Optimal Transmission Scheduling in Symmetric Communication Models With Intermittent Connectivity

Anand Ganti, Eytan Modiano, *Senior Member, IEEE*, and John N. Tsitsiklis, *Fellow, IEEE*

***Abstract*—We consider a slotted system with N queues, and independent and identically distributed (i.i.d.) Bernoulli arrivals at each queue during each slot. Each queue is associated with a channel that changes between "on" and "off" states according to i.i.d. Bernoulli processes. We assume that the system has K identical transmitters ("servers"). Each server, during each slot, can transmit up to C packets from each queue associated with an "on" channel. We show that a policy that assigns the servers to the longest queues whose channel is "on" minimizes the total queue size, as well as a broad class of other performance criteria. We provide several extensions, as well as some qualitative results for the limiting case where N is very large. Finally, we consider a "fluid" model under which fractional packets can be served, and subject to a constraint that at most C packets can be served in total from all of the N queues. We show that when $K = N$, there is an optimal policy which serves the queues so that the resulting vector of queue lengths is "Most Balanced" (MB).**

***Index Terms*—Longest-queue-first, minimum-delay scheduling, stochastic coupling, transmission scheduling, wireless channel.**

I. Introduction

Wireless and satellite nodes are often limited to a small number of transmitters and channels, and these have to be allocated to users in the face of competing demands. For example, satellite systems employ hundreds or even thousands of narrow beams over which information can be transmitted at high data rates. Each of the downlink beams covers a different region within the satellite's footprint. Data packets to be transmitted along the different beams arrive at the satellite, either from the ground or from neighboring satellites, and are stored in on-board buffers. In this context, there is often only a limited number of on-board transmitters, so that not all beams can be served by the transmitters simultaneously. This gives rise to a scheduling problem involving the allocation of the transmitters to the different downlink beams. Further complicating matters is the fact that, due to weather and atmospheric conditions, the transmission rate along the different beams varies with time; hence, the quality of the links must be taken into account in making scheduling decisions. For a different context, a wireless base station typically has far fewer channels available for transmissions than the number of users to be served. Again, this raises a nearly identical problem of allocating channels to the different users. Problems of this type have received much attention recently [1]–[5], [8], [10], [11], sometimes motivated by next-generation wireless data systems.

In this paper, we focus on the special case where all arrival streams and channel-state processes are statistically identical. This symmetry sometimes leads to rather simple optimal policies, although their optimality can be hard to establish. We model the system as a discrete-time queueing system, with arrivals and channel states described by independent Bernoulli processes. More specifically, we assume that the numbers of arrivals to the ith queue during the nth slot, denoted by $A_i(n)$, are independent Bernoulli random variables, with the same parameter for all i and n. Furthermore, we assume that the state of the ith channel during the nth slot, denoted by $G_i(n)$, can only take one of two values, namely, 0 or 1. We designate 0 as the "off" state, and 1 as the "on" state. When the channel is in the "off" state, no transmission is possible. When the channel is in the "on" state, the channel can be utilized, and we will be saying that the corresponding queue is "connected." Again, we assume that the $G_i(n)$ are independent and identically distributed (i.i.d.) Bernoulli random variables, which are also independent from the arrival processes. Finally, we let $B_i(n)$ represent the number of packets in the ith queue at the beginning of time slot n.

We assume that there are K transmitters ("servers"). Each server can only serve one queue at a time, and can only be assigned to a connected queue. At each slot, each server can transmit up to C packets, where the number C reflects power limitations or other constraints on an individual transmitter. Fig. 1 depicts the system considered in this paper.

Such queueing systems, with multiple queues and stochastically varying connectivities, have been studied in [11] where the authors use a coupling argument to establish that when $K = 1$ (single transmitter) and $C = 1$ (one packet per slot), a policy that serves the longest connected queue (LCQ) maximizes the stability region of the system, and also results in optimal average queue lenghts. Furthermore, [1] shows that the LCQ policy results in a maximal stability region under more general assumptions on the arrival and channel state processes.

Manuscript received July 14, 2004; revised November 3, 2006. This work was supported in part by the National Science Foundation under Grant NCR-9627610, by DARPA under the Next Generation Internet Initiative, and by the ARO under Grant DAAD10-00-10466. The material in this paper was presented in part at the 40th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, October 2002.

A. Ganti is with the Sandia National Laboratories, Albuquerque, NM 87185 USA (e-mail: aganti@sandia.gov).

E. Modiano and J. N. Tsitsiklis are with The Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: modiano@mit.edu; jnt@mit.edu).

Communicated by G. Sasaki, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2006.890695

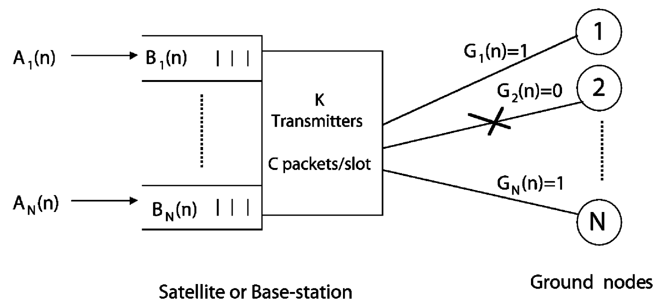

Fig. 1. The symmetric transmission model.

In Section II, we generalize the result in [11] and establish the optimality of a policy that allocates the servers to the longest connected queues, for general K and C. In Section III, we discuss extensions to various settings where our i.i.d. assumptions are violated. In Section IV, we consider the case where the number N of queues increases, while the total arrival rate and the number of transmitters is held constant. We show, under a slightly different arrival model (Poisson instead of Bernoulli arrivals), that the average queue size is a bounded function of N, under a fairly simple heuristic policy. In Section V, we consider a "fluid" model under which fractional packets can be served, and subject to a constraint that at most C packets can be served in total over all of the N queues. We show that when $K = N$, there is an optimal policy that serves the queues so that the resulting vector of queue lengths is "Most Balanced." Finally, Section VI contains some brief concluding remarks.

