t_n, u_n)$ we have

$$D_l(t) \leq D_l(t_n) + \int_{t_n}^{t} \mu_l f_l(\tau)\, d\tau \quad (13)$$

$$\hat{D}_l(t) = \hat{D}_l(t_n) + \mu_l T_l(t). \quad (14)$$

From (13), (14) and using the fact that at t_n we have $\hat{D}_l(t_n) - D_l(t_n) \geq -W_l$, it follows

$$\sum_{l \in C(t)} \mu_l^{-1}(\hat{D}_l(t) - D_l(t)) \geq \sum_{l \in C(t)} T_l(t) - \sum_{l \in C(t)} \int_{t_n}^{t} f_l(\tau)\, d\tau - \sum_{l \in C(t)} \mu_l^{-1} W_l. \quad (15)$$

Replacing from (11) and (12)–(15), we get

$$\sum_{l \in C(t)} \mu_l^{-1}(\hat{D}_l(t) - D_l(t)) \geq -\tau - \sum_{l \in C(t)} \mu_l^{-1} W_l. \quad (16)$$

Let $\hat{C}(t)$ be the set of classes l such that $\hat{D}_l(t) - D_l(t) < 0$. Clearly $\hat{C}(t) \subseteq C(t)$. From (16)

$$\sum_{l \in \hat{C}(t)} \mu_l^{-1}(\hat{D}_l(t) - D_l(t)) \geq -\tau - \sum_{l \in C(t)} \mu_l^{-1} W_l - \sum_{l \in (C(t) - \hat{C}(t))} \mu_l^{-1}(\hat{D}_l(t) - D_l(t)) \quad (17)$$

and from (10)

$$\sum_{l \in \hat{C}(t)} \mu_l^{-1}(\hat{D}_l(t) - D_l(t)) \geq -\tau - 2 \sum_{l=1}^{L} \mu_l^{-1} W_l. \quad (18)$$

Note that since all the terms in the summation in the left side of (18) are negative, from (18) it follows that

$$\hat{D}_l(t) - D_l(t) \geq -\mu_l \tau - 2\mu_l \sum_{j=1}^{L} \mu_j^{-1} W_j, \qquad l \in \hat{C}(t)$$

and therefore (6) is proved for $l \in \hat{C}(t)$. For any l that does not belong to $\hat{C}(t)$, (6) clearly holds.

Let $\{t_i^l\}_{i=1}^{\infty}$ be the times at which the ith customer of class l completes service at the fluid server. Let $\{\hat{t}_i^l\}_{i=1}^{\infty}$ be the times at which the ith customer of class l completes service in the SF server. Let $\tilde{t}_i^l$ be the time at which message i will initiate service in the SF server. If $\tilde{t}_i^l \leq t_i^l$ then clearly (7) holds. If $\tilde{t}_i^l > t_i^l$ then $\hat{D}_l(t) < D_l(t)$ for $t \in (t_i^l, \tilde{t}_i^l]$. In the worst case message i will start service after all classes $m \in \hat{C}(t_i^l)$ serve at most $(D_m(t_i^l) - \hat{D}_m(t_i^l)) + W_m$ amount of work. That implies

$$\tilde{t}_i^l - t_i^l \leq \sum_{m \in \hat{C}(t_i^l)} (\mu_m^{-1}(D_m(t_i^l) - \hat{D}_m(t_i^l)) + \mu_m^{-1} W_m)$$

which together with (18) and the fact that $\hat{t}_i^l - \tilde{t}_i^l \leq \mu_l^{-1} W_l$, imply (7). ◇

Since the arrival processes are identical in the fluid and the SF systems, relation (6) provides the following bound on the difference of the backlogs $X_l(t)$ and $\hat{X}_l(t)$, respectively, in the two systems

$$\hat{X}_l(t) - X_l(t) \leq \mu_l \max_{j=1,\cdots,L} \left\{ \frac{W_j}{\mu_j} \right\} + 2\mu_l \sum_{j=1}^{L} \mu_j^{-1} W_j.$$

The bound is independent of the arrival processes. Related work has been done by Georgiades *et al.* in [4]. In that work they obtained nonpreemptive analogs and corresponding lower bounds for fluid service schedules. Note that the fluid server in [4] was SF while in our study pipelining is allowed.

V. SF Schedules From Fluid Schedules: The Network Case

Following a similar approach as for the single server, given a fluid schedule S we can get a SF schedule $\hat{S}$ which is tracking S and is such that the evolution of the system under $\hat{S}$ is close to that under S. Three systems operating under different schedules will be distinguished in the description of the construction of the SF schedule from the fluid schedule.

The first one is the fluid system, which evolves according to (3) given the schedule S. The corresponding departure processes are denoted by $D_{lm}(t)$. The second one is the system under the SF schedule $\hat{S}$, which is to be constructed. The corresponding departure processes are denoted by $\hat{D}_{lm}(t)$. Finally, there are M emulated fluid servers, one for each network server, which are used to facilitate the construction of schedule $\hat{S}$. The emulated fluid server i is receiving work from all the different streams $\hat{D}_{l(m-1)}(t)$ such that $s_m^l = i$ [by convention $\hat{D}_{l0}(t) = A_l(t)$]. It provides service to those streams following schedule S and lets $\tilde{D}_{lm}(t)$ be the departure processes that correspond to the emulated fluid server.

