Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity

Leandros Tassiulas and Anthony Ephremides, *Fellow, IEEE*

Abstract— **Consider N parallel queues competing for the attention of a single server. At each time slot each queue may be *connected* to the server or not depending on the value of a binary random variable, the *connectivity* variable. The server is allocated to one of the connected queues at each slot; the allocation decision is based on the connectivity information and on the lengths of the connected queues only. At the end of each slot, service may be completed with a given fixed probability. Such a queueing model is appropriate for some communication networks with changing topology (radio networks with mobile users, or networks with variable links such as meteor-burst communication channels). In the case of infinite buffers, necessary and sufficient conditions are obtained for stabilizability of the system in terms of the different system parameters. The allocation policy that serves the longest connected queue stabilizes the system when the stabilizability conditions hold. The same policy minimizes the delay for the special case of symmetric queues (i.e., queues with equal arrival, service, and connectivity statistics) is provided. In a system with a single buffer per queue, an allocation policy is obtained that maximizes the throughput and minimizes the delay when the arrival and service statistics of different queues are identical.**

Index Terms— **Time varying topology, random connectivity, stability, maximum throughput, minimum delay, mobile radio networks, meteor-burst channels.**

I. Introduction

TIME-VARYING connectivity is inherent in several types of communication networks including wireless systems with mobile nodes, systems with meteor-burst communication channels, or networks in environments with hard interference (manufacturing floor). In all the cases, the connectivity varies unpredictably with time and is appropriately modeled as a random process. In this paper, we consider a queueing model of a single-hop network with randomly changing connectivity and we study the effect of varying connectivity on the performance of the system.

The queueing model consists of a single server and N parallel queues (Fig. 1). The time is slotted. At slot t each queue i may be either *connected* to the server or not; that is denoted by the binary variable $C_i(t)$, which is equal to 1 and 0 respectively. It is called the connectivity variable of queue i. The connectivity varies randomly with time. There are exogeneous arrivals at each queue. At each slot t the server is either allocated to one of the queues or idles; the *control variable* $U(t)$ indicates the queue served during slot t or is equal to e if the server idles. If the queue i at which the server is allocated is disconnected then no service is provided. If it is connected then service is provided and the served packet completes its service requirements and leaves the system with some probability; if the packet does not complete service it remains in the queue.

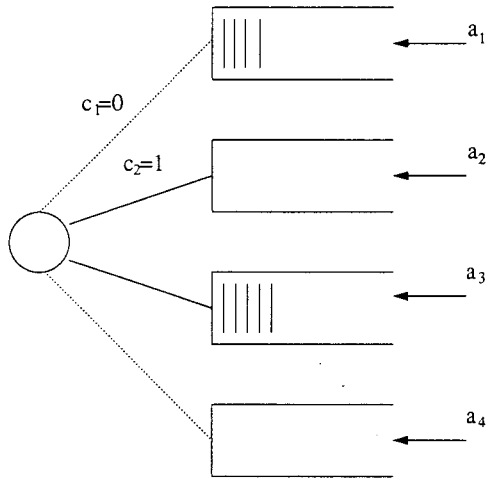

Fig. 1. Single-hop network with time varying connectivity. Solid line between a queue and the server denotes that the queue is connected to the server (it may receive service). Dashed line denotes that the queue is disconnected.

Radio networks with meteor-burst communication channels and cellular networks with mobile users and small cell sizes are two among the several examples of systems with time varying connectivity mentioned above. In the first case, there is a central station (the server) and N users (the queues) each one of which is connected to the station through a meteor burst communication channel. These channels have the property that can not be used continuously, but only during time intervals of random duration which occur at random time instants (whenever there exists a meteor burst) [4], [13]. At each time slot a user may communicate with the central station if its channel is active; hence, a subset of the users (those with active channels) are competing for the attention of the station at each slot.

As the cell size in cellular network decreases, (that is the tendency in the future cellular networks in order to maximize the spatial spectrum reutilization [5], [10]), the variability in the distance between a mobile user and the station of the cell

Manuscript received August 20, 1991; revised February 24, 1992. This work was presented in part at the IEEE International Symposium on Information Theory, Budapest, Hungary, June 24–28, 1991.

L. Tassiulas is with the Department of Electrical Engineering, Polytechnic University, 6 Metrotech Center, Brooklyn, NY 11201.

A. Ephremides is with the Department of Electrical Engineering, University of Maryland, College Park, MD 20742.

IEEE Log Number 9204101.

results in variation of their radio connectivity. At each time slot, only the users which are within a certain distance from the cell station may communicate with it. The model of a single server with parallel queues of time varying connectivity arises in this case as well.

In our queueing model the server allocation is controlled. The lengths of the connected queues are available to the controller for decision making. The allocation decision at slot t may be based on the history of the observations and the past allocation decisions. When the buffers have unlimited capacity, depending on the allocation policy and the statistics of the arrivals, services and connectivities, two things may happen. The system either reaches a steady state behavior or the queue lengths start growing without bound. In the former case the system is stable while in the latter unstable. We obtain necessary and sufficient conditions on the arrivals, service and connectivity statistics for the existence of an allocation policy under which the system is stable. We also give a policy under which the system is stable whenever there exists some policy that stabilizes it. The performance of the system with respect to queueing delay is studied then. In a symmetric system, the allocation policy that minimizes the delay is obtained. The problem of optimal server allocation in a changing connectivity system with a single buffer per node is studied last. In that case, if an arriving packet at some node i finds the buffer full then is blocked from admission into the system. A policy that maximizes the throughput and at the same time minimizes the delay is obtained.

The issue of changing connectivity in communication networks has been addressed in the past in several different contexts. In [7], a deterministic flow network with time varying link capacities is considered. The variation of the capacities of the links with time is known. The problem is to determine a dynamic flow that maximizes the amount of commodity reaching the destination within some time τ. In [8], [9], the shortest path problem in a network where the edge weight changes with time is considered. Algorithms for finding the minimum weight path at all time instances are provided. In these papers, as well as in certain of the references therein, the problem of changing connectivity is addressed in a deterministic setting. Our model captures the random nature of changing connectivity where the link connectivity is not known in advance, but is revealed gradually.

One special case of the model studied in this paper is when the connectivities are fixed and equal to one at all slots. In this case all queues are connected to the server at all times and the model is reduced to that of allocating a server to a set of parallel queues. That is a well known problem of optimal queueing control [12] and has been studied extensively in the past [2], [3]. The time varying connectivity makes the server allocation problem considerably more complicated than the case where all queues are available for service all the time. This is illustrated as we present the results that we obtained for the system with time varying connectivity in contrast with what is known for systems with fixed connectivity.

This paper is organized as follows. In Section II, we specify the model. In Section III, the stability properties of the system are investigated. The issue of queueing delay is studied in Section IV. In Section V we study throughput and delay performance in a system with a single buffer per node. A few words about the notation before we proceed. The random quantities are denoted by upper case letters; for the nonrandom quantities we reserve the lower case letters. Vectors are denoted by boldface characters. A random process, that is a sequence of random variables indexed by time, is denoted by the same symbol as the random variables without the time index.

II. The Model

During slot t there are $A_i(t)$ exogeneous arrivals at queue i. When queue i is connected and the server is allocated to that queue, the service is completed with some probability. That is represented at slot t by the binary random variable $M_i(t)$, which is equal to 1 if the service is completed and to 0, otherwise. The stability and delay optimality results are obtained under different assumptions on the statistics of the arrival, service and connectivity processes. Those in assumption 5 are stated as needed later. Let $X_i(t)$ be the number of packets in the ith queue by the end of slot t (or the beginning of slot $t+1$). Until Section V, we study the system under the assumption of unlimited buffer capacity. Under this assumption the number of packets at queue i evolves with time according to the equation

$$X_i(t) = (X_i(t-1) - 1\{U(t)=i\} \cdot C_i(t)M_i(t))^+ + A_i(t), \qquad t = 1, \cdots, \quad (2.1)$$

where $1\{\cdot\}$ is the indicator function of the event enclosed in the brackets and $(\cdot)^+$ is equal to the number enclosed in the parenthesis if this number is nonnegative and to 0, otherwise. We assume that the system starts at time 0 from some arbitrary state that is $X_i(0) = x_i$, $i = 1, \cdots, N$. We assume that the controller that allocates the server is informed at the beginning of each slot about the connectivity at that slot as well as about the lengths of the queues which are connected. This information is represented by $\boldsymbol{Y}(t) = (\boldsymbol{X}(t-1) \otimes \boldsymbol{C}(t), \boldsymbol{C}(t))$ where $\boldsymbol{X}(t) = (X_i(t): i = 1, \cdots, N)$ is the vector of queue lengths at slot t, $\boldsymbol{C}(t) = (C_i(t): i = 1, \cdots, N)$ is the vector of the connectivities at slot t and $\otimes$ denotes the pointwise product[1] between vectors. The server is allocated based on the available information $\boldsymbol{Y}(t)$. We study the stability properties and the delay performance of the system under policies that base their decisions on the available control information.

Remark 1: A single-hop radio network with a central station and several radio nodes that need to communicate with the station corresponds to the above model as follows. The server corresponds to the central station and the queues to the radio nodes. The packets have constant length equal to one slot; each time a packet is transmitted it is received successfully with some probability. Unsuccessful transmissions are due to channel errors and not to collisions since the transmissions are scheduled. The variable $M_i(t)$, in this case, indicates whether a transmission of node i at time t was successful or not (if node

[1] If $\boldsymbol{a}(a_i: i = 1, \cdots, N)$, $\boldsymbol{b} = (b_i: i = 1, \cdots, N)$ and $\boldsymbol{c} = \boldsymbol{a} \otimes \boldsymbol{b}$, then $\boldsymbol{c} = (a_i b_i: i = 1, \cdots, N)$.

i was transmitting at slot t). If a transmission is unsuccessful then the packet remains at node i.