II. Optimality of LCQ Policies

We start with a precise description of the problem, together with some notation. We use $B(n)$ to denote the vector $(B_1(n), \ldots, B_N(n))$ of queue lengths at the beginning of time slot n. Similarly, $A(n) = (A_1(n), \ldots, A_N(n))$ is the vector with the number of arrivals at each queue during time slot n, and $G(n) = (G_1(n), \ldots, G_N(n))$ is the vector of channel connectivities at time slot n. Finally, $W(n) = (W_1(n), \ldots, W_N(n))$ is the vector of packets withdrawn from each queue during time slot n. The dynamics of the system are described by the equation

$$B(n+1) = B(n) - W(n) + A(n), \qquad n = 1, 2, \ldots. \quad (1)$$

A policy π is a family of mappings that, for any i and for any time n, determines $W_i(n)$ as a nonnegative integer function of the past history and present state

$$\mathcal{H}(n) = (B(1), G(1), A(1), \ldots, G(n-1), A(n-1), G(n)),$$

and satisfies

$$W_i(n) \leq G_i(n) \min\{C, B_i(n)\}.$$

We also introduce the history

$$\mathcal{H}_-(n) = (B(1), G(1), A(1), \ldots, G(n-1), A(n-1))$$

until just before the channel states are to be observed, as well as the history

$$\mathcal{H}_+(n) = (B(1), G(1), A(1), \ldots, G(n-1), A(n-1), G(n), W(n))$$

until just before the new arrivals $A(n)$ are to be observed. We make the following assumption.

Assumption 1:

(a) For every n and h, and conditioned on $\mathcal{H}_-(n) = h$, the random variables $G_i(n)$, $i = 1, \ldots, N$, are independent and Bernoulli, with a parameter which is the same for all i, n, h.

(b) For every n and h, and conditioned on $\mathcal{H}_+(n) = h$, the random variables $A_i(n)$, $i = 1, \ldots, N$, are independent and Bernoulli, with a parameter which is the same for all i, n, h.

Let us now make a few remarks. The definition of the history $\mathcal{H}(n)$ implies the following sequence of events. The queue connectivities $G(n)$ are observed, then the packet withdrawals $W(n)$ are determined, and finally, the new arrivals $A(n)$ occur and determine the next queue lengths $B(n+1)$. The policy π is allowed to be non-Markovian, because this facilitates the proof of our subsequent results. Finally, the constraints $W_i(n) \leq G_i(n) \min\{C, B_i(n)\}$ reflect our interpretation of $G_i(n)$ and C: if $G_i(n) = 0$, the ith queue cannot be served; otherwise, the number of packets that can be withdrawn is limited by the number $B_i(n)$ of available packets, as well as the "server capacity" C. Finally, note that we do not include past decisions $W(\cdot)$ or queue lengths $B(\cdot)$ in the history, since under any given policy, past decisions and queue lengths can be recovered from the history.

A. LCQ Policies and an Ordering on Configurations

We say that a policy is a LCQ ("Longest Connected Queues") policy if it operates as follows. Consider the set of queues i that are "connected," i.e., $G_i(n) = 1$. Out of that set, select up to K queues with the largest values of $B_i(n)$, and serve $\min\{B_i(n), C\}$ packets from each one of them. Note that fewer than K queues can be served if and only if the number of nonempty connected queues is smaller than K. Moreover, in the event that there are multiple queues with equal values of $B_i(n)$ competing for a server, the policy can choose between them arbitrarily.

We wish to establish that LCQ policies are optimal for a wide class of performance criteria, expected total queue size being one of them. The intuitive reason is that a LCQ policy tries to keep the queued packets spread over multiple queues and has a better chance of avoiding idling when some channels are off. Using dynamic programming language, the cost-to-go of a particular "configuration" $B(n)$ is lower when the packets are more spread out to different queues. Our first step is to provide a mathematical definition of "more spread out," that is suitable for our purposes.

Let $\mathcal{Z}_+$ be the set of nonnegative integers, and let $\mathcal{Z}_+^N$ be the Cartesian product of N copies of $\mathcal{Z}_+$. Given two vectors

$b = (b_1, \ldots, b_N)$ and $\tilde{b} = (\tilde{b}_1, \ldots, \tilde{b}_N)$ in $\mathcal{Z}_+^N$, we write $\tilde{b} \leq b$ if we have $\tilde{b}_i \leq b_i$ for all i. The relation $\leq$ defines a partial order on the set $\mathcal{Z}_+^N$. We now introduce another relation $\sqsubseteq$ on $\mathcal{Z}_+^N$. (This relation depends on C, but we will avoid the more accurate notation $\sqsubseteq_C$ because C will be clear from the context.) We will write $\tilde{b} \sqsubseteq b$ if one of the following three relations hold:

(i) $\tilde{b} = b$;

(ii) b and $\tilde{b}$ differ in only two components, say i and j, and $\tilde{b}_i = b_j, \tilde{b}_j = b_i$;

(iii) b and $\tilde{b}$ differ in only two components, say i and j, and there exists a positive integer k such that

$$b_j \leq \min\{\tilde{b}_i, \tilde{b}_j\} \leq \max\{\tilde{b}_i, \tilde{b}_j\} \leq b_i$$
$$\tilde{b}_i = b_i - kC, \quad \tilde{b}_j = b_j + kC.$$

In Case (ii), the vectors b and $\tilde{b}$ are permutations of each other, and can be viewed as equivalent. In Case (iii), $\tilde{b}$ is "more balanced" than b, and can be obtained from b by moving a multiple of C packets from a larger component of b to a smaller one. (We refer to such a change from b to $\tilde{b}$ as a "balancing C-interchange.") For example, if $C = 5$ and we start with $b = (10, 3)$, we can move $C = 5$ packets from the largest component to the smaller one to obtain $\tilde{b} = (5, 8) \sqsubseteq (10, 3)$. However, if we were to move $2C = 10$ packets, to obtain the new vector $\tilde{b} = (0, 13)$, this is not a balancing C-interchange, because $13 > \max\{b_1, b_2\} = 10$.