Following the construction of Section IV, we obtain the SF schedule $\hat{S}$ for server i from the emulated fluid server i. The latter is the one with input and output streams $\hat{D}_{l(m-1)}(t)$ and $\tilde{D}_{lm}(t)$, respectively, where l, m are such that $s_m^l = i$. Since the SF schedule is defined in terms of the departure processes $\tilde{D}_{lm}(t)$, which in turn are defined in terms of the departure processes $\hat{D}_{lm}(t)$ of the SF system, it is important to specify explicitly how this is done without deadlock. Note that the construction in the following is inductive.

Let $r_0 = 0$ be the time at which the system starts and r_n, $n = 1, 2, \cdots$ the sequence of times at which the server allocation at any server may change in the SF system. The service allocation may change either because of a service completion or because of a new arrival in an idle server. Given the service allocation at r_n, the departure processes $\hat{D}_{lm}(t)$ and $\tilde{D}_{lm}(t)$ are well defined for $r_n \leq t \leq r_{n+1}$. We specify in the following how the service allocation is determined at the times r_n. At time $r_0 = 0$ we have $\tilde{D}_{lm}(0) = \hat{D}_{lm}(0)$ and every nonempty server may select a message for service arbitrarily. The next event that may trigger a change of the server allocation in the SF system is either an exogenous arrival(s) in an idle server(s) or a service completion in some of the busy servers. Until the time r_1 that either of these events will happen, the processes $\hat{D}_{lm}(t), \tilde{D}_{lm}(t), 0 \leq t \leq r_1$ are well defined. At time r_1, any server i that is not in the middle of service and for which there are messages waiting will be allocated based on the construction of schedule $\hat{S}$ for the emulated fluid server i. This is possible since $\hat{D}_{lm}(t), \tilde{D}_{lm}(t), 0 \leq t \leq r_1$ are well defined. The operation of the system is well specified until time r_2, where the allocation is determined in the same manner as at r_1. In this way, the SF schedule is determined inductively. Note that the schedule applied to the third system of emulated servers is open loop, irrespective of whether the fluid policy in the first system is open or closed loop.

Theorem 2: Let $D_{lm}(t), \hat{D}_{lm}(t), l = 1, \cdots, L, m = 1, \cdots, M_l$ be the departure processes under the fluid and SF schedules.

It holds that for $l = 1, \cdots, L, m = 1, \cdots, M_l$

$$\hat{D}_{lm}(t) - D_{lm}(t) \geq -\sum_{n=1}^{m} \mu_{ln} \max_{j,k:\, s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} - 2 \sum_{n=1}^{m} \mu_{ln} \sum_{j,k:\, s_k^j = s_n^l} \mu_{jk}^{-1} W_j. \quad (19)$$

The proof of the above theorem is based on the proof of Theorem 1 as well as the following result.

Consider a fluid server with a single arrival stream and service rate $\{\mu(t), t \geq 0\}$ bounded by micrometers. The evolution of the workload, which is identical with the backlog in this case, is specified by (3), appropriately simplified. Consider two versions of the server with (cumulative) arrival processes $A_1(t), A_2(t)$, respectively, and let $D_1(t), D_2(t)$ be the corresponding (cumulative) departure processes.

Lemma 1: If for the cumulative arrival processes $A_1(t), A_2(t)$ it holds

$$A_1(t) \geq A_2(t) - L \quad (20)$$

then for the corresponding cumulative departure processes $D_1(t), D_2(t)$ it holds

$$D_1(t) \geq D_2(t) - L. \quad (21)$$

Proof: Let $Q_i(t), i = 1, 2$ be the queue backlogs. Using a reflection mapping representation (7), they can be written as follows:

$$Q_i(t) = A_i(t) - B(t) - \inf_{0 \leq s \leq t} \{A_i(s) - B(s)\} \wedge 0 \quad (22)$$

where $B(t)$ is the cumulative service process

$$B(t) = \int_0^t \mu(s)\, ds$$

and $a \wedge b$ stands for $\min\{a, b\}$. From (20), it follows that

$$\inf_{0 \leq s \leq t} \{A_1(s) - B(s)\} \geq \inf_{0 \leq s \leq t} \{A_2(s) - B(s)\} - L. \quad (23)$$

From (23), the fact that $D_i(t) = A_i(t) - Q_i(t)$, and after some calculations we get

$$D_1(t) - D_2(t) \geq \inf_{0 \leq s \leq t} \{A_2(s) - B(s)\} - L) \wedge 0 - \inf_{0 \leq s \leq t} \{A_2(s) - B(s)\} \wedge 0. \quad (24)$$

By considering separately the cases of the infimum being negative or nonnegative in (24), (21) follows. ◇