III. System Stabilizability

We consider the system to be stable if in the long run it approaches a stationary behavior, that is if the backlog in the nodes does not grow to infinity. We study the system stability under some independence assumptions on the arrival, service and connectivity processes. All the results in this section are obtained under the following statistical assumption.

A1: The processes $\boldsymbol{A}_i$, M_i, C_i $i = 1, \cdots, N$ are i.i.d. and independent; furthermore the arrivals satisfy $E[A_i^2(t)] < \infty$.

Consider the class of stationary policies G that allocate the server at slot t based on the available information $\boldsymbol{Y}(t)$. A policy in G is specified by a function g: $\mathcal{Y}^1 \rightarrow \{1, \cdots, N, e\}$ where $\mathcal{Y}^1$ is the space at which $\boldsymbol{Y}(t)$ lies. The allocation decision at slot t is $U(t) = g(\boldsymbol{Y}(t))$. Under any policy in G and because of the independence assumptions on the arrivals, services and connectivities, the queue length process $\boldsymbol{X} = \{\boldsymbol{X}(t)\}_{t=1}^{\infty}$ is a time homogeneous Markov chain with state space $\mathcal{X} = Z_+^N$.

Definition 1: The system is defined to be stable under some allocation policy in G if the Markov chain $\boldsymbol{X}$ is irreducible and the probability distribution of $\boldsymbol{X}(t)$ converges in the sense that

$$\lim_{t\rightarrow\infty} P[\boldsymbol{X}(t) \leq \boldsymbol{b}] = F(\boldsymbol{b}), \qquad \forall \boldsymbol{b} \in \mathcal{X}, \tag{3.1}$$

where $F(\cdot)$ is a probability distribution on $\mathcal{X}$.

Definition 2: The system is called *stabilizable* if there exists an allocation policy in $\boldsymbol{G}$ under which it is stable.

The necessary and sufficient stabilizability conditions involve the expectations of $A_i(t)$, $C_i(t)$, $M_i(t)$ that are denoted by $a_i = E[A_i(t)]$, $p_i = E[C_i(t)]$ and $m_i = E[M_i(t)]$. Consider the class G_0 of stationary policies that base the allocation decision at slot t on the lengths of all queues $\boldsymbol{X}(t-1)$ and not only of the connected ones. Under any such policy $\boldsymbol{X}(t)$ is a Markov chain. The next lemma provides a condition that is necessary for stabilizability of the system even if the queue lengths of the disconnected queues are observable at each slot.

Lemma 1: If there exists a policy π in G_0 under which the system is stable, then

$$\sum_{i\in Q} \frac{a_i}{m_i} < 1 - \prod_{i\in Q}(1-p_i), \qquad \forall Q \subset \{1, \cdots, N\}. \tag{3.2}$$

Proof: Assume that the system is operating under some policy in G_0 and is stable. Definition 1 implies that the Markov chain $\boldsymbol{X}$ is ergodic and possesses a stationary distribution. We start the system with its stationary distribution therefore the queue length process is stationary and ergodic. Let $h_j(t)$ be the indicator variable that is equal to 1 if queue j is connected and receives service at slot t and to 0, otherwise. The departure process from queue j is $\{h_j(t)M_j(t)\}_{t=1}^{\infty}$ and is stationary and ergodic. The departure rate from queue j is

$$E[h_j(t)M_j(t)] = m_j E[h_j(t)].$$

Since the system is stationary and ergodic, in each queue the departure rate should be equal to the arrival rate; that is,

$$m_j E[h_j(t)] = a_j. \tag{3.3}$$

Hence, from (3.3) we have, for any set of queues Q,

$$\sum_{j\in Q} \frac{a_j}{m_j} = \sum_{j\in Q} E[h_j(t)]. \tag{3.3a}$$

The sum in the right-hand side of (3.3a) can be written as

$$\sum_{j\in Q} E[h_j(t)] = E\left[E\left[\sum_{j\in Q} h_j(t) | C_l(t),\ X_l(t-1),\ l\in Q\right]\right]. \tag{3.4}$$

Consider the partition of the probability space into the events

$$B_1 = \{C_j(t) = 0, j\in Q\},$$

$$B_2 = \{C_j(t) = 0, j\in Q\}^c \cap \{X_j(t-1) = 0, j\in Q\},$$

$$B_3 = \{C_j(t) = 0, j\in Q\}^c \cap \{X_j(t-1) = 0, j\in Q\}^c,$$

where A^c is the complementary set of A. Notice that

$$E\left[\sum_{j\in Q} h_j(t) | C_j(t),\ X_j(t-1),\ j\in Q;\ B_l\right] = 0, \qquad l = 1, 2,$$

$$E\left[\sum_{j\in Q} h_j(t) | C_j(t),\ X_j(t-1),\ j\in Q;\ B_3\right] \leq 1;$$

hence, we have

$$\begin{aligned} &E\left[E\left[\sum_{j\in Q} h_j(t) | C_l(t),\ X_l(t-1),\ l\in Q\right]\right] \\ &= E\left[\sum_{l=1}^{3} E\left[\sum_{j\in Q} h_j(t) | C_l(t)\,,\right.\right. \\ &\qquad \left.\left. \cdot X_l(t-1),\ l\in Q;\ B_l\right] P[B_l]\right] \\ &\leq 1 - P[B_1] - P[B_2]. \end{aligned} \tag{3.5}$$

Since the Markov chain $\boldsymbol{X}$ is irreducible and ergodic, under the stationary distribution we have $P[X_j(t) = 0,\ j\in Q] > 0$ for any $Q \subset \{1, \cdots, N\}$; hence, we have

$$\begin{aligned} P[B_2] = (1 - P[C_j(t) = 0, j\in Q]) \\ \cdot P[X_j(t-1) = 0, j\in Q] > 0. \end{aligned} \tag{3.5a}$$

Because of the independence of the connectivity processes that correspond to different queues, we have

$$P[B_1] = \prod_{i\in Q}(1-p_i). \tag{3.5b}$$

Relations (3.5), (3.5a), (3.5b) imply

$$E\left[E\left[\sum_{j\in Q} h_j(t)|C_l(t),\, X_l(t),\, l\in Q\right]\right] < 1-\prod_{i\in Q}(1-p_i). \quad (3.6)$$

Equations (3.3a), (3.4), (3.6) imply (3.2). □

Note that $\sum_{i\in Q}(a_i/m_i)$ is the rate with which work (in the form of service slots) is entering the set Q of queues and $1-\prod_{i\in Q}(1-p_i)$ is the proportion of slots at which at least one queue of Q is connected and can receive service; hence the necessity of (3.2) for stability can be visualized. The sufficiency though of (3.2) for stability can not be seen easily in advance since the rate at which service is provided to the queues within set Q is strictly less than $1-\prod_{i\in Q}(1-p_i)$. That is because the connected queues of the set Q at each slot t may be either empty or have length less than that of another connected queue out of Q. In the next lemma it is shown that conditions (3.2) are sufficient for stabilizability as well. Consider the *longest connected queue* (LCQ) policy that during slot t allocates the server according to the function $g_0\colon \mathcal{Y}^1 \to \{e, 1, \cdots, N\}$ defined by

$$g_0(\boldsymbol{x}, \boldsymbol{c}) = \begin{cases} e, & \text{if } x_i c_i = 0, \\ & \quad i = 1, \cdots, N, \\ \arg\max_{i=1,\cdots,N}\{x_i c_i\}, & \text{otherwise.} \end{cases}$$

As its name implies, the LCQ policy allocates the server at slot t to the connected queue $i(C_i(t)=1)$ with maximum length. The policy LCQ is shown next to stabilize the system as long as there exists a policy in G_0 under which it is stable. In the following, we let $h_j(t) = 1\{g_0(\boldsymbol{X}(t-1), \boldsymbol{C}(t)) = j\}$.

Lemma 2: The system is stable under LCQ if

$$\sum_{i\in Q}\frac{a_i}{m_i} < 1-\prod_{i\in Q}(1-p_i), \qquad \forall Q \subset \{1,\cdots,N\}.$$

Proof: Under LCQ, $\boldsymbol{X}$ is clearly irreducible. We use Foster's criterion for ergodicity of a Markov chain ([1]) to show that $\boldsymbol{X}$ is ergodic under the condition of the lemma; from ergodicity (3.1) is implied. Consider the function V defined on the state space $\mathcal{X}$ of the chain by $V(\boldsymbol{x}) = \sum_{i=1}^N m_i^{-1}x_i^2$. For all $\boldsymbol{x}\in\mathcal{X}$, we have

$$\begin{aligned} &E[V(\boldsymbol{X}(t+1))|\boldsymbol{X}(t)=\boldsymbol{x}] \\ &\quad = E\left[\sum_{i=1}^N m_i^{-1}X_i^2(t+1)|\boldsymbol{X}(t)=\boldsymbol{x}\right] \\ &\quad \le E\left[\sum_{i=1}^N m_i^{-1}(x_i+A_i(t+1))^2|\boldsymbol{X}(t)=\boldsymbol{x}\right] \\ &\quad = V(\boldsymbol{X}(t)) + 2\sum_{i=1}^N m_i^{-1}a_i x_i \\ &\qquad + \sum_{i=1}^N m_i^{-1}E[A_i^2(t+1)] < \infty. \end{aligned} \quad (3.6a)$$