We finally define a partial order $\preceq$ on $\mathcal{Z}_+^N$, by taking the union of the partial order $\leq$ and the relation $\sqsubseteq$ defined above, and forming their transitive closure. That is, $\tilde{b} \preceq b$ if and only if there exists a sequence $b^1, \ldots, b^l$ such that $b^1 = b$, $b^l = \tilde{b}$, and for $i = 1, \ldots, l-1$, we have either $b^{i+1} \leq b^i$ or $b^{i+1} \sqsubseteq b^i$. In words, we have $\tilde{b} \preceq b$ ("$\tilde{b}$ is preferable to b"), if we can form $\tilde{b}$ by starting from b and performing a sequence of operations, where each operation is a permutation of two components, a balancing C-interchange, or a removal of some packets. We now introduce the class of performance objectives that we will consider.

Let $\mathcal{F}$ be the class of real-valued functions on $\mathcal{Z}_+^N$ that are monotone nondecreasing with respect to the partial order $\preceq$; that is, $f \in \mathcal{F}$ if and only if

$$\tilde{b} \preceq b \text{ implies } f(\tilde{b}) \leq f(b).$$

Thus, $\mathcal{F}$ consists of those functions f that are nondecreasing, permutation-invariant,[1] and whose value does not increase if a balancing C-interchange is performed on its argument.

The class $\mathcal{F}$ contains the function $f(b) = b_1 + \cdots + b_N$, which corresponds to the total queue size. The class $\mathcal{F}$ also contains any function which is permutation-invariant, nondecreasing, and convex. In fact, it suffices that f be convex only on one-dimensional sets of the form

$$S_{ij}(b) = \{\tilde{b} \in \mathcal{Z}_+^N \mid \tilde{b}_i + \tilde{b}_j = b_i + b_j, \text{ and } \tilde{b}_k = b_k \text{ for } k \neq i, j\}.$$

In the special case where $C = 1$, the class $\mathcal{F}$ is closely related to the class of "increasing Schur-convex functions" [7], the main difference being that we are restricting to integer arguments. We note that when $C > 1$, the resulting class $\mathcal{F}$ is strictly larger than the class obtained with $C = 1$, as will be shown at the end of this section.

[1]A permutation is defined as a bijection of $\{1, \ldots, N\}$ onto itself. A function f is said to be permutation-invariant if $f(b_{\sigma(1)}, \ldots, b_{\sigma(N)}) = f(b_1, \ldots, b_N)$, for every permutation σ and every b.

B. The Dynamic Coupling Method

Our proof involves a dynamic coupling argument. We outline the structure of this method, and provide some related definitions. Consider the process under some policy π, for a particular sample path ω of the underlying sequence of random variables $(B(1), G(1), A(1), G(2), A(2), \ldots)$. Note that under the given policy π, the values of the sequences $\{B(n)\}$ and $\{W(n)\}$ are completely determined by ω. A *dynamic coupling* is a mapping that determines from ω a new sequence $\tilde{\omega}$, resulting in new sequence of random variables [2] $(\tilde{B}(1), \tilde{G}(1), \tilde{A}(1), \tilde{G}(2), \tilde{A}(2), \ldots)$, with $\tilde{B}(1) = B(1)$, and the following properties.

(a) Given that $\mathcal{H}_-(n) = h$, $\tilde{G}(n)$ is obtained from $G(n)$ by permuting its components, according to a permutation that is completely determined by h. In particular, given $\mathcal{H}_-(n) = h$, the mapping from $G(n)$ to $\tilde{G}(n)$ is one-to-one.

(b) Given that $\mathcal{H}_+(n) = h$, $\tilde{A}(n)$ is obtained from $A(n)$ by permuting its components, according to a permutation that is completely determined by h. In particular, given $\mathcal{H}_+(n) = h$, the mapping from $A(n)$ to $\tilde{A}(n)$ is one-to-one.

In particular, given the history

$$\tilde{\mathcal{H}}_-(n) = (\tilde{B}(1), \tilde{G}(1), \tilde{A}(1), \ldots, \tilde{G}(n-1), \tilde{A}(n-1))$$

and knowledge of the policy π, one can recover the history $\mathcal{H}_-(n)$ under policy π, as well as the values of $B(2), \ldots, B(n)$ and $W(1), \ldots, W(n-1)$. Using property (a), once $\tilde{G}(n)$ is observed, the value of $G(n)$ can be inferred. This implies that $W(n)$ can be inferred. Thus, given the history

$$\tilde{\mathcal{H}}_+(n) = (\tilde{B}(1), \tilde{G}(1), \tilde{A}(1), \ldots, \tilde{G}(n-1), \tilde{A}(n-1), \tilde{G}(n), \tilde{W}(n))$$

one can recover the history $\mathcal{H}_+(n)$ under policy π. Using property (b), once $\tilde{A}(n)$ is observed, the value of $A(n)$, as well as of $B(n+1)$, can be inferred. As a consequence, a new policy $\tilde{\pi}$ that operates on a system driven by the sequence $\tilde{\omega}$, can use the knowledge of the values of $B(n)$ and $W(n)$ under ω and policy π. Finally, since the conditional distribution of $G(n)$ given $\mathcal{H}_-(n)$ (respectively, the conditional distribution of $A(n)$ given $\mathcal{H}_+(n)$) is permutation-invariant,[3] we have that the sequence $\{(\tilde{G}(n), \tilde{A}(n))\}$ has the same distribution as the sequence $\{(G(n), A(n))\}$.