Proof of Theorem 2: The processes $\hat{D}_{lm}(t), \tilde{D}_{lm}(t)$ are well defined from the description of the SF schedule. For stream l we will show that (19) holds at all stages $m = 1, \cdots, M_l$ by induction. For $m = 1$, server s_1^l is receiving the same arrival process $A_l(t)$ both in the SF system and the fluid system. Since $f_{l1}(t)$ is the same in the fluid network and the emulated fluid server, the departure processes $D_{l1}(t)$ and $\tilde{D}_{l1}(t)$ are identical. Note that this is true despite the fact that the other traffic streams besides l, which are seen by server s_1^l, might be different in the fluid and the SF system, since (3) implies that the service received by a certain stream is not affected by the traffic of the other streams served by the server. Since $\hat{D}_{l1}(t)$ is the departure process under the SF schedule derived from the emulated fluid server s_1^l, from Theorem 1 it follows that

$$\hat{D}_{l1}(t) - \tilde{D}_{l1}(t) \geq -\mu_{l1} \max_{j,k:\ s_k^j = s_1^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} - 2\mu_{l1} \sum_{j,k:\ s_k^j = s_1^l} \mu_{jk}^{-1} W_j. \quad (25)$$

Inequality (25), together with the fact that $\tilde{D}_{l1}(t)$ and $D_{l1}(t)$ are identical, implies (19) for $m = 1$

$$\hat{D}_{lm}(t) - D_{lm}(t) \geq -\sum_{n=1}^{m} \mu_{ln} \max_{j,k:\ s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} - 2 \sum_{n=1}^{m} \mu_{ln} \sum_{j,k:\ s_k^j = s_n^l} \mu_{jk}^{-1} W_j. \quad (26)$$

The arrival process seen by the server at the $m+1$ stage of the lth stream is $\hat{D}_{lm}(t)$. Stream l in stage $m+1$ is served by a fluid server with rate $f_{l(m+1)}(t)\mu_{lm}$. Note that by the definition of the fluid model in (3), the rate of the service received by stream l in stage m is independent of the other streams sharing the server. Let $\tilde{D}_{l(m+1)}(t)$ be the departure process of that server when its arrival stream is $\hat{D}_{lm}(t)$. From the assumption that (19) holds for m and Lemma 1 it follows

$$\tilde{D}_{l(m+1)}(t) - D_{l(m+1)}(t) \geq -\sum_{n=1}^{m} \mu_{ln} \max_{j,k:\ s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} - 2 \sum_{n=1}^{m} \mu_{ln} \sum_{j,k:\ s_k^j = s_n^l} \mu_{jk}^{-1} W_j. \quad (27)$$

From Theorem 1 we get

$$\hat{D}_{l(m+1)}(t) - \tilde{D}_{l(m+1)}(t) \geq -\mu_{l(m+1)} \max_{j,k:\ s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} - 2\mu_{l(m+1)} \sum_{j,k:\ s_k^j = s_{(m+1)}^l} \mu_{jk}^{-1} W_j$$

which together with (27) complete the induction step. ◇

Bounds on the difference of the departure times under the fair queueing and processor sharing were obtained by Greenberg and Madras [6].

VI. Optimal and Suboptimal Evacuation

Assume that a number of customers are present in the network at time $t = 0$ and there are no further arrivals. The problem of scheduling until they complete their service requirements at all stages is considered. One quantity of interest in this problem is the time by which the service of all customers at all stages will be completed, called the *evacuation time*. We are interested in the minimum achievable evacuation time and for schedules that achieve it. A schedule that achieves minimum evacuation time and the corresponding evacuation time are obtained in the following for the fluid model. Clearly in the fluid case, the minimum evacuation time is no greater than in the SF case. Finding a SF schedule with minimum evacuation time is a formidable task in general. Based on the fluid schedule and using the technique of the previous section to derive SF schedules, nearly optimal evacuation schedules are derived for the SF case.

Let $\boldsymbol{w} = \{w_{lm}:\ l = 1, \cdots, L, m = 1, \cdots, M_l\}$ be the workload vector at time $t = 0$. Note that $\boldsymbol{w}$ is determined uniquely by the customers in the system at $t = 0$ and their service times. The opposite is not true since there are several different combinations of customers and service times with the same workload vector. The minimum evacuation time in the fluid model depends on the initial condition only through the workload vector and will be denoted as $E(\boldsymbol{w})$. The workload vector $\boldsymbol{w}(t)$ will be referred also as state of the network at time t.

Note that there is an one-to-one correspondence between a backlog vector $\boldsymbol{x}$ and the corresponding workload vector $\boldsymbol{w}$, as it is implied by relation (1). Let A be the matrix such that $\boldsymbol{x} = A\boldsymbol{w}$. A nonnegative vector $\boldsymbol{w}$ is a workload vector if and only if $A\boldsymbol{w} \geq 0$. Let us denote by $\mathcal{W}$ the space of all workload vectors. The following theorem characterizes the feasibility of the transition from one workload vector to another and it is crucial in the characterization of the minimum evacuation time.

Theorem 3: When there are no arrivals, the transition of the network from state $\boldsymbol{w}^1$ at time t_1 to $\boldsymbol{w}^2$ at $t_1 + \tau, \boldsymbol{w}^1, \boldsymbol{w}^2 \in \mathcal{W}$ is possible if and only if $\boldsymbol{w}^1 \geq \boldsymbol{w}^2$ and

$$\max_{i=1,\cdots,M} \left\{ \sum_{l,m:\ s_m^l = i} \frac{w_{lm}^1 - w_{lm}^2}{\mu_{lm}} \right\} \leq \tau. \quad (28)$$

The constant policy

$$f_{lm}(t) = f_{lm} = \frac{w_{lm}^1 - w_{lm}^2}{\mu_{lm}\tau}, \quad l = 1, \cdots, L, m = 1, \cdots, M_l \quad (29)$$

achieves the transition.