We show that if condition (3.2) is satisfied then for a fixed $\epsilon > 0$ there exists a number b, which may be a function of the first and second moments of the arrival, service, and connectivity processes, for which we have

$$E[V(\boldsymbol{X}(t+1)) - V(\boldsymbol{X}(t))|\boldsymbol{X}(t)] < -\epsilon, \quad \text{if } V(\boldsymbol{X}(t)) > b. \quad (3.7)$$

Notice that the set

$$V_b = \{\boldsymbol{x}\colon V(\boldsymbol{x}) \le b,\, \boldsymbol{x}\in Z_+^N\}$$

has finite cardinality for all b. From (3.6a), (3.7), we can conclude that $\boldsymbol{X}(t)$ is ergodic. We proceed now to show (3.7). By simple calculations we get

$$\begin{aligned} &E[V(\boldsymbol{X}(t+1)) - V(\boldsymbol{X}(t))|\boldsymbol{X}(t)] \\ &\quad = E\left[\sum_{i=1}^N m_i^{-1}(X_i(t+1)-X_i(t)) \right. \\ &\qquad \left. \cdot\,(X_i(t+1)-X_i(t)+2X_i(t))|\boldsymbol{X}(t)\right] \\ &\quad = E\left[\sum_{i=1}^N 2m_i^{-1}X_i(t)(X_i(t+1)-X_i(t))|\boldsymbol{X}(t)\right] \\ &\qquad + E\left[\sum_{i=1}^N m_i^{-1}(X_i(t+1)-X_i(t))^2|\boldsymbol{X}(t)\right]. \end{aligned} \quad (3.8)$$

The second term of the sum in the right-hand side of (3.8) can be upper bounded as

$$\begin{aligned} &E\left[\sum_{i=1}^N m_i^{-1}(X_i(t+1)-X_i(t))^2|\boldsymbol{X}(t)\right] \\ &\qquad \le E\left[\sum_{i=1}^N m_i^{-1}(A_i(t+1))^2|\boldsymbol{X}(t)\right] + 1 \\ &\qquad = \sum_{i=1}^N m_i^{-1}E[A_i^2(t)] + 1. \end{aligned} \quad (3.9)$$

For the first term of the sum in the right-hand side of (3.8), we have

$$\begin{aligned} &E\left[\sum_{i=1}^N 2m_i^{-1}X_i(t)(X_i(t+1)-X_i(t))|\boldsymbol{X}(t)\right] \\ &\quad = E\left[\sum_{i=1}^N 2m_i^{-1}X_i(t)A_i(t+1)|\boldsymbol{X}(t)\right] \\ &\qquad - E\left[\sum_{i=1}^N 2m_i^{-1}X_i(t)M_i(t+1)h_i(t+1)|\boldsymbol{X}(t)\right]. \end{aligned} \quad (3.10)$$

The first term of the sum in the right-hand side of (3.10) is

$$E\left[\sum_{i=1}^N 2m_i^{-1}X_i(t)A_i(t+1)|\boldsymbol{X}(t)\right] = 2\sum_{i=1}^N X_i(t)\frac{a_i}{m_i}. \quad (3.11)$$

We need to introduce some notation before we manipulate the second term of the sum in (3.10). Consider a permutation e_i, $i = 0,\cdots,N$ of the integers 0 to N which is such that $e_0 = 0$, $X_{e_i}(t) \ge X_{e_{i-1}}(t)$, for $i = 2,\cdots,N$, and if

$X_{e_i}(t) = X_{e_{i-1}}(t)$ then $e_i > e_{i-1}$. Consider also a partition of the probability space into the events D_i, $i = 0, \cdots, N$, defined by

$$D_0 = \{\boldsymbol{C}(t+1) = \mathbf{0}\},$$
$$D_i = \{C_{e_i}(t+1) = 1, C_{e_j}(t+1) = 0 \text{ for} N \geq j > i\} \quad \text{for } i = 1, \cdots, N.$$

The probabilities of the events D_i are

$$P[D_0] = \prod_{i=1}^{N}(1 - p_i),$$
$$P[D_i] = p_{e_i} \prod_{j=i+1}^{N} (1 - p_{e_j}), \qquad i = 1, \cdots, N. \quad (3.12)$$

Clearly the permutation as well as the events D_i depend on the state $\boldsymbol{X}(t)$ and the connectivity vector $\boldsymbol{C}(t)$ at each slot t. Now we can calculate the second term of the sum in the right-hand side of (3.10) to be

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t) M_i(t+1) h_i(t+1) | \boldsymbol{X}(t)\right] \\ &= E\left[\sum_{i=1}^{N} 2X_i(t) h_i(t+1) | \boldsymbol{X}(t)\right] \\ &= E\left[\sum_{i=1}^{N} 2X_{e_i}(t) h_{e_i}(t+1) | \boldsymbol{X}(t)\right] \\ &= \sum_{j=0}^{N} E\left[\sum_{i=1}^{N} 2X_{e_i}(t) h_{e_i}(t+1) | \boldsymbol{X}(t), D_j\right] P(D_j). \end{aligned} \quad (3.13)$$

Notice that from the definition of the policy, in the event D_j, queue e_j is served if it is not empty. If it is empty then every other connected queue is empty as well. Therefore, we have

$$E\left[\sum_{i=1}^{N} 2X_{e_i}(t) h_{e_i}(t+1) | \boldsymbol{X}(t), D_j\right] = 2X_{e_j}(t). \quad (3.13a)$$

From (3.12), (3.13), (3.13a), we get

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t) M_i(t+1) h_i(t+1) | \boldsymbol{X}(t)\right] \\ &\qquad = \sum_{i=1}^{N} 2X_{e_i}(t) p_{e_i} \prod_{j=i+1}^{N} (1 - p_{e_j}), \end{aligned} \quad (3.14)$$

where $\prod_{j=N+1}^{N}(\cdot) = 1$. By a simple calculation in the right side of (3.14), we get

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t) M_i(t+1) h_i(t+1) | \boldsymbol{X}(t)\right] \\ &= 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) \\ &\quad \cdot \left(1 - \prod_{i=j}^{N}(1 - p_{e_i})\right) + 2X_{e_1}(t) \\ &\quad \cdot \left(1 - \prod_{i=1}^{N}(1 - p_{e_i})\right). \end{aligned} \quad (3.15)$$

Using the permutation we defined earlier and after some calculations (3.11) can be written as

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t) A_i(t+1) | \boldsymbol{X}(t)\right] \\ &= 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) \sum_{i=j}^{N} \frac{a_{e_i}}{m_{e_i}} \\ &\quad + 2X_{e_1}(t) \sum_{i=1}^{N} \frac{a_{e_i}}{m_{e_i}}. \end{aligned} \quad (3.16)$$

From (3.10), (3.15), and (3.16), we get

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t)(X_i(t+1) - X_i(t)) | \boldsymbol{X}(t)\right] \\ &= 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) \sum_{i=j}^{N} \frac{a_{e_i}}{m_{e_i}} \\ &\quad + 2X_{e_1}(t) \sum_{i=1}^{N} \frac{a_{e_i}}{m_{e_i}} \\ &\quad - 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) \left(1 - \prod_{i=j}^{N}(1 - p_{e_i})\right) \\ &\quad - 2X_{e_1}(t) \left(1 - \prod_{i=1}^{N}(1 - p_{e_i})\right) \\ &= 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) \\ &\quad \cdot \left(\sum_{i=j}^{N} \frac{a_{e_i}}{m_{e_i}} - 1 + \prod_{i=1}^{N}(1 - p_{e_i})\right) \\ &\quad + 2X_{e_1}(t) \left(\sum_{i \in 1}^{N} \frac{a_{e_i}}{m_{e_i}} - 1 + \prod_{i \in 1}^{N}(1 - p_{e_i})\right). \end{aligned} \quad (3.17)$$

We define

$$c = \max_{Q \subset \{1, \cdots, N\}} \left\{\sum_{i \in Q} \frac{a_i}{m_i} - 1 + \prod_{i \in Q}(1 - p_i)\right\}. \quad (3.17a)$$

From (3.17), (3.17a) we get

$$\begin{aligned} &E\left[\sum_{i=1}^{N} 2m_i^{-1} X_i(t)(X_i(t+1) - X_i(t)) | \boldsymbol{X}(t)\right] \\ &\leq 2\sum_{j=2}^{N} (X_{e_j}(t) - X_{e_{j-1}}(t)) c + 2X_{e_1}(t) c \\ &= 2X_{e_N}(t) c. \end{aligned} \quad (3.18)$$

From (3.8), (3.9), and (3.18), we get

$$\begin{aligned} &E[V(\boldsymbol{X}(t+1)) - V(\boldsymbol{X}(t)) | \boldsymbol{X}(t)] \\ &\qquad \leq \sum_{i=1}^{N} E[(A_i(t))^2] + 1 + 2X_{e_N}(t) c. \end{aligned} \quad (3.19)$$

If $V(\boldsymbol{X}(t)) \geq b$, then

$$X_{e_N}(t) \geq \sqrt{\frac{b}{\sum_{i=1}^{N} 1/m_i^2}}$$

and from condition (3.2), we have $c < 0$. Hence, from (3.19) we get

$$E[V(\boldsymbol{X}(t+1)) - V(\boldsymbol{X}(t))|\boldsymbol{X}(t)] \leq \sum_{i=1}^{N} E[(A_i(t))^2] + 1 + 2c\sqrt{\frac{b}{\sum_{i=1}^{N} 1/m_i^2}}. \quad (3.20)$$

If

$$b = \sum_{i=1}^{N} 1/m_i^2 \left(\frac{\epsilon + 1 + \sum_{i=1}^{N} E[(A_i(t))^2]}{2c} \right)^2,$$

then the right-hand side of (3.20) equals to $-\epsilon$ and the proof is complete. □

The next theorem summarizes these results.