Suppose now that the new policy $\tilde{\pi}$ that operates on a system driven by $\{(\tilde{G}(n), \tilde{A}(n))\}$ results in a sequence $\{\tilde{B}(n)\}$ such that $\tilde{B}(n) \preceq B(n)$ for all n and all sample paths. In that case,

[2]More precisely, given the mapping $\omega \mapsto \tilde{\omega}$, any random variable X, results in a new random variable $\tilde{X}$ defined by $\tilde{X}(\omega) = X(\tilde{\omega})$.

[3]The conditional distribution of a random vector $X \in \Re^N$, given some history $\mathcal{H}$, is said to be permutation-invariant if $\mathbb{P}(X = (x_1, \ldots, x_N) \mid \mathcal{H} = h) = \mathbb{P}(X = (x_{\sigma(1)}, \ldots, x_{\sigma(N)}) \mid \mathcal{H} = h)$, for every permutation σ, every possible realization h of $\mathcal{H}$, and every x.

for any $f \in \mathcal{F}$, we have $f(\tilde{B}(n)) \leq f(B(n))$, for all n and all sample paths, Since in a dynamic coupling the distribution of $\{(\tilde{G}(n), \tilde{A}(n))\}$ is the same as that of $\{(G(n), A(n))\}$, the stochastic process $\{f(\tilde{B}(n))\}$ resulting from policy $\tilde{\pi}$ is *stochastically smaller* than the process $\{f(B(n))\}$ resulting from policy π.[4] In that case, we will say that policy $\tilde{\pi}$ *dominates* policy π, and we will write $\tilde{\pi} \preceq \pi$. Because stochastic dominance is transitive, so is policy dominance; that is, if π^1 dominates π^2, and if π^2 dominates π^3, then π^1 dominates π^3.

C. Main Result

Note that if we can establish that an LCQ policy dominates every other policy, it will follow that an LCQ policy minimizes performance criteria such as $\mathbb{E}[f(B(n))]$ or

$$\sum_{k=1}^{k^*} \alpha^k \mathbb{E}[f(B(k))]$$

for any $f \in \mathcal{F}$, any k^*, and any $\alpha \geq 0$. The same statement can be made for average cost criteria such as

$$\lim_{k^* \to \infty} \frac{1}{k^*} \sum_{k=1}^{k^*} \mathbb{E}[f(B(k))].$$

Our main result establishes that this is indeed the case.

Theorem 1: An LCQ policy dominates every other policy.

Proof: For the purposes of this proof, we will be working with a "relaxed" version of the original problem. We modify the evolution equation (1) to

$$B(n+1) = B(n) - W(n) + H(n) + A(n), \quad n = 1, 2, \ldots \quad (2)$$

where $H(n) = (H_1(n), \ldots, H_N(n))$ is an additional nonnegative control variable which (similar to $W(n)$) is chosen on the basis of the history $\mathcal{H}(n)$. In this relaxed problem, we allow a policy to add an arbitrary number of packets to any queue, right after the packet withdrawals $W(n)$ are determined. Let Φ_0 be the set of all policies for the relaxed problem. For $n \geq 1$, let Φ_n be the set of all policies under which we have $H(\tau) = 0$, for $\tau = 1, \ldots, n$. Finally, let $\Phi_\infty = \cap_{n=1}^{\infty} \Phi_n$: this is the set of policies that have $H(n) = 0$ for all n (never add extra packets), and coincides with the set of policies for the original problem. If we show that LCQ dominates every policy on Φ_0, it will follow that it dominates every policy in Φ_∞, which is the desired result.

Note that within the relaxed problem, we can assume without loss of generality, that for every n and for every queue i that is served at time n, we have $W_i(n) = G_i(n) \min\{C, B_i(n)\}$. Indeed, if the policy were to serve fewer than $G_i(n) \min\{C, B_i(n)\}$ packets, we could increase $W_i(n)$ to $G_i(n) \min\{C, B_i(n)\}$, and accordingly increase $H_i(n)$, resulting in the same value for the next queue size $B_i(n+1)$. We will henceforth restrict all policies to have this property.

We say that a policy has the LCQ property at time n if at that time, it can only serve the largest K connected queues (resolving ties arbitrarily). Let $\mathcal{L}_0$ be the set of all policies for the relaxed problem (this is the same as the set Φ_0). For $n \geq 1$, let $\mathcal{L}_n$ be the set of policies that have the LCQ property at times $\tau = 1, \ldots, n$. Finally, let $\mathcal{L}_\infty = \cap_{n=1}^{\infty} \mathcal{L}_n$: this is the set of policies that have the LCQ property at all times.

Consider now the set $S(n)$ of queues served by a policy π at some time n. We say that π deviates from LCQ at time n by at most ℓ, if we can make π have the LCQ property at time n by changing at most ℓ of the elements of $S(n)$. Let $\mathcal{L}_n^\ell$ be the set of policies that belong to $\mathcal{L}_{n-1}$ (i.e., have the LCQ property before time n), and which deviate from LCQ at time n by at most ℓ.