Proof: The necessity of (28) is obvious so we focus on the sufficiency in the following. Note first that because of (28) and since $\boldsymbol{w}^1 \geq \boldsymbol{w}^2$, the policy defined by relation (29) is indeed feasible since

$$\sum_{l,m:\ s_m^l = i} f_{lm} = \frac{1}{\tau} \sum_{l,m:\ s_m^l = i} \frac{w_{lm}^1 - w_{lm}^2}{\mu_{lm}} \leq 1, \quad i = 1, \cdots, M.$$

Note that the service fractions f_{lm}, as they are defined by (29), guarantee that if they are fully utilized by all streams at all stages, then the network will be driven to $\boldsymbol{w}^2$ within time τ. It is enough to show that at no stage service capacity will be underutilized due to nonavailable work. It is shown in the following that

$$\dot{w}_{lm}(t) = (w^2_{lm} - w^1_{lm})\tau^{-1}, \qquad t \in (t_1, t_1 + \tau) \tag{30}$$

from which the theorem follows. This is shown by induction on the service stages for each traffic stream l. Let n be the smallest index for which w^2_{ln} is nonzero and x^1_{ln}, x^2_{ln} are the initial and final backlogs of stream l in stage n, in which case $w^1_{ln} = x^1_{ln}$ and of course $w^2_{ln} = x^2_{ln}$, since there are no arrivals. It can be easily checked that (30) holds for $m \leq n$. Assume now that (30) holds for some arbitrary $m \geq n$. We show that in this case it will hold for $m+1$ as well and the induction step is complete. The following cases are distinguished.

Assume first that $x^1_{l(m+1)} > 0$. Note that $\dot{w}_{l(m+1)}(t) = (w^2_{l(m+1)} - w^1_{l(m+1)})\tau^{-1}$ if $x_{l(m+1)}(t) > 0$. Note also that $\dot{x}_{l(m+1)}(t) = \dot{w}_{l(m+1)}(t) - \dot{w}_{lm}(t)$ and since (30) holds, it can be calculate that $\dot{x}_{l(m+1)}(t) = (x^2_{l(m+1)} - x^1_{l(m+1)})\tau^{-1}$ if $x_{l(m+1)}(t) > 0$. Hence if $x^1_{l(m+1)} > 0$, then $x_{l(m+1)}(t)$ will be positive for all $t \in (t_1, t_1 + \tau)$ and (30) follows for $m+1$.

Assume now that $x^1_{l(m+1)} = 0$. Then

$$w^1_{l(m+1)} = w^1_{lm}. \tag{31}$$

If $x^2_{l(m+1)} = 0$ then

$$w^2_{l(m+1)} = w^2_{lm} \tag{32}$$

therefore $f_{lm}\mu_{lm} = f_{l(m+1)}\mu_{l(m+1)}$ from (29), and from the definition of the fluid model

$$\dot{w}_{l(m+1)}(t) = \dot{w}_{lm}(t). \tag{33}$$

From (31) to (33) and the induction hypothesis, (30) for $m+1$ follows. If $x^2_{l(m+1)} > 0$, then $w^2_{l(m+1)} > w^2_{lm}$ and from (29) and (31) it follows that

$$f_{lm}\mu_{lm} > f_{l(m+1)}\mu_{l(m+1)}. \tag{34}$$

From (34), the definition of the fluid model, and the induction hypothesis, it follows that

$$\dot{w}_{l(m+1)}(t) = f_{l(m+1)}\mu_{l(m+1)}. \tag{35}$$

By replacing $f_{l(m+1)}$ in (35) from (29), (30) follows. ◇

Corollary 1: The minimum evacuation time of the fluid model from some initial state $\boldsymbol{w}$ is equal to

$$E(\boldsymbol{w}) = \max_{i=1,\cdots,M} \left\{ \sum_{l,m:\, s^l_m = i} \frac{w_{lm}}{\mu_{lm}} \right\} \tag{36}$$

and the policy

$$f_{lm}(t) = f_{lm} = \frac{w_{lm}}{\mu_{lm} \displaystyle\sum_{k,n:\, s^k_n = s^l_m} \frac{w_{kn}}{\mu_{kn}}} \tag{37}$$

achieves the evacuation in minimum time.