Theorem 1: The necessary and sufficient stabilizability condition is

$$\sum_{i \in Q} \frac{a_i}{m_i} < 1 - \prod_{i \in Q}(1 - p_i), \qquad \forall Q \subset \{1, \cdots, N\}. \quad (3.20a)$$

Furthermore, policy LCQ stabilizes the system as long as it is stabilizable.

Corollary 1: When the arrival and service rates as well as the connectivity probabilities of all queues are the same and equal to a, m, and p respectively, then the necessary and sufficient stabilizability condition (3.2) is equivalent to

$$\frac{a}{m} < \frac{1 - (1-p)^N}{N}. \quad (3.21)$$

Proof: Since all nodes are identical, for any set Q with k nodes, condition (3.2) is written as

$$\frac{a}{m} < \frac{1 - (1-p)^k}{k}. \quad (3.22)$$

When Q includes all nodes of the network then (3.2) is identical to (3.21). To show that (3.21) implies (3.22) for all k, it is enough to show

$$\frac{1 - (1-p)^k}{k} \geq \frac{1 - (1-p)^{k+1}}{k+1}, \qquad k = 1, 2, \cdots,$$

which is true since

$$\frac{1 - (1-p)^k}{k} \geq \frac{1 - (1-p)^{k+1}}{k+1}$$

$$\Leftrightarrow (k+1)p\sum_{i=0}^{k-1}(1-p)^i \geq kp\sum_{i=0}^{k}(1-p)^i$$

$$\Leftrightarrow \sum_{i=0}^{k-1}(1-p)^i \geq k(1-p)^k. \qquad \square$$

For a symmetric system like that considered in the corollary, the maximal total throughput is equal to $1-(1-p)^N$ and the performance degradation due to time varying connectivity is equal to $(1-p)^N$, which is the probability that all nodes are disconnected during a particular slot.

Heuristic Interpretation of the LCQ Policy: In the changing connectivity system, service is wasted when at some slot all the connected queues are empty and the server is forced to idle. This phenomenon is unlikely to happen when the backlog in the system is distributed as evenly as possible to the queues so that the smallest number of queues are empty. The LCQ policy achieves this even distribution of the backlog by serving the longest connected queues which are more unlikely to become empty than the short ones.

When the system has fixed connectivities ($C_i(t) = 1$ a.s., $i = 1, \cdots, N$, $t = 1, \cdots$), it is well known [12] that the necessary and sufficient stabilizability condition is

$$\sum_{i=1}^{N} \frac{a_i}{m_i} < 1.$$

Furthermore, under the necessary and sufficient stabilizability condition the system is stabilized by any work conserving policy that is by any policy which never idles the server if there are packets in the system. When the connectivities are time varying a policy is defined to be *work conserving* if it does not idle the server when there is a nonempty connected queue. Any work conserving policy in the latter case does not necessarily stabilize the system even if it is stabilizable. This is demonstrated in the following counterexample.

Counterexample 1: Consider a system with two queues which have Bernoulli arrivals with rates a_1 and a_2 respectively. The server provides deterministic service to both queues, ($m_1 = m_2 = 1$); queue 1 is constantly available for service ($p_1 = 1$) while queue 2 is available with probability $p_2 < 1$. The stability condition (3.2) in this case is equivalent to the following:

$$a_1 + a_2 < 1, \qquad a_2 < p_2. \quad (3.23)$$

Consider the nonidling policy π' that always gives priority to queue 1. We claim that (3.23) is not sufficient for stability of the system under π'. Assume that the system starts with queue 1 being empty. At slot t, queue 2 may receive service if it is connected and no packet arrived at queue 1 during slot $t-1$. Hence, if the system is stable, the stationary probability of the event that queue 2 is served at slot t is less than or equal to $p_2(1-a_1)$. Therefore, a necessary stability condition for queue 2 is that

$$a_2 < p_2(1 - a_1). \quad (3.24)$$

Clearly, we can find nonnegative numbers a_1, a_2, p_2 that satisfy (3.23) but do not satisfy (3.24); hence, (3.23) is not sufficient for stability under π'.

Remark 2: In order to verify the stabilizability condition (3.2) we have to verify the inequality in (3.2) for all subsets Q of the set $\{1, \cdots, N\}$. The number of subsets of that set is 2^N. Hence, for a large number of queues, verifying whether the system is stabilizability for certain arrival, service and connectivity rates becomes an intractable task. It is an interesting open problem to find an efficient algorithm, if there exists one, that verifies stabilizability in polynomial time.

Remark 3: Condition (3.2) is necessary for the existence of a policy in G under which the system is stable and sufficient for stability of the system under LCQ. The policies in G_0 may base their decision on the lengths of all queues in the system irrespectively of whether they are connected or not while LCQ base its decisions on the lengths of the connected queues only. Hence, the additional information on which the policies in G_0 may base their decisions, that is the lengths of the unobservable queues, is irrelevant for the stability of the system.

Remark 4: The independence of the processes $\boldsymbol{A}_i, \boldsymbol{M}_i, \boldsymbol{C}_i$, $i = 1, \cdots, N$, has not been used in the proof of Theorem 1. The stability result in that theorem holds under the more general assumption that the variables $A_i(t)$, $M_i(t)$, $C_i(t)$, $i = 1, \cdots, N$, are independent in different slots and identically distributed but not necessarily independent among themselves in the same slot. Under that more general assumption the theorem holds if the term $1 - \prod_{i \in Q}(1 - p_i)$ in the right side of relationship (3.20a) is replaced by $P[\sum_{i \in Q} C_i(t) > 0]$.

IV. Optimal Server Allocation

In this section, we study the problem of optimal server allocation with respect to delay. We consider a symmetric system that is one in which the following assumption holds.

A2: The arrival service and connectivity processes in different queues have identical statistics. Furthermore the variables $A_i(t)$, $i =, \cdots, N$, $t = 1, \cdots$, are binary.

The policy LCQ defined in section 3, which allocates the server at each slot to the longest connected queue is shown to be optimal in a symmetric system; more specifically it minimizes, in the stochastic ordering sense, the process of total number of packets in the system. In the following, we give the definition of stochastic ordering and a theorem that will be used later (for more details on the notion of stochastic ordering the reader is referred to [11]). Consider the discrete-time processes $mbiX = \{X(t)\}_{t=1}^{\infty}$, $\boldsymbol{Y} = \{Y(t)\}_{t=1}^{\infty}$ and the space of all real valued sequences $\mathcal{R} = R^{Z_+}$. We say that the process $\boldsymbol{X}$ is stochastically smaller than the process $\boldsymbol{Y}$, and write $\boldsymbol{X} \leq_{st} \boldsymbol{Y}$ if $P\{f(X) > z\} \leq P\{f(Y) > z\}$ for every $z \in R$, where f: $\mathcal{R} \rightarrow R$ is measurable and $f(x) \leq f(y)$ for every $x, y \in \mathcal{R}$ such that $x(t) \leq y(t)$ for $t \in Z_+$. The next theorem provides alternative characterizations of the stochastic ordering relationship between two processes.

Theorem 2 ([11]): The following three statements are equivalent.

a) $\boldsymbol{X} \leq_{st} \boldsymbol{Y}$.

b) $P(g(X(t_1), \cdots, X(t_n)) > z) \leq P(g(Y(t_1), \cdots, Y(t_n)) > z)$ for all $(t_1, \cdots, t_n)$, all z, and n, and for all g: $R^n \rightarrow R$, measurable and such that $x_j \leq y_j$, $1 \leq j \leq n$ implies $g(x_1, \cdots, x_n) \leq g(y_1, \cdots, y_n)$.

c) There exist two stochastic processes $X' = \{X'(t)\}_{t=1}^{\infty}$, $Y' = \{Y'(t)\}_{t=1}^{\infty}$ on a common probability space with the same probability laws as X and Y, respectively, such that $X'(t) \leq Y'(t)$ a.s. for every $t \in Z_+$.

Note that if the process of total number of packets in the system under policy LCQ is stochastically smaller than the corresponding process under some other policy π then the average number of packets in the system under LCQ is smaller than that under π (if the average is well defined). Therefore, by Little's law, we know that the average delay under LCQ is smaller than that under π. Hence, optimality in the stochastic ordering sense is stronger than average delay optimality and implies the latter.

A. Optimality of LCQ

Consider the class of policies $\tilde{G}$ that take an action at slot t based on the entire history of the past observations and control actions. A policy in $\tilde{G}$ is specified by a sequence of functions $\{g_t(\cdot)\}_{t=1}^{\infty}$, g_t: $\mathcal{Y}^t \times \{e, 1, \cdots, N\}^{t-1} \rightarrow \{e, 1, \cdots, N\}$ where $\mathcal{Y}^t$ is the space where $\boldsymbol{Y}^t(t) = (\boldsymbol{Y}(1), \cdots, \boldsymbol{Y}(t))$ lies. The allocation decision at slot t is $U(t) = g_t(\boldsymbol{Y}^t(t), \boldsymbol{U}^t(t))$ where $\boldsymbol{U}^t(t) = (U(1), \cdots, U(t-1))$. Clearly $\tilde{G}$ is a bigger class of policies than G. We show that LCQ is optimal within $\tilde{G}$. We need notation to consider the process of total number of packets in the system $Q = \{Q(t)\}_{t=1}^{\infty}$ where $Q(t) = \sum_{i=1}^{N} X_i(t)$. The next theorem states that LCQ minimizes in the stochastic ordering sense the process of total number of packets in the system.