With these definitions and notation at hand, we can now provide an outline of the proof. The idea is to start with a general policy and modify it progressively, to obtain a sequence of policies, each dominating the previous ones, and which have the LCQ property for larger amounts of time. For any given time, the argument involves a number of steps, with each step effecting a reduction of the amount by which the LCQ property is violated. The overall proof involves a sequence of steps that relies on the following two lemmas.

Lemma 1: Given a policy $\pi \in \mathcal{L}_\tau^\ell \cap \Phi_\infty$ (i.e., has the LCQ property before time τ, deviates from LCQ by at most ℓ at time τ, and never adds extra packets), with $\ell > 0$, we can construct a dominating policy $\tilde{\pi} \in \mathcal{L}_\tau^{\ell-1}$.

Lemma 2: Given a policy $\tilde{\pi} \in \mathcal{L}_\tau^\ell$, we can construct a dominating policy $\hat{\pi} \in \mathcal{L}_\tau^\ell \cap \Phi_\infty$ (i.e., which does not add extra packets).

The proofs of the preceding lemmas can be found in the Appendix.

The proof of the theorem is completed by applying Lemmas 1–2 repeatedly. Starting from an arbitrary policy $\pi \in \Phi_\infty$, we obtain a sequence of policies each of which dominates the previous one. We obtain policies that belong to $\mathcal{L}_1^K \cap \Phi_\infty, \mathcal{L}_1^{K-1} \cap \Phi_\infty, \ldots, \mathcal{L}_1^0 \cap \Phi_\infty$. The last such policy belongs to $\mathcal{L}_2^K \cap \Phi_\infty$. Continuing similarly, we obtain policies π^τ that belong to $\mathcal{L}_\tau^0 \cap \Phi_\infty$ for increasing values of τ. Furthermore, our construction is such that each such policy π^τ agrees with the preceding policy $\pi^{\tau-1}$ until time $\tau - 1$. Therefore, this sequence of policies also defines a limiting policy π^*, which for every τ, agrees with π^τ until time τ. Therefore, π^* has the LCQ property at all times, belongs to Φ_∞, and dominates all of the policies in the sequence, including the original policy π. This concludes the proof of the theorem. **q.e.d.**

Given some $f \in \mathcal{F}$, let $V_n(b)$ be the optimal (over all policies) value of $\mathbb{E}[f(B(n)) \mid B(1) = b]$, and let us compare two initial conditions such that $\tilde{b} \sqsubseteq b$. The argument in Step B of the proof of Lemma 1 shows that no matter what the optimal policy does starting from b, there is another policy which starting from $\tilde{b}$ maintains the relation $\tilde{b}(n) \sqsubseteq b(n)$ for all n, and results in smaller or equal expected cost. This shows the following.

Corollary 1: For any $f \in \mathcal{F}$, and any n, if $\tilde{b} \sqsubseteq b$ then $V_n(\tilde{b}) \leq V_n(b)$. In particular, $V_n \in \mathcal{F}$.

Indeed, one can prove using dynamic programming and induction on n, that if $f \in \mathcal{F}$, then $V_n \in \mathcal{F}$ for all n, and an LCQ policy minimizes $\mathbb{E}[f(B(n)) \mid B(1) = b]$, for all b. However,

[4] A process $\{\tilde{X}(n)\}$ is stochastically smaller than another process $\{X(n)\}$ (symbolically, $\tilde{X} \overset{st}{\leq} X$), if there exists another process $\{\hat{X}(n)\}$ defined on the same probability space as $\{X(n)\}$, that has the same probability distribution as $\{\tilde{X}(n)\}$, and satisfies $\hat{X}_i(n) \leq X_i(n)$, almost surely, for every component i, and every time n.

this is a weaker result, compared to the stochastic dominance established by Theorem 1, and the proof is not much easier.

Let us now consider a different relation $\sqsubseteq_1$, which is defined the same as $\sqsubseteq$, except that we take $C = 1$ in that definition. Let $\mathcal{F}_1$ be the class of monotonic functions f such that $\tilde{b} \sqsubseteq_1 b$ implies that $f(\tilde{b}) \leq f(b)$. This class, which is closely related to the class of increasing Schur-convex functions, clearly satisfies $\mathcal{F}_1 \subseteq \mathcal{F}$. Is it true that when $f \in \mathcal{F}_1$, the optimal value function $V_n(b) = \min_\pi \mathbb{E}[f(B(n)) \mid B(1) = b]$ belongs to $\mathcal{F}_1$? If that were true, it would imply that a configuration $\tilde{b}$ obtained by performing a balancing 1-interchange on another configuration b would be preferable. However, this is not the case, as shown by the example that follows. This explains why we had to work with the relation $\sqsubseteq$ instead of the simpler relation $\sqsubseteq_1$. Furthermore, since the value function in that example does not belong to $\mathcal{F}_1$ but belongs to $\mathcal{F}$ (by the above corollary), this shows that $\mathcal{F}$ is strictly larger than the class obtained when $C = 1$, as claimed at the end of Section II-A.

Example 1: Consider a system in which there are no arrivals, $C = 2$, only two queues ($N = 2$), and only one transmitter ($K = 1$). Let the cost function, defined by $f(b) = b_1 + b_2$, correspond to expected queue length. Let $p > 0$ be the probability that a queue is connected at any given time. If the initial state is $b = (2, 0)$, the expected value of the cost $f(B(2))$ at time 2 is $2(1 - p)$, because there is probability $1 - p$ that queue 1 is not connected at time 1. Consider now the more balanced state $\tilde{b} = (1, 1)$ and note that $\tilde{b} \sqsubseteq_1 b$. (On the other hand, it is not true that $\tilde{b} \sqsubseteq b$.) The expected cost is equal to $1 + (1 - p)^2$. (There will always be one packet at time 2, and there will be a second packet if both queues were disconnected at time 1.) Since $2(1 - p) < 1 + (1 - p)^2$, we see that the "less balanced" configuration $(2, 0)$ is preferable.