Proof: From Theorem 3 with $\boldsymbol{w}^1 = \boldsymbol{w}, \boldsymbol{w}^2 = 0$ it follows that the constant policy with

$$\hat{f}_{lm} = \frac{w_{lm}}{\mu_{lm} \displaystyle\max_{i=1,\cdots,M} \left\{ \sum_{k,n:\, s^k_n = i} \frac{w_{kn}}{\mu_{kn}} \right\}}$$

is feasible and achieves the evacuation in minimum time. Note that the policy defined in (37) is feasible as well and satisfies the following

$$\hat{f}_{lm} \leq f_{lm}, \qquad l = 1, \cdots, L; m = 1, \cdots, M_l.$$

It is not hard to show, by arguing on the derivative of the workload as in the proof of Theorem 3, the following intuitive fact. If in a constant policy some, or all, of the service fractions increase without violating the feasibility of the policy, then the evacuation time can only decrease. For the policy defined by (37), though, the evacuation time will remain the same, since it is equal to the minimum for the policy with service fractions $\hat{f}_{lm}$. ◇

Consider now the case where neither preemption nor pipelining is allowed and let $\hat{S}$ be the SF schedule that corresponds to the fluid schedule in (37). The difference between the evacuation time $\hat{E}$ under $\hat{S}$ and $E(\boldsymbol{w})$ is bounded by a constant independent of $E(\boldsymbol{w})$, as we will show in the following. The next theorem generalizes for networks the property (7) that holds for a single node. While this property is not generalizable in networks for arbitrary time-varying fluid schedules, it can be generalized for fluid schedules with constant service fractions.

Theorem 4: Let $t^{lm}_i, \hat{t}^{lm}_i, l = 1, \cdots, L, m = 1, \cdots, M_l$ be the departure times of the ith customer of traffic class l from the mth stage under the minimum evacuation time fluid policy and the corresponding SF one. Then it holds

$$\hat{t}^{lm}_i - t^{lm}_i \leq \sum_{n=1}^{m} \max_{j,k:\, s^j_k = s^l_n} \left\{ \frac{W_j}{\mu_{jk}} \right\} + 2 \sum_{n=1}^{m} \sum_{j,k:\, s^j_k = s^l_n} \mu^{-1}_{jk} W_j + \sum_{n=1}^{m} \mu^{-1}_{ln} W_l. \tag{38}$$

The following result is needed in the proof of the theorem. Consider two streams of identical messages with arrival time sequences $\{t^1_i\}_{i=1}^{\infty}$ and $\{t^2_i\}_{i=1}^{\infty}$, respectively. The streams are served by a constant service rate server. Assume that there is a nondecreasing sequence of nonnegative numbers $\{b_i\}_{i=1}^{\infty}$ such that

$$t^1_i - t^2_i \leq b_i, \qquad i = 1, 2, \cdots \tag{39}$$

Lemma 2: If the arrival times of streams 1 and 2 satisfy (39) then for the departure times $\{\tilde{t}^1_i\}_{i=1}^{\infty}, \{\tilde{t}^2_i\}_{i=1}^{\infty}$ it holds

$$\tilde{t}^1_i - \tilde{t}^2_i \leq b_i, \qquad i = 1, 2, \cdots. \tag{40}$$

Proof: By induction. For $i = 1$, (40) clearly holds. Assume that it holds for $i = m$. Message $m+1$ of stream 2 will start service at or later than the departure time of message m of the same stream. Therefore if $\hat{t}^2_{m+1}$ is the service starting time of that message, from the induction hypothesis

$$\tilde{t}^1_m - \hat{t}^2_{m+1} \leq b_m. \tag{41}$$

Message $m+1$ of stream 1 will start service at time

$$\max\{\tilde{t}^1_m, t^1_{m+1}\}.$$

From the assumptions (39) and (41) and the nondecreasing property of the sequence of b_i's, the lemma follows. ◇

Proof of Theorem 4: By induction. For $m = 1$, (38) holds from Theorem 1. Assume that it holds for m. The arrival process for class l seen by the fluid server at the $m+1$ stage is $\hat{D}_{lm}(t)$. Let $\tilde{D}_{l(m+1)}(t)$ be the departure process of the fluid server that sees the arrival process $\hat{D}_{lm}(t)$. Since (38) holds for m, from the Lemma 6 we have

$$\tilde{t}_i^{l(m+1)} - t_i^{l(m+1)} \leq \sum_{n=1}^{m} \max_{j,k: s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} + 2 \sum_{n=1}^{m} \sum_{j,k:\ s_k^j = s_n^l} \mu_{jk}^{-1} W_j + \sum_{n=1}^{m} \mu_{ln}^{-1} W_l. \tag{42}$$

Furthermore from Theorem 1, relation (7) we have

$$\hat{t}_i^{l(m+1)} - \tilde{t}_i^{l(m+1)} \leq \max_{j,k:\ s_k^j = s^l_{(m+1)}} \left\{ \frac{W_j}{\mu_{jk}} \right\} + 2 \sum_{j,k:\ s_k^j = s^l_{(m+1)}} \mu_{jk}^{-1} W_j + \mu_{l(m+1)}^{-1} W_l. \tag{43}$$

From (42) and (43) the induction step is complete and the theorem follows. ◇

Based on Theorem 4, the evacuation time $\hat{E}$ is bounded as it is stated in the following.