Theorem 3: Let Q be the process of total number of packets in the system when the initial state is $\boldsymbol{x}_0$ and some policy $\pi \in \tilde{G}$ acts on it and Q_0 the corresponding process when LCQ acts on the system. Then,

$$Q_0 \leq_{st} Q. \tag{4.1}$$

We need the following lemma in the proof of the theorem.

Lemma 3: For e very policy $\pi \in \tilde{G}$, there exists a policy $\tilde{\pi} \in \tilde{G}$ that acts similarly to LCQ at $t = 1$ and is such that when the system is in state $\boldsymbol{x}_0$ at $t = 0$ and policies π, $\tilde{\pi}$ act on it, the corresponding processes Q and $\tilde{Q}$ of total number of packets in the system can be constructed by appropriate coupling of the arrival, service and connectivity processes so that

$$\tilde{Q}(t) \leq_{st} Q(t) \quad \text{a.s.}, \qquad t = 0, 1, 2, \cdots. \tag{4.2}$$

Proof: We construct $\tilde{\pi}$ and we couple the queue length realizations under π and $\tilde{\pi}$ appropriately so that (4.2) holds. Let $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ be the queue length processes under policies π and $\tilde{\pi}$, respectively. More specifically, we show that at every slot t the queue lengths satisfy either relationship (4.4) or relationship (4.5) defined later, both of these relationships imply (4.2). We show that (4.4) or (4.5) are satisfied at every slot using the technique of forward induction in time [12]. We show first that at $t = 1$ the relations (4.4), (4.5) are satisfied; then we show that if these relationships are satisfied at some time t then they are satisfied at $t + 1$ as well.

At slot $t = 1$, let the same queues have the same connectivity variables under the two policies.

If π and $\tilde{\pi}$, take the same action at $t = 1$ then let the arrival, service, and connectivity variables be the same at the same queues under both policies for every subsequent slot and take $\tilde{\pi}$ to coincide with π for $t = 2, \cdots$. Then the queue length processes are identical under both policies and (4.2) follows immediately.

If π idles at $t = 1$ while $\tilde{\pi}$ serves queue j, then let the same queues have, the same arrival, service and connectivity variables under both policies at all subsequent slots. At $t = 1$, we have

$$X_l(t) = \tilde{X}_l(t), \qquad \text{if } l \neq j, \qquad \tilde{X}_j(t) \leq X_j(t). \tag{4.3}$$

Let policy $\tilde{\pi}$ be identical to π at all subsequent slots $t = 2, 3, \cdots$. If (4.3) holds at t, we can easily see that it holds at $t + 1$ as well and (4.2) follows by induction.

If π serves queue k while $\tilde{\pi}$ serves queue j at $t = 1$, then at that time slot let queues k and j have the same service variable under π and $\tilde{\pi}$, respectively. At each time $t \geq 1$, consider the indicator variables $s(t)$, $\tilde{s}(t)$, $l(t)$, $\tilde{l}(t)$ defined as follows:

$$s(t) = \arg\min_{m=j,\,k} \{X_m(t)\}, \qquad \tilde{s}(t) = \arg\min_{m=j,\,k} \{\tilde{X}_m(t)\},$$

$$l(t) = \arg\max_{m=j,\,k} \{X_m(t)\}, \qquad \tilde{l}(t) = \arg\max_{m=j,\,k} \{\tilde{X}_m(t)\}.$$

If we have $X_j(t) = X_k(t)$ then we take $s(t) = \min\{j, k\}$. Similarly for the rest indicator variables. In the following we write $X_s(t)$ instead of $X_{s(t)}(t)$. The same for the rest of the above indicator variables. We distinguish the following cases.

Case 1: $X_k(0) = X_j(0)$. Assign the same arrival variables at $t = 1$ to the queues j and k under $\tilde{\pi}$ and π, respectively. Assign the same arrival variables at $t = 1$ to the queues k and j under $\tilde{\pi}$ and π, respectively. Assign the same arrival variables under both policies to each one of the rest of the queues. Then, at $t = 1$, the queue lengths satisfy the following relationships

$$\begin{aligned} X_s(t) &= \tilde{X}_{\tilde{s}}(t), \qquad X_l(t) = \tilde{X}_{\tilde{l}}(t), \\ X_i(t) &= \tilde{X}_i(t), \qquad i \neq k, j. \end{aligned} \tag{4.4}$$

Case 2: $X_k(0) < X_j(0)$. At slot $t = 1$, assign the same arrival variables to the same queues under π and $\tilde{\pi}$. If the service at $t = 1$ is not completed then the queue lengths at $t = 1$ satisfy (4.4). If the service is completed, we distinguish the following cases.

a) $X_k(0) < X_j(0) - 1$. In this case, we can easily verify that the queue lengths satisfy the following relationships

$$\begin{aligned} X_s(t) &= \tilde{X}_{\tilde{s}}(t) - 1, \qquad X_l(t) = \tilde{X}_{\tilde{l}}(t) + 1, \\ X_i(t) &= \tilde{X}_i(t), \qquad i \neq k, j. \end{aligned} \tag{4.5}$$

for $t = 1$.

b) $X_k(0) = X_j(0) - 1$. In this case, the queue lengths are as follows depending on the arrivals. If during slot 1 a packet arrives only at queue k (under both policies) then the queue lengths at the end of slot 1 satisfy (4.4); otherwise the queue lengths satisfy relations (4.5).

Cases 1) and 2) cover all the possibilities since it is not possible to have $X_k(0) > X_j(0)$ given that LCQ serves the longest queue. Hence, at the end of slot 1, the queue lengths under π and $\tilde{\pi}$ satisfy either (4.4) or (4.5). Note that in both cases we have

$$\sum_{i=1}^{N} X_i(t) = \sum_{i=1}^{N} \tilde{X}_i(t). \tag{4.5a}$$

Hence, if at each slot either (4.4) or (4.5) hold then (4.5a) holds at all slots and (4.2) is satisfied. We show in the following that if the queue lengths at slot t satisfy either (4.4) or (4.5) then we can couple the processes $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ by choosing appropriately the connectivity, arrival and service variables at slot $t+1$, and define $\tilde{\pi}$ such that at slot $t + 1$ one of the relations (4.4) and (4.5) is satisfied again. From induction, we can conclude that there exists a $\tilde{\pi}$ such that the queue length processes under π and $\tilde{\pi}$ satisfy either (4.4) or (4.5) at any t; hence (4.2) holds and the lemma follows. We distinguish the following cases for $\boldsymbol{X}(t)$.

Case 1′: Relations (4.4) hold at t.

Let at slot $t + 1$ the queues $l(t)$, $\tilde{l}(t)$ have the same connectivity arrival and service variables under π and $\tilde{\pi}$, respectively. Similarly for the queues $s(t)$ and $\tilde{s}(t)$. Let all queues, other than k, j, have the same connectivity, arrival and service variables at $t + 1$ under π and $\tilde{\pi}$, respectively. If π serves queue $l(t)$ at slot $t + 1$, let $\tilde{\pi}$ serve $\tilde{l}(t)$; if π serves queue $s(t)$, let $\tilde{\pi}$ serve $\tilde{s}(t)$. Let $\tilde{\pi}$ be identical to π, otherwise. Then we can easily check that at $t + 1$, (4.4) are satisfied.

Case 2′: Relations (4.5) hold at t and $\tilde{X}_{\tilde{s}}(t) < \tilde{X}_{\tilde{l}}(t)$.

Let the connectivity, arrival and service variables at slot $t + 1$ as well as the policy $\tilde{\pi}$ be as in Case 1′). If we have $\tilde{X}_{\tilde{s}}(t) \leq \tilde{X}_{\tilde{l}}(t) - 2$ then (4.5) hold at slot $t + 1$. If we have $\tilde{X}_{\tilde{s}}(t) = \tilde{X}_{\tilde{l}}(t) - 1$, the following may hold. If queues $s(t)$ and $\tilde{s}(t)$ are served under π and $\tilde{\pi}$, respectively, then at slot $t + 1$ (4.5) hold. If instead queues $l(t)$ and $\tilde{l}(t)$ are served then if the service is not completed, (4.5) hold at slot $t + 1$. If the service is completed and we have an arrival at queues $s(t)$, $\tilde{s}(t)$ and no arrivals at $l(t)$, $\tilde{l}(t)$ then (4.4) hold at slot $t + 1$. If the service is completed and the arrivals are not as above then (4.5) hold at $t + 1$.

Case 3′: Relations (4.5) hold at t and $X_{\tilde{s}}(t) = \tilde{X}_{\tilde{l}}(t)$.

Let the connectivity variables at slot $t+1$ be as in Case 1′). If π serves queue $l(t)$ at slot $t+1$, let $\tilde{\pi}$ serve $\tilde{l}(t)$; if π serves queue $s(t)$ let $\tilde{\pi}$ serve $\tilde{s}(t)$. Let $\tilde{\pi}$ be identical to π, otherwise. We distinguish the following cases.

Case 3a: Queues $l(t)$, $\tilde{l}(t)$ are served under π, $\tilde{\pi}$, respectively.

Let the service variables of $l(t)$, $\tilde{l}(t)$ be identical under π, $\tilde{\pi}$. Let all queues other than j or k have the same arrivals under both policies. If service is not completed let queues $l(t)$, $\tilde{l}(t)$ have the same arrivals under π, $\tilde{\pi}$ respectively and similarly for queues $s(t)$, $\tilde{s}(t)$. If there is an arrival at $s(t)$, $\tilde{s}(t)$ and no arrival at $l(t)$, $\tilde{l}(t)$ at $t + 1$ then (4.4) hold at $t + 1$, otherwise

(4.5) hold. If service is completed let queues $s(t)$, $\tilde{l}(t)$ have the same arrivals at $t+1$ under π, $\tilde{\pi}$ and similarly for $l(t)$, $\tilde{s}(t)$. Then, (4.4) hold at $t+1$.