III. Extensions

In this section, we discuss various extensions of Theorem 1. We are mainly interested in relaxing the i.i.d. assumptions we have made on the channel states and on the random arrivals. All of these extensions are possible with essentially the same proof as for Theorem 1. For this reason, we only provide brief justifications.

In our discussion of the dynamic coupling method, the key property that was used was that the sequence $\{(\tilde{G}(n), \tilde{A}(n))\}$ had the same distribution as the sequence $\{(G(n), A(n))\}$. This was a consequence of our i.i.d. Bernoulli assumptions, but in fact the only property required is permutation-invariance. This shows that the result in Theorem 1 remains valid under the following, more general, assumption. Some examples are discussed below.

Assumption 2:

(a) The conditional distribution of $G(n)$ given $\mathcal{H}_-(n)$ is permutation-invariant. (Recall that $\mathcal{H}_-(n) = (B(1), G(1), \ldots, G(n-1), A(1), \ldots, A(n-1))$.)

(b) The conditional distribution of $A(n)$ given $\mathcal{H}_+(n)$ is permutation-invariant. (Recall that $\mathcal{H}_+(n) = (B(1), G(1), \ldots, G(n), A(1), \ldots, A(n-1), G(n), W(n))$.

Example 2: **Dependence of channel states.** For an example where the permutation invariance assumption holds, suppose that the number, call it L, of connected queues at time n is random, possibly dependent on the past history, and with an arbitrary conditional distribution. (With our original model, L had a binomial distribution.) Given that $L = l$, we assume that the set of connected queues is equally likely to be any l-element subset of $\{1, \ldots, N\}$. In particular, $\mathbb{P}(G_i(n) = 1)$ need not be the same for all n (although it must be the same for all i).

Example 3: **Dependence in the arrival processes.** A similar example arises if the total number of arrivals at time n, call it M, has an arbitrary conditional distribution, and then the arrivals are assigned to queues in a permutation-invariant fashion. Thus, we can allow complicated dependencies between the numbers of arrivals at different times, possibly reflecting the dynamics in the rest of the network, or a fairly general flow control mechanism (as long as the flow control mechanism does not distinguish between different packet classes).

Example 4: **Non-Bernoulli arrivals.** A special case of the above arises when Assumption 1(a) on the channel state $G(n)$ holds, except that $\mathbb{P}(G_i(n) = 1)$ changes with time, and similarly for $A(n)$. In particular, we may assume $G_i(n) = 0$ for all i, and for all n in certain predetermined intervals. Arrivals accumulate during such an interval, but since no queue can be served, this interval is equivalent to a single time slot, but with the number of arrivals having a more general distribution. This establishes that Theorem 1 remains valid if the random variables $A_i(n)$ are i.i.d. and each one of them can be expressed as a sum of independent Bernoulli random variables. The special case where all of the Bernoulli random variables have the same parameters allows the $A_i(n)$ to be independent binomial random variables. Because a Poisson distribution can be approximated by a sequence of binomial distributions, we can also use a limiting argument to establish that Theorem 1 remains valid if the random variables $A_i(n)$ are i.i.d. Poisson. The details of this argument are not particularly interesting, and will not be given here. Nevertheless, we will revisit the Poisson case in Section III-A5.

Example 5: **Unknown channel states.** Another interesting variation arises if the channel states $G_i(n)$ are not known at the time that the control vector $W(n)$ is chosen. That is, the scheduler selects up to K queues, and will only be able to serve the subset of the selected queues that happen to be connected. It can be checked that the proof remains valid, with minimal modifications. In fact, a more complex channel model is possible. At each time n, and for each channel i, let there be two independent channel variables $G_i(n)$ and $\bar{G}_i(n)$. The ith queue will be connected if and only if $G_i(n) = \bar{G}_i(n) = 1$. However, the scheduler only gets to observe $G_i(n)$ before selecting the queues to be served. Once more, Theorem 1 remains valid. With this model, there can be an attempt to transmit a packet from queue i, each time that $G_i(n) = 1$. The number of attempts until a successful transmission occurs is the same as the number of independent Bernoulli trials until a "success" occurs, and has a geometric distribution.

By reinterpreting this model, we see that Theorem 1 remains valid if the random variables $\bar{G}_i(n)$ are absent, but packets have i.i.d. geometric transmission times, of duration unknown to the transmitter (i.e., after a transmission attempt, the packet departs the system with some fixed probability), in a model that allows preemptive service (i.e., when a "transmission attempt" fails, we can switch to serving a different packet). This can be used to model, for example, an error-prone system where even when the channel is in the "on" state, packets may incur transmission errors and require retransmission. A similar result has also been established for the case of $K = 1$ and $C = 1$ in [11].

IV. Comparison of LCQ and a Naive Policy

In this section, we consider the legitimate question whether an optimal (i.e., LCQ) policy results in a substantial performance improvement, compared to a naive policy. Given that closed-form expressions are not possible, we approach this question by considering the asymptotic case where the number N of queues becomes large, while keeping the number K of transmitters and the total arrival rate $\lambda = N\mathbb{E}[A_1(n)]$ constant. We consider the model of Section II, with $C = 1$ and the same i.i.d. assumptions (as in Assumption 1), except that we allow the $A_i(n)$ to take values larger than 1. We will first consider a naive, and clearly nonoptimal, randomized policy and show that it has the largest possible stability region. However, we show that the expected sum of the queue lengths, in steady state, increases linearly with N. In contrast, we show that with an optimal policy, the expected sum of the queue lengths does not increase with N, at least when the $A_i(n)$ are Poisson.