Corollary 2: The evacuation times $E(\boldsymbol{w})$ and $\hat{E}$ for the fluid schedule and the corresponding SF schedule, respectively, satisfy the following relationship:

$$\hat{E} \leq E(\boldsymbol{w}) + \max_{l=1,\cdots,L} \left\{ \sum_{n=1}^{M_l} \max_{j,k:\ s_k^j = s_n^l} \left\{ \frac{W_j}{\mu_{jk}} \right\} + 2 \sum_{n=1}^{M_l} \sum_{j,k:\ s_k^j = s_n^l} \mu_{jk}^{-1} W_j + M_l \mu_l^{-1} W_l \right\}. \tag{44}$$

An interesting class of scheduling policies are the work-conserving policies, defined as those where every server works with its full capacity if there is work at any of the stages served by the server. An upper bound on the evacuation time from some initial state $\boldsymbol{w}$, that holds for every work-conserving policy is

$$U(\boldsymbol{w}) = \sum_{i=1}^{M} \sum_{l,m:\ s_m^l = i} \frac{w_{lm}}{\mu_{lm}}. \tag{45}$$

To see this, note that $U(\boldsymbol{w})$ is the sum over all servers i of the quantity $\Sigma_{l,m:\ s_m^l = i}\ w_{lm}/\mu_{lm}$, which is the total time that server i needs to spend in serving all the work that has to go through it, when it serves in full speed (no idling). Consider a policy in which exactly one server works at full speed at each time instant while all the others idle. Such a policy achieves an evacuation time equal to $U(\boldsymbol{w})$. The fact that $U(\boldsymbol{w})$ is an upper bound on the evacuation time under any work conserving policy follows easily if we observe that as long as the network is nonempty, there will be at least one server with nonzero backlog which will be working full speed since the policy is work-conserving. Note that $U(\boldsymbol{w})$ is an upper-bound to the evacuation time for both fluid and SF networks. There are networks where the evacuation time under certain work-conserving policies can be arbitrarily close to (45). A few examples follow.

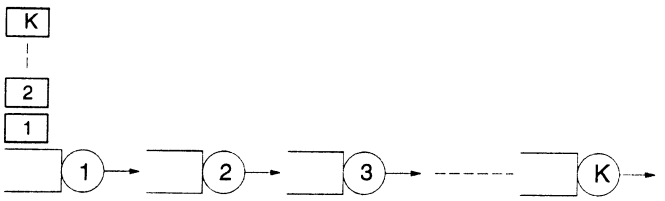

Fig. 3. Assume that message i has service time 1 in server i and ϵ in the other servers. If the messages are served in the order 1 to K and ϵ is sufficiently small, then the evacuation time is equal to $K + (K-1)\epsilon$, while if they are served in the order K to 1 it is $1 + (K-1)\epsilon$, equal to the minimum.

Consider the tandem network in Fig. 3. There are K servers and K messages initially at queue 1. Message i has service time equal to 1 in server i and equal to ϵ at any other server. If the messages are served in the order from 1 to K and ϵ is small enough, then the evacuation time is equal to $K + (K-1)\epsilon$. As ϵ goes to 0, the evacuation time approaches the upper bound for the evacuation time under any work-conserving policy. If the messages are served in the order from K to 1, then the network evacuates in time $1 + (K-1)\epsilon$ which is equal to the minimum. It is not difficult to construct examples with larger numbers of servers and more complicated topologies in which the evacuation time is equal to the upper bound as well.

The large evacuation time for the example in Fig. 3 is due to the fact that the relative service times of two messages change from server to server. Even if the relative service time of any two messages is the same for all servers serving both messages, it is possible to have evacuation times considerably larger than the minimum under certain work-conserving policies. Consider the tandem network in Fig. 4 with $K+1$ servers and service rate of server i equal to 2^{-i}. There are $K+1$ types of traffic, all at server 0 at time 0. Traffic of type i departs from the network after server i. The initial amount of traffic i is equal to $2^{-i}, i = 0, 1, \cdots, K-1$, and to $2^{-(K-1)}$ for $i = K$. If priority is given to the low index traffic over the high index traffic, the network will empty at time $(K+2)$. If priority is given to the high index traffic over the low index, then the network will empty at time 2.

If all the servers have the same rate for all traffic types, then under all work-conserving policies the tandem network (Fig. 3) will empty at the same time (fluid case). There are topologies though in which the evacuation time under certain work-conserving policies can be considerably larger than the optimal evacuation time, even if the capacities of all servers

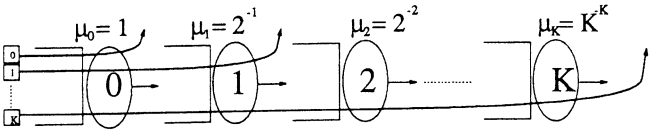

Fig. 4. There are $K+1$ types of traffic, all at server 0 at time 0. Traffic of type i departs from the network after server i. The initial amount of traffic is equal to $2^{-i}, i = 0, 1, \cdots, K-1$ and to $2^{-(K-1)}$ for $i = K$. If priority is given to the low index traffic over the high index traffic, the network will empty in $(K+2)$ time units. If priority is given to the high index traffic over the low index, then the network will empty in two time units.