Case 3b: Queues $s(t)$, $\tilde{s}(t)$ are served under π, $\tilde{\pi}$, respectively.

Let the service variables of $s(t)$, $\tilde{s}(t)$ be identical under π, $\tilde{\pi}$. Let queues $l(t)$, $\tilde{l}(t)$ have the same arrivals under π, $\tilde{\pi}$ respectively, and similarly for the queues $s(t)$, $\tilde{s}(t)$. Let all queues $i \neq j$, k have the same arrivals under the two policies. If service is completed then (4.5) hold at $t+1$. If service is not completed then either (4.4) or (4.5) hold depending on whether there are arrivals at $s(t)$, $\tilde{s}(t)$ and no arrivals at $l(t)$, $\tilde{l}(t)$, or not.

Case 3c: Queue $i \neq j$, k is served under π, $\tilde{\pi}$, respectively.

Let queues $l(t)$, $\tilde{l}(t)$ under π, $\tilde{\pi}$, respectively have the same arrivals and similarly for the queues $s(t)$, $\tilde{s}(t)$. Let all queues $i \neq j$, k have the same arrivals under the two policies. Let queue i have the same service variables under π, $\tilde{\pi}$. If there is an arrival at $s(t)$, $\tilde{s}(t)$ and no arrival at $l(t)$, $\tilde{l}(t)$ at $t+1$ then (4.4) hold at $t+1$, otherwise (4.5) hold. □

We proceed now in the proof of the theorem.

Proof of Theorem 3: From Lemma 3, we have that for any policy π, we can construct a policy π_1 which is similar to LCQ at $t=1$ and such that for the corresponding total number of packets processes Q, Q_1, we have

$$Q_1(t) \leq Q(t) \text{ a.s.}, \qquad t = 0, 1, \cdots.$$

By repeating the construction, we can show that there exists a policy π_2 that agrees with π_1 at the first slot, agrees with LCQ in the second slot is such that for the corresponding process Q_2, we have

$$Q_2(t) \leq Q_1(t) \text{ a.s.}, \qquad t = 0, 1, \cdots.$$

If we repeat the argument k times, we obtain policies $\pi_i, i = 1, \cdots, k$ such that policy π_i agrees with LCQ at the first i slots and for the corresponding processes, we have

$$Q_k(t) \leq Q_{k-1}(t) \leq \cdots \leq Q_1(t) \leq Q(t) \text{ a.s.}, \qquad t = 0, 1, \cdots. \quad (4.6)$$

Consider the time slots $t_1, t_2, \cdots, t_n$ and a function g as in Theorem 2b). Consider also the policy π_{t_n} previously defined. By construction, the variables $Q_{t_n}(t_1), \cdots, Q_{t_n}(t_n)$ have the same joint probability distribution with the variables $Q_0(t_1), \cdots, Q_0(t_n)$ where Q_0, Q_{t_n} are the processes of total number of packets in the system under the policies LCQ, π_{t_n} respectively. Hence, for all z, we have

$$P(g(Q_{t_n}(t_1), \cdots, Q_{t_n}(t_n)) > z) = P(g(Q_0(t_1), \cdots, Q_0(t_n)) > z). \quad (4.7)$$

From (4.6), $Q_{t_n}(t) \leq Q(t)$ a.s. for all $t = 0, 1, \cdots$, therefore, we have

$$P(g(Q_{t_n}(t_1), \cdots, Q_{t_n}(t_n)) > z) \leq P(g(Q(t_1), \cdots, Q(t_n)) > z). \quad (4.8)$$

Equations (4.7) and (4.8) and Theorem 2b complete the proof. □

The longest connected queue is indeed the queue which is most unlikely to become empty among the connected ones. Hence, by serving the longest connected queue at the current slot, the LCQ policy minimizes in some sense the likelihood of having at some future slot only empty queues connected, in which case the server will be forced to idle.

Remark 5: The fact that policy π in Lemma 3 bases its decisions on the lengths of the connected queues only is not essential in the proof of the lemma. That proof goes through even if π is any policy that bases its decision on the history of the lengths of all queues in the system in addition to the connectivities and past control actions. Therefore, LCQ is optimal within the class of policies that base their decisions on the complete system history.

When the connectivities are fixed ($C_i(t) = 1, i = 1, \cdots, N$, $t = 1, \cdots$) then in the symmetric system any work conserving policy minimizes the delay. Furthermore in the general case (asymmetric system), if the service processes are i.i.d. (geometric service requirements) the optimal policy is known to be the one that serves the nonempty queue with largest m_i. In the case of varying connectivities, work conservation is not enough for optimality. Serving the queue with the largest backlog is essential for optimal system performance.

A related result is the characterization of the worst work conserving policy. This is the *shortest connected queue* (SCQ) policy which allocates the server at slot t to the connected, nonempty queue with minimum length. The following theorem states that policy SCQ maximizes in the stochastic order sense the process of total number of packets in the system within the class of work conserving policies.

Theorem 4: If Q is the process of total number of packets in the system when the initial state is x_0 and a work conserving policy π acts on it and Q' the corresponding process when SCQ acts on the system, then we have

$$Q \leq_{st} Q'.$$

Proof: It is analogous to the proof of Theorem 3 and it is not repeated here. □

Apparently SCQ has no practical significance since it maximizes the delay. The result though in Theorem 4 emphasizes the fact that serving queues with large backlog improves the delay. If we consider a hierarchy of the work conserving policies with respect to how close they follow the rule to serve queues with large backlogs then LCQ is in the top of this hierarchy and SCQ in the bottom. It is intuitively appealing that their delay performances are the best and worst, respectively, within the class of work conserving policies.

B. Discussion

Assume that the arrival, service and connectivity processes are i.i.d. In this case, the problem of minimizing the delay can be casted as a discrete-time Markov Decision Process. Consider the cost function defined by

$$J_\pi(\boldsymbol{x}_0) = \limsup_{T\to\infty} E^\pi_{\boldsymbol{x}_0}\left[\frac{1}{T}\sum_{t=0}^{T}\sum_{i=1}^{N} c_i X_i(t)\right], \qquad (4.9)$$

where π is a policy in $\tilde{G}$, $\boldsymbol{x}_0$ is the initial system state and the expectation is taken with respect to the probability measure induced by π when the system starts from $\boldsymbol{x}_0$. Minimizing the delay is equivalent to minimizing (4.9) within $\tilde{G}$ when all c_i are equal. When the c_i's are unequal the minimization of (4.9) corresponds to minimization of a weighted average delay. This minimization problem falls within the category of discrete-time Markov Decision Processes (MDP) with partial observations [6]. The controlled Markov chain is $(\boldsymbol{X}(t-1), \boldsymbol{C}(t))$, the control action is $\boldsymbol{U}(t)$ and the evolution of the chain is governed by (2.1). The observation at time t is $\boldsymbol{Y}(t) = (\boldsymbol{X}(t-1) \otimes \boldsymbol{C}(t), \boldsymbol{C}(t))$. The optimal policies in MDP with partial observations are in general nonstationary since the action taken at slot t is a function of all past observations. Those policies are usually hard to specify. The optimality result obtained in Section IV-A implies that LCQ minimizes (4.9). Therefore, in a symmetric system, the policy that minimizes (4.9) is stationary. In a general asymmetric system, the optimization of (4.9) remains an open problem. We conjecture that the optimal server allocation policy is stationary in the general case as well. We are lead to that conjecture by the observation that the control action at slot t can affect the connected queues only and since the arrivals, services and connectivities are assumed i.i.d., all the relevant control information about these queues is contained in their current lengths. Therefore, the situation is analogous to the complete observation case where the optimal policy is stationary according to known results in MDP theory. Nevertheless, we believe that a μc rule type of policy can not be optimal and the allocation decisions of the optimal policy are a complicated function of the state; therefore the policy is difficult to be specified completely.

To get some intuition on why a μc rule type of policy can not be optimal, consider the system in counterexample 1, with the cost function (4.9) with costs $c_1 > c_2$. According to the μc rule, queue 1 has priority over queue 2 irrespectively of the queue lengths. Assume that c_1 is slightly greater than c_2 and p_2 is small. Consider the case where at slot t queue 1 has 1 packet while queue 2 has several packets and both queues are connected. If we serve queue 1 at that slot then the instantaneous cost will be lower by $c_1 - c_2$ compared to the instaneous cost if we serve queue 2. At slot $t+1$ though, queue 1 will be empty with probability $1 - a_1$ and if queue 2 is disconnected then no service will be provided. If queue 2 is served at slot t then at slot $t+1$ we will be able to serve queue 1. Hence, if $c_1 - c_2$ is sufficiently small, yet greater than zero, then by serving queue 2 at slot t we can achieve better overall cost.

In our study, we have assumed that the connectivities become available for decision making in the beginning of each slot. An interesting case is when the connectivities are not observable and the server at slot t is allocated based on the queue lengths $\boldsymbol{X}(t-1)$ only. If the connectivity processes are i.i.d. then the changing connectivity model is reduced to one with fixed connectivities where the service variable for queue i at slot t is $C_i(t)M_i(t)$.

V. Optimization of Throughput and Delay in a Finite Buffer System

When the buffers in the nodes have finite length then an arriving packet is blocked from admission when it finds the buffers full. The number of packets which are successfully transmitted, that is the throughput of the system, is an important performance measure in addition to delay. In this section, we study both throughput and delay performance in a finite buffer system with one buffer per node. A policy is obtained that is both throughput and delay optimal.