Let us first discuss the stability region. The largest possible service rate occurs when all queues have available packets, in which case, the expected number of packets served per time slot is equal to

$$\mu(N,K) = \mathbb{E}[\min\{K, L(n)\}]$$

where $L(n) = \sum_{i=1}^{N} G_i(n)$ is the number of connected queues. As one would intuitively expect, and as established in [1], there exists a policy under which the system is stable (positive recurrent) if and only if $\lambda < \mu(N,K)$. Note that as $N \to \infty$, we have $\mu(N,K) \to K$.

We now describe a naive randomized policy, denoted by π_R. This policy operates as follows. If the number $L(n)$ of connected queues is no larger than K, all connected queues are served. Otherwise, π_R serves a K-element subset of the connected queues, with every subset being equally likely to be selected, regardless of the queue lengths.

In essence, π_R, unlike LCQ policies, disregards queue length information. In fact, π_R may even select a connected queue that has no packets. A somewhat more reasonable policy might be one that only selects at random between connected queues with a nonzero number of packets. However, this should make only a small difference at the boundary of the stability region (when λ approaches $\mu(N,K)$), because one would expect almost all of the queues to have a nonzero number of packets. Furthermore, this modified policy is difficult to analyze.

Theorem 2: Under policy π_R, and if $\lambda < \mu(N,K)$, the system is stable. Furthermore, if $\mathbb{E}\left[A_i^3(n)\right] < \infty$, we have

$$\mathbb{E}\left[\sum_{i=1}^{n} B_i(n)\right] = N\frac{\lambda + N\mathbb{E}\left[A_i^2(n)\right] - 2\lambda^2/N}{2(\mu(N,K)-\lambda)}$$

where the expectation is taken with respect to the steady-state distribution.

Proof: At each time slot, the expected number of selected queues is $\mu(N,K)$. Thus, at each time, queue i has probability $\mu(N,K)/N$ to be selected for service. Therefore, for queue i viewed in isolation, the expected number of arrivals per unit time is λ/N, and the service rate is $\mu(N,K)/N$. As long as $\lambda < \mu(N,K)$, it is stable. By a standard argument based on the Lyapunov function $B_i^3(n)$, and using the assumption $\mathbb{E}\left[A_i^3(n)\right] < \infty$, one also obtains that $\limsup_{n\to\infty} \mathbb{E}\left[B_i^2(n)\right] < \infty$.

Let us now consider queue i in steady state, and the evolution equation

$$B_i(n+1) = B_i(n) - W_i(n) + A_i(n).$$

Taking expectations of both sides, we obtain $\mathbb{E}\left[W_i^2(n)\right] = \mathbb{E}[W_i(n)] = \lambda/N$, where the first equality follows because $C = 1$, which implies that $W_i(n) \in \{0,1\}$. Also, if we square the above equation, and then take expectations, we obtain

$$\begin{aligned}\mathbb{E}\left[B_i^2(n+1)\right] &= \mathbb{E}\left[B_i^2(n)\right] + \mathbb{E}\left[W_i^2(n)\right] + \mathbb{E}\left[A_i^2(n)\right]\\ &\quad -2\mathbb{E}\left[B_i(n)W_i(n)\right] - 2\mathbb{E}[A_i(n)]\mathbb{E}[(W_i(n) - B_i(n)]\end{aligned}$$

where we have used the independence of $A_i(n)$ from $B_i(n)$ and $W_i(n)$. In steady state, we have $\mathbb{E}\left[B_i^2(n+1)\right] = \mathbb{E}\left[B_i^2(n)\right]$. Furthermore, whenever $B_i(n) \neq 0$, the conditional expectation of $W_i(n)$ is equal to the probability that queue i is selected for service, which is $\mu(N,K)/N$. Thus,

$$\begin{aligned}0 = \frac{\lambda}{N} + \mathbb{E}[A_i^2(n)] - 2\mathbb{E}[B_i(n)]\frac{\mu(N,K)}{N}\\ -2\frac{\lambda}{N}\left(\frac{\lambda}{N} - \mathbb{E}[B_i(n)]\right).\end{aligned}$$

We solve for $\mathbb{E}[B_i(n)]$ and recover the desired result. **q.e.d.**

According to Theorem 2, the naive policy has the largest possible stability region. However, the total expected number of packets in queue increases linearly with N. In the special case of Bernoulli arrivals, we have $\mathbb{E}[A_i^2(n)] = \lambda/N$, and as $N \to \infty$, the total expected number of packets in queue behaves like

$$N\frac{\lambda}{K-\lambda}.$$

The same asymptotic applies if the random variables $A_i(n)$ are Poisson.

Let $Q_N^*(n) = \sum_{i=1}^{N} B_i(n)$ stand for the total number of packets in the system at time n, when the number of queues is N, arrivals are i.i.d. Poisson, and an optimal policy is used starting from zero initial queue lengths. Let $Q_{2N}^*(n)$ be defined similarly, and with the same initial conditions, except that the number of queues is doubled (while K and λ are held constant). Our next result, to be contrasted with the preceding discussion of π_R, indicates that doubling N does not increase the expected total queue size.

Theorem 3: Suppose that the random variables $A_i(n)$ are i.i.d. and Poisson. Let the initial queues be empty. Then, $\mathbb{E}\left[Q_{2N}^*(n)\right] \leq \mathbb{E}\left[Q_N^*(n)\right]$, for all N.