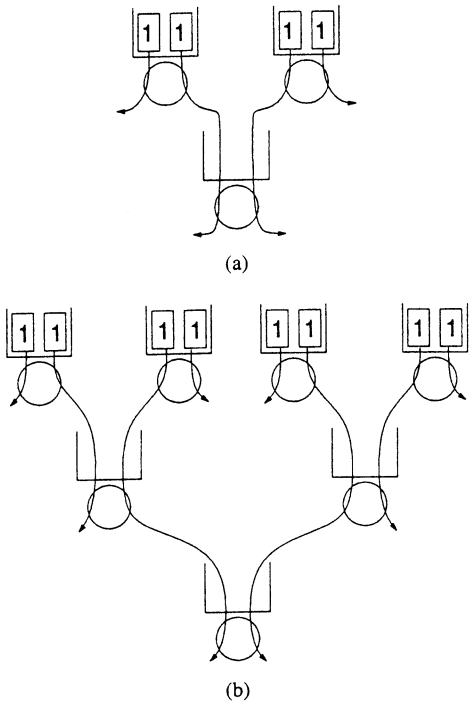

Fig. 5. One unit of traffic resides at the origin of each traffic stream in the networks (a) and (b). The service rate is equal to 1 at all servers for all traffics. The optimal evacuation time is equal to 2 in both cases. The work-conserving policy that gives priority at every server to the traffic that will exit the network in the smallest numbers of hops has evacuation time equal to 3 and 4, respectively, for the networks in (a) and (b), respectively.

are equal. Consider the network in Fig. 5(a). Traffic types 1 and 4 need service from servers 1 and 2, respectively, and then they leave the system. Traffic types 2 and 3 are served by servers 1 and 2, respectively, at the first stage and then they are both served by server 3 before they leave the system. If priority is given to traffic 2 and 3 in servers 1 and 2, then the network will empty in 2 time units, while if priority is given to traffic 1 and 4, respectively, the network will empty in 3 time units. Networks with $K+1$ service stages can be constructed where the minimum evacuation time is equal to 2 while there are work-conserving policies that have evacuation time equal to $2+K$. In Fig. 5(b) it is shown how the three-stage network can be constructed from two two-stage networks in parallel with the addition of a server in the third stage. If at each server priority is given to the traffic that will depart from the network earlier, then the evacuation time is equal to 4, compared to the minimum, that is equal to 2. A class of networks with interesting properties regarding the evacuation time is the class of ring networks, an example of which is depicted in Fig. 6. If the service rate of all streams is the same at each server and all servers have the same service rate, then in the fluid model every work-conserving policy evacuates the ring in minimum time [3], [5]. This property does not hold if the servers have different service rates, as it is indicated by the example of the tandem in Fig. 4, which is just a special case of a ring. If the service is nonpreemptive, then it is no longer true that any work-conserving policy achieves minimum evacuation time. In [11] it was shown that the policy that gives priority to the message with the furthest destination achieves minimum evacuation time in that case. Minimizing the evacuation time is a reasonable objective if the traffic in the network is uniform and the performance requirements of the different traffic types are identical. It is possible, though, that there are different types of traffic with different requirements on their service completion time. In this case, the objective of the scheduling is that each traffic type completes service within the prespecified time, which is called deadline in the following.

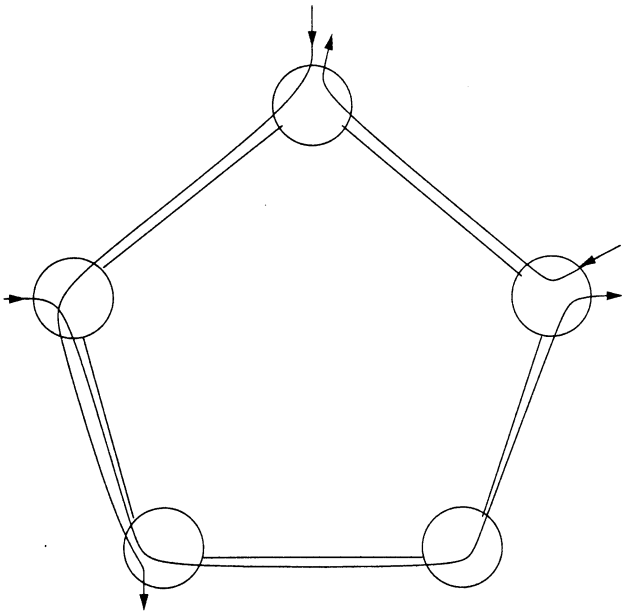

Fig. 6. An unidirectional ring network is depicted. Each server has a unique upstream and a unique downstream node.

A. Scheduling Traffic with Multiple Evacuation Time Deadlines

Assume that different traffic types in the network have different deadlines and let $D_1, \cdots, D_J$ be the collection of distinct deadlines. Without loss of generality we assume that the deadlines are in increasing order $D_j < D_{j+1}, j = 1, \cdots, J-1$ and let $D_0 = 0$. Let $\boldsymbol{w}^j$ be the workload vector of the traffic with deadline D_j. The problem of feasibility of a certain set of deadlines and the computation of the corresponding schedule is equivalent to a linear programming problem, as it is stated in the following.