When there is a single buffer per node an arriving packet at node i during slot t is accepted if the buffer is empty in the beginning of the slot $(X_i(t-1) = 0)$ or node i is the one selected for service and the service is successful at slot t in which case its packet is forwarded from its buffer in the beginning of the slot and the buffer is empty. The queue length vector in this case belongs to $\{0, 1\}^N$; the queue length at node i evolves according to the equation

$$X_i(t) = \max\{A_i(t),\, X_i(t-1) \cdot (1 - 1\{U(t) = i\}C_i(t)M_i(t))\}, \qquad (5.1)$$

where the variables $X_i(t)$, $U(t)$, $C_i(t)$, $M_i(t)$, $A_i(t)$ are as defined in Section II. Our results in this section are obtained under the following statistical assumption.

A3: At each slot there can be at most one arrival at each node, that is the variables $A_i(t), i = 1, \cdots, N$, are binary; furthermore we assume that the arrival and service processes have identical statistics at different nodes.

The number of packets blocked from admission into the system during slot t is

$$B(t) = \sum_{i=1}^{N} A_i(t)X_i(t-1)(1 - 1\{U(t) = i\}C_i(t)M_i(t)).$$

Note that the number of packets blocked from admission into the system plus the number of packets which are admitted in the system during slot t and they are finally served is equal to the number of packets arrived during slot t. Therefore, maximizing the throughput of the system is equivalent to minimizing the number of blocked packets. Consider the policy $\hat{\pi} \in G$, which during slot t allocates the server according to the function $\hat{g}$: $\mathcal{Y}^1 \to \{e, 1, \cdots, N\}$ defined by

$$\hat{g}(\boldsymbol{x}, \boldsymbol{c}) = \begin{cases} e, & \text{if } x_i c_i = 0, \\ & \quad i = 1, \cdots, N, \\ \underset{\substack{i=1,\cdots,N \\ x_i c_i > 0}}{\arg\min}\{p_i\}, & \text{otherwise.} \end{cases}$$

That is, $\hat{\pi}$ allocates the server at slot t to the connected nonempty queue $i(C_i(t) = 1)$ with the smallest probability of

being connected. Policy $\hat{\pi}$ minimizes in the stochastic ordering sense both the process of blocked packets and the process of total number of packets in the system.

Theorem 5: Consider an arbitrary policy $\tilde{\pi} \in G$ and let policies $\hat{\pi}$ and $\tilde{\pi}$ schedule transmissions starting from the same initial state $\boldsymbol{x}$ at $t = 0$. Let Q, B be the processes of the total number of packets in the system and of the blocked packets respectively under $\hat{\pi}$; let $\tilde{Q}$, $\tilde{B}$ be the corresponding processes under $\tilde{\pi}$. Then, we have

$$Q \leq_{st} \tilde{Q}, \tag{5.2}$$

$$B \leq_{st} \tilde{B}. \tag{5.3}$$

Proof: We construct the queue length processes $\boldsymbol{X}$, $\tilde{\boldsymbol{X}}$ under $\hat{\pi}$, $\tilde{\pi}$, respectively, by appropriate coupling of the arrivals services and connectivities such that

$$Q(t) \leq \tilde{Q}(t) \text{ a.s.}, \qquad t = 0, 1, \cdots, \tag{5.4}$$

$$B(t) \leq \tilde{B}(t) \text{ a.s.}, \qquad t = 0, 1, \cdots. \tag{5.5}$$

Hence, (5.2), (5.3) follows.

We show that a particular partial ordering (defined next) holds between the system states under the two policies at every slot. This partial ordering implies relations (5.4), (5.5). Assume that the queues are indexed so that $p_i \leq p_{i+1}, i = 1, \cdots, N-1$. We say $\boldsymbol{x} \prec \boldsymbol{y}$, $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^N$ if

$$\sum_{i=1}^{j} x_i \leq \sum_{i=1}^{j} y_i, \qquad j = 1, \cdots, N. \tag{5.6}$$

We construct the queue length processes such that for all $\tau = 0, 1, \cdots$, we have

$$\boldsymbol{X}(\tau) \prec \tilde{\boldsymbol{X}}(\tau). \tag{5.7}$$

We use forward induction. At $\tau = 0$ we have $\boldsymbol{X}(0) = \tilde{\boldsymbol{X}}(0)$; therefore for $\tau = 0$, (5.7) follows. Assume that (5.7) is true for $\tau = t$; we show that it is true for $\tau = t + 1$ as well. Let $A_i(t+1)$, $C_i(t+1)$, $M_i(t+1)$, $i = 1, \cdots, N$, $U(t+1)$ be the arrival, connectivity, service, and control variables under $\hat{\pi}$ and $\tilde{A}_i(t+1)$, $\tilde{C}_i(t+1)$, $\tilde{M}_i(t+1)$, $i = 1, \cdots, N$, $\tilde{U}(t+1)$ under $\tilde{\pi}$. First, we show that the following hold

$$\boldsymbol{Y}(t+1) \prec \tilde{\boldsymbol{Y}}(t+1), \tag{5.8}$$

where

$$Y_i(t+1) = X_i(t)(1 - 1\{U(t+1) = i\} \cdot C_i(t+1)M_i(t+1)), \qquad i = 1, \cdots, N,$$

$$\tilde{Y}_i(t+1) = \tilde{X}_i(t)(1 - 1\{U(t+1) = i\} \cdot \tilde{C}_i(t+1)\tilde{M}_i(t+1)), \qquad i = 1, \cdots, N.$$

Let $j(l)$, $\tilde{j}(l)$ be the lth nonempty queue starting from queue 1 in states $\boldsymbol{X}(t)$, $\tilde{\boldsymbol{X}}(t)$. If $\tilde{j}(l) > j(l)$ then we have $\sum_{i=1}^{j} X_i(t) = l > \sum_{i=1}^{j} \tilde{X}_i(t)$, which contradicts the induction hypothesis; therefore, we have

$$\tilde{j}(l) \leq j(l), \qquad l = 1, \cdots, Q(t), \tag{5.9}$$

and, by the assumption about the indexing of the queues,

$$p_{\tilde{j}(l)} \leq p_{j(l)}, \qquad l = 1, \cdots, Q(t). \tag{5.10}$$

Because of (5.10), we may construct $C_{j(l)}(t+1)$, $\tilde{C}_{\tilde{j}(l)}(t+1)$ in a common probability space such that

$$\tilde{C}_{\tilde{j}(l)}(t+1) = 1 \Rightarrow C_{j(l)}(t+1) = 1, \qquad l = 1, \cdots, Q(t).$$

We distinguish the following cases.

Case 4: No nonempty queue is connected at $t+1$ under $\hat{\pi}$.

In this case, no queue is served and we have

$$\sum_{i=1}^{j} Y_i(t+1) = \sum_{i=1}^{j} X_i(t), \qquad j = 1, \cdots, N. \tag{5.11}$$

Since $C_{j(l)}(t+1) = 0$, $l = 1, \cdots, Q(t)$, and because of the coupling of the connectivities, we have $\tilde{C}_{\tilde{j}(l)}(t+1) = 0$, $l = 1, \cdots, Q(t)$; therefore no queue with index $j \leq \tilde{j}(Q(t))$ is served and we have

$$\sum_{i=1}^{j} \tilde{Y}_i(t+1) = \sum_{i=1}^{j} \tilde{X}_i(t), \qquad j \leq \tilde{j}(Q(t)), \tag{5.12}$$

$$\sum_{i=1}^{j} \tilde{Y}_i(t+1) \geq Q(t), \qquad j > \tilde{j}(Q(t)). \tag{5.13}$$

From (5.11), (5.12), and the induction hypothesis, we have $\sum_{i=1}^{j} Y_i(t+1) \leq \sum_{i=1}^{j} \tilde{Y}_i(t+1)$ for $j \leq \tilde{j}(Q(t))$, and from (5.11), (5.13), we have $\sum_{i=1}^{j} Y_i(t+1) = Q(t) \leq \sum_{i=1}^{j} \tilde{Y}_i(t+1)$ for $j > \tilde{j}(Q(t))$. Hence, relation (5.8) holds.

Case 5: Some nonempty queue is connected at $t+1$ under $\hat{\pi}$.