Theorem 5: The deadlines $D_j, j = 1, \cdots, J$ can be met by the traffic with workload vectors $\boldsymbol{w}^j, j = 1, \cdots, J$, respectively, if and only if there are workload vectors $\boldsymbol{y}^{jk} \in \mathcal{W}, j = 1, \cdots, J, k = 0, 1, \cdots, j$ such that the following set of linear

equalities and inequalities hold:

$$\boldsymbol{y}^{j0} = \boldsymbol{w}^j, \quad \boldsymbol{y}^{jj} = 0, \qquad j = 1, \cdots, J$$
$$\boldsymbol{y}^{j(k+1)} \leq \boldsymbol{y}^{jk}, \qquad k = 0, 1, \cdots, j-1$$

and

$$\sum_{l,m:\, s_m^l = i} \frac{\sum_{j=k}^{J} y_{lm}^{jk} - \sum_{j=k+1}^{J} y_{lm}^{j(k+1)}}{\mu_{lm}} \leq D_{k+1} - D_k,$$
$$i = 1, \cdots, N. \tag{46}$$

The following piecewise-constant policy achieves the transition

$$f_{lm}(t) = \frac{\sum_{j=k}^{J} y_{lm}^{jk} - \sum_{j=k+1}^{J} y_{lm}^{j(k+1)}}{\mu_{lm}(D_{k+1} - D_k)}, \qquad D_{k+1} > t > D_k;$$
$$k = 0, \cdots, J-1; \quad l = \cdots, L; \; m = 1, \cdots, M_l. \tag{47}$$

Interpretation and Proof: Assume that there exist workload vectors $\boldsymbol{y}^{jk}$ as specified in the above theorem. Then it can be verified that the piecewise constant policy given in (47) is well defined and achieves the evacuation within the deadlines. Furthermore, the workload vector $\boldsymbol{y}^{jk}$ represents the remaining traffic with deadline D_j at time $D_k, k = 1, \cdots, J$ when the policy given in (47) is followed. Assume now that there exists an evacuation policy that achieves the deadlines. The vectors $\boldsymbol{y}^{jk}$ of remaining traffic of class j at time D_k satisfy relationships (46). ◇

The feasibility of a set of deadlines and a schedule that achieves them can be obtained by finding workload vectors $\boldsymbol{y}^{jk}$ such that the set of equalities and inequalities of Theorem 5 are satisfied. This is a linear programming problem that involves $J(J-1)/2$ unknown workload vectors. The policy in (47) that achieves the deadlines is piecewise constant with a number of changes equal to the number of distinct deadlines.

It is easy to verify that a necessary condition for achieving the deadlines is the following: by time D_j the network can be evacuated from all the traffic with deadlines less than or equal to D_j. That is

$$E\left(\sum_{k=1}^{j} \boldsymbol{w}^k\right) \leq D_j, \qquad j = 1, \cdots, J \tag{48}$$

that meets the deadlines. As a counterexample consider the network of Fig. 5(a). Assume that the deadline of traffic types 1 and 4 is equal to one, while the deadlines of traffic types 2 and 3 are equal to two. Then (48) is satisfied since the evacuation time is equal to one when only traffic types 1 and 4 are present in the network, while the evacuation time is equal to two when all the traffic are present in the network. The deadlines, though, are not achievable by all traffic types under any scheduling policy.

It turns out that for ring networks, (48) is the necessary and sufficient condition for achieving the deadlines. In fact, any work-conserving policy that gives priority to the traffic with the tighter deadline at every node, achieves the set of deadlines if and only if (48) is satisfied. To see this, note that under any such policy, for any j, the traffic of types $k > j$ is transparent to the traffic types $l \leq j$. The necessity and sufficiency follows from the fact that under any work-conserving policy the ring empties at the same time, equal to the minimum.

REFERENCES

[1] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," in *Proc. SIGCOM '89*, pp. 1–12.

[2] S. Felperin, P. Raghavan, and E. Upfal, "A theory of wormhole routing in parallel computers," in *Proc. IEEE Symp. Foundations of Computer Science*, 1992, pp. 563–572.

[3] L. Georgiadis, R. Guerin, and I. Cidon, "Throughput properties of fair policies in ring networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 718–728, Dec. 1993.

[4] L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on single link: Delay and buffer requirements," in *Proc. IEEE INFOCOM '94*, pp. 524–532.

[5] L. Georgiadis, W. Szpankowski, and L. Tassiulas, "A scheduling policy with maximal stability region for ring networks with spatial reuse," *Queueing Systems: Theory and Applicat.*, vol. 19, pp. 131–148, 1995.

[6] A. C. Greenberg and N. Madras, "How fair is fair queueing?" *J. Ass. Comput. Mach.*, vol. 3, 1992.

[7] J. M. Harrison, *Brownian motion and stochastic flow systems*. New York: Wiley, 1985.

[8] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communications switching technique," *Comput. Networks*, vol. 3, pp. 267–286, 1979.

[9] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, June 1993.

[10] ——, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr. 1994.

[11] L. Tassiulas and J. Joung, "Performance measures and scheduling policies in ring networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 576–584, 1995.

Leandros Tassiulas 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 in electrical engineering from the University of Maryland, College Park, in 1989 and 1991, respectively.

From 1991 to 1995 he was an Assistant Professor in the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY. In 1995 he joined the Department of Electrical Engineering, University of Maryland, where he is now an Associate Professor. He holds a joint appointment with the Institute for Systems Research and is a member of the Center for Satellite and Hybrid Communication Networks, established by NASA. His research interests are in computer and communication networks, with emphasis on wireless communications (terrestrial and satellite systems) and high-speed network architectures and management, in control and optimization of stochastic systems and in parallel and distributed processing.

Dr. Tassiulas coauthored a paper that received the INFOCOM '94 Best Paper Award. He received a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF Faculty Early Career Development Award in 1995, and an Office of Naval Research Young Investigator Award in 1997.