If no queue is served under $\tilde{\pi}$ during $t+1$ then $\tilde{Y}_i(t+1) = \tilde{X}_i(t)$, $i = 1, \cdots, N$, while $Y_i(t+1) \leq X_i(t)$, $i = 1, \cdots, N$. Therefore, (5.8) follows from the induction hypothesis for $\tau = t + 1$. If some queue is served under both policies, then assign the same service variables to the queues that are being served under both policies. If service is not completed, then $\boldsymbol{Y}(t+1) = \boldsymbol{X}(t)$, $\tilde{\boldsymbol{Y}}(t+1) = \boldsymbol{X}(t)$, and (5.8) follows from the induction hypothesis. If service is completed at $t+1$, then let $j_0 = j(l_0)$, $\tilde{j}_0 = \tilde{j}(\tilde{l}_0)$ be the queues served under $\hat{\pi}$ and $\tilde{\pi}$, respectively. Since $\hat{\pi}$ serves the nonempty queue with the smallest probability of being connected, we have

$$C_{j(l)}(t+1) = 0, \qquad 1 \leq l < l_0.$$

From the coupling of the connectivities it is implied that

$$C_{\tilde{j}(l)}(t+1) = 0, \qquad 1 \leq l < l_0,$$

and we have

$$\tilde{j}_0 \geq \tilde{j}(l_0). \tag{5.13a}$$

If $\tilde{j}_0 \geq j_0$, then for $j \geq j_0$, we have

$$\sum_{i=1}^{j} Y_i(t+1) = \sum_{i=1}^{j} X_i(t) - 1 \leq \sum_{i=1}^{j} \tilde{X}_i(t) - 1 \leq \sum_{i=1}^{j} \tilde{Y}_i(t+1), \tag{5.14}$$

and for $j > j_0$, we have

$$\sum_{i=1}^{j} \tilde{Y}_i(t+1) = \sum_{i=1}^{j} \tilde{X}_i(t) \geq \sum_{i=1}^{j} X_i(t) = \sum_{i=1}^{j} Y_i(t+1). \tag{5.15}$$

From (5.14) and (5.15), (5.8) follows. If $\tilde{j}_0 < j_0$, then for $j < \tilde{j}_0$ we have

$$\sum_{i=1}^{j} \tilde{Y}_i(t+1) = \sum_{i=1}^{j} \tilde{X}_i(t) \geq \sum_{i=1}^{j} X_i(t) = \sum_{i=1}^{j} Y_i(t+1). \tag{5.16}$$

For $j \geq j_0$, we have

$$\sum_{i=1}^{j} Y_i(t+1) = \sum_{i=1}^{j} X_i(t) - 1 \leq \sum_{i=1}^{j} \tilde{X}_i(t) - 1 = \sum_{i=1}^{j} \tilde{Y}_i(t+1). \tag{5.17}$$

For $\tilde{j}_0 \leq j < j_0$ and because of (5.13a), we have

$$\sum_{i=1}^{j} Y_i(t+1) = \sum_{i=1}^{j} X_i(t) \leq l_0 - 1 \leq \sum_{i=1}^{j} \tilde{X}_i(t) - 1 = \sum_{i=1}^{j} \tilde{Y}_i(t+1). \tag{5.18}$$

From (5.16), (5.17), (5.18), we get (5.8) for $\tau = t+1$.

Now, given that (5.8) holds, we show that (5.7) holds at $t+1$. Let $\tilde{m}(l)$, $m(l)$ be the lth empty queue starting from queue 1 for the states $\boldsymbol{Y}(t+1)$, $\boldsymbol{Y}(t+1)$, respectively. Let $Q'(t+1)$, $\tilde{Q}'(t+1)$ be the number of packets in the system when the states are $\boldsymbol{Y}(t+1)$, $\tilde{\boldsymbol{Y}}(t+1)$, respectively. We couple the arrivals under the two policies such that

$$A_{m(l)}(t+1) = \tilde{A}_{\tilde{m}(l)}(t+1), \qquad l = 1, \cdots, N - \tilde{Q}'(t+1).$$

Consider an arbitrary queue j and let k and $\tilde{k}$ be the number of empty queues with index less than or equal to j under $\hat{\pi}$ and $\tilde{\pi}$, respectively. Because of (5.8), we have $k \geq \tilde{k}$. We get

$$\sum_{i=1}^{j} \tilde{X}_i(t+1) = \sum_{i=1}^{j} \tilde{Y}_i(t+1) + \sum_{l=1}^{\tilde{k}} \tilde{A}_{\tilde{m}(l)}(t+1), \tag{5.19}$$

$$\sum_{i=1}^{j} X_i(t+1) = \sum_{i=1}^{j} Y_i(t+1) + \sum_{l=1}^{k} A_{m(l)}(t+1), \tag{5.20}$$

$$\sum_{i=1}^{j} \tilde{Y}_i(t+1) - \sum_{i=1}^{j} Y_i(t+1) = k - \tilde{k}. \tag{5.21}$$

From the coupling of the arrivals, we have

$$\sum_{i=1}^{k} A_{m(l)}(t+1) - \sum_{i=1}^{\tilde{k}} A_{\tilde{m}(l)}(t+1) = \sum_{l=\tilde{k}1}^{k} A_{m(l)}(t+1) \leq k - \tilde{k}. \tag{5.22}$$

Subtracting (5.20) from (5.19) and replacing from (5.21), (5.22), we get

$$\sum_{i=1}^{j} \tilde{X}_i(t+1) - \sum_{i=1}^{j} X_i(t+1) \geq 0, \qquad j = 1, \cdots, N.$$

Hence, (5.7) holds for $\tau = t+1$. Notice that

$$\boldsymbol{X}(t) \prec \tilde{\boldsymbol{X}}(t) \Rightarrow Q(t) \leq \tilde{Q}(t), \qquad t = 1, \cdots.$$

Therefore (5.7) implies (5.2).

Now we show (5.3). Let $j'(l)$ and $\tilde{j}'(l)$ be the lth nonempty queues, starting from queue 1, for the states $\boldsymbol{Y}(t+1)$ and $\tilde{\boldsymbol{Y}}(t+1)$, respectively. Couple the arrivals at $t+1$ as follows:

$$A_{j'(l)}(t+1) = \tilde{A}_{\tilde{j}'(l)}(t+1), \quad l = 1, \cdots, Q'(t+1). \tag{5.23}$$

For the number of blocked packets, we have

$$B(t+1) = \sum_{l=1}^{Q'(t+1)} A_{j(l)}(t+1),$$

$$\tilde{B}(t+1) = \sum_{l=1}^{\tilde{Q}'(t+1)} \tilde{A}_{\tilde{j}(l)}(t+1),$$

and from (5.23),

$$\tilde{B}(t+1) - B(t+1) = \sum_{l=1}^{\tilde{Q}'(t+1)} \tilde{A}_{\tilde{j}(l)}(t+1) - \sum_{l=1}^{Q'(t+1)} A_{j(l)}(t+1) = \sum_{l=Q'(t+1)}^{\tilde{Q}'(t+1)} \tilde{A}_{\tilde{j}(l)}(t+1) \geq 0;$$

therefore (5.3) holds. □

Regarding the heuristic interpretation of $\hat{\pi}$, the argument is similar to that for the LCQ policy. We would like to minimize the likelihood of having in a slot only empty queues connected. By serving the connected queue with the smallest probability of being connected we achieve exactly that.

VI. Conclusion

A single-hop radio network with randomly changing connectivity has been considered in this paper. Its stability properties have been characterized and a policy that minimizes the delay has been obtained. In the case of finite buffers, the policy that maximizes the throughput and minimizes the delay has been obtained, too. Time varying connectivity

arises in communication networks whenever the quality of the communication link changes with time. This phenomenon is inherent in several types of radio networks as was mentioned in the introduction. In view of the changing connectivity, the resource allocation problem becomes more challenging than in fixed connectivity systems, as indicated by the results reported in this paper. There are several open problems for further investigation related to the issue of changing connectivity; we discuss few of them next.

An interesting variation of the problem we studied is the case where the connectivity information is not available for decision making and the server allocation can be based on the queue lengths, the arrivals, and the departures. If the connectivity variables in different slots are independent then, as we mentioned in Section IV, the server allocation problem under no connectivity information is equivalent to a server allocation problem in a fixed connectivity system. If the connectivities at neighboring time slots are statistically dependent then the problem of optimal allocation becomes more complicated. The queue lengths are not a state any more and the problem should be casted as a partially observable Markov Decision Process (under the appropriate independence assumptions on the arrivals and services). The study of stability and optimal delay performance in the latter case of dependent connectivities are open problems for further investigation.

In our study, we have assumed that each queue is either connected to the server or not, that is, the connectivities are binary variables. In certain cases, that assumption is inappropriate and the connectivity should be represented by a multivalued variable where the different values correspond to different connectivity qualities. It is of interest to study the resource allocation problem under the assumption of multivalued connectivities.

ACKNOWLEDGMENT

L. Tassiulas would like to thank A. Makowski for useful discussions on the subject of this paper and P. Bhattacharya for comments and suggestions on an earlier draft of the paper. Finally, the comments of the reviewers improved the presentation of the material and they are acknowledged.

REFERENCES

[1] S. Asmussen, *Applied Probability and Queues*. New York: John Wiley, 1987.

[2] J. Baras, D. J. Ma, and A. Makowski, "K competing queues with geometric service requirements and linear costs: The μc rule is always optimal," *Syst. Contr. Lett.*, vol. 6, pp. 173–180, 1985.

[3] C. Buyukkoc, P. Varaiya, and J. Walrand, "The μc rule revisited," *Adv. Appl. Prob.*, vol. 17, pp. 234–235, 1985.

[4] Y. Chandramouli, M. F. Neuts, and V. Ramaswami, "A queueing model for meteor burst packet communication systems," *IEEE Trans. Commun.*, vol. 37, Oct. 1989.

[5] D. Goodman, "Cellular packet communications," *IEEE Trans. Commun.*, vol. 38, Aug. 1990.

[6] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification and Adaptive Control*. Englewood Cliffs, NJ: Prentice Hall, 1986.

[7] R. G. Ogier, "Minimum-delay routing in continuous-time dynamic networks with piecewise-constant capacities," *Networks*, vol. 18, pp. 303–318, 1988.

[8] A. Orda and R. Rom, "Shortest path and minimum-delay algorithms in networks with time-dependent edge length," *J. ACM*, vol. 37, pp. 607–625, 1990.

[9] ——, "Minimum weight paths in time-dependent networks," *Networks*, vol. 21, pp. 295–320, 1991.

[10] R. Steel and V. K. Prabhu, "Mobile radio cell structure for high user density and large data rates," *Proc. IEE*, pt. F, no. 5, pp. 396–404, Aug. 1985.

[11] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models*. New York: John Wiley, 1983.

[12] J. Walrand, *Queueing Networks*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[13] D. Yavuz, "Meteor burst communications," *IEEE Commun. Mag.*, vol. 28, no. 9, Sept. 1990.