Proof: We consider an optimal policy π that operates on an "original" system with N queues. We will use a coupling argument to construct a policy $\tilde{\pi}$ that operates on a "new" system with $2N$ queues under which $\tilde{Q}_{2N}(n) \leq Q^*_N(n)$, for all n. This will then imply that $\mathbb{E}[Q^*_{2N}(n)] \leq \mathbb{E}[\tilde{Q}_{2N}(n)] \leq \mathbb{E}[Q^*_N(n)]$, for all n. Throughout, we use a tilde to indicate quantities associated with the new system, and no tilde for the original system.

At each time n, we map arrivals to queue i in the original system to queues $2i-1$ and $2i$ in the new system, as follows. Each arrival to i in the original system is independently assigned to queue $2i-1$ or $2i$ in the new system, with equal probability. This ensures that arrivals to individual queues in the new system are i.i.d. Poisson, with rate $\lambda/(2N)$.

Also, at each time n, we map the channel state of queue i in the original system to the longest queue among queues $2i-1$ and $2i$ in the new system. We generate the channel state of the shortest queue among queues $2i-1$ and $2i$ by drawing an independent Bernoulli random variable with the same parameter $\mathbb{P}(G_i(n)=1)$.

The policy $\tilde{\pi}$ is defined in terms of the actions of policy π, as follows. If π serves queue i, then $\tilde{\pi}$ serves the longest of queues $2i-1$ and $2i$.

We will now use induction on n, to show that

$$B_i(n) = \tilde{B}_{2i-1}(n) + \tilde{B}_{2i}(n).$$

This is trivially true for $n=1$. Suppose it is also true at some time n. If $B_i(n) > 0$, and queue i is served by π (in particular, queue i is connected), then the longest among queues $2i-1$ and $2i$ is also nonempty and connected. Thus, whenever there is a withdrawal from queue i in the original system, there is a withdrawal from queue $2i-1$ or $2i$ in the new system. Furthermore, because of the way arrivals have been coupled, the number of arrivals to queue i in the original system equals the total number of arrivals in queues $2i-1$ and $2i$ in the new system. Therefore,

$$B_i(n+1) = \tilde{B}_{2i-1}(n+1) + \tilde{B}_{2i}(n+1)$$

which completes the induction. It then follows that $\tilde{Q}_{2N}(n) = Q^*_N(n)$ for all n, which completes the proof. **q.e.d.**

We conjecture that, for large N, Theorem 3 remains valid in many cases where $A_i(n)$ does not have a Poisson distribution, e.g., if each $A_i(n)$ is Bernoulli with parameter λ/N, or more generally, if the distance (suitably defined) between the distribution of $A_i(n)$ and a Poisson distribution with the same mean decays faster than $1/N$ as $N \to \infty$. Another interesting question is whether $\mathbb{E}[Q^*_M(n)] \leq \mathbb{E}[Q^*_N(n)]$ for every $M > N$ (Theorem 3 only deals with the case $M = 2N$).

V. A Fluid Model With a Total Service Constraint

In this section, we consider a different type of constraint on the transmitters. Instead of requiring that there can be up to C packets served from each queue selected for service, we introduce an aggregate constraint, namely, that the total number of packets served is bounded by C). This is a reasonable model if there is a power constraint that applies to the entire system. If we further assume that $C=1$, we are dealing with the model in [11], and an LCQ policy is optimal. If we assume that $K=1$, we are dealing with a special case of the model of Section II and an LCQ policy is again optimal. Other than these cases, the structure of optimal policies is unknown. We will henceforth focus on the special case where $K=N$ (unlimited number of transmitters), and a variant of the model, which allows serving a noninteger number of packets from each queue. We will establish that a generalization of LCQ, which we call a "Most Balanced" (MB) policy is optimal.

We use the same notation as in Section II, and make the following assumption.

Assumption 3:

(a) The distributions of $G(n)$ and of $A(n)$ are permutation-invariant.

(b) The random vectors $G(1), A(1), G(2), A(2), \ldots$ are independent.

Let $\Re_+$ be the set of nonnegative reals. Suppose that at time n we have $(B(n), G(n)) = (b, g)$, for some vectors b and g. We assume that the set $W(b,g)$ of feasible vectors of packet withdrawals, when the system is in state (b,g), is of the form

$$W(b,g) = \left\{ w \in \Re^N_+ | w_i \leq b_i \text{ for all } i, \text{ and } \sum_{i=1}^{N} w_i \leq C \right\}.$$

A **Most Balanced** policy always chooses a $w \in W(b,g)$ that minimizes

$$\max_{i:g_i=1} (b_i - w_i).$$

For example, if $b = [5,4,3,2,6,1]$, $g = [1,1,1,1,0,0]$, and $C=2$, a most balanced policy will let $w = [1.5, 0.5, 0, 0, 0, 0]$, resulting in the configuration $b - w = [3.5, 3.5, 3, 2, 6, 1]$. It is not hard to show that the most balanced policy is uniquely defined. Finally, for the purposes of this section, we let $\mathcal{F}$ be the set of all functions $f : \Re^N_+ \mapsto \Re$ that are convex, nondecreasing, and permutation invariant.

Theorem 4: Let $B(n)$ be the vector of queue sizes at time n. For any function $f \in \mathcal{F}$ and for every $n \geq 0$, the MB policy minimizes $\mathbb{E}[f(B(n))]$.

Proof: The proof uses a dynamic programming argument. Let $V_n(b)$ be the minimum of $\mathbb{E}[f(B(n))]$, over all policies, starting from the initial condition $B(1) = b$. We then have $V_0(b) = f(b)$, and the following recursion, for $n \geq 1$:

$$\tilde{V}_n(\tilde{b}) = \sum_a \mathbb{P}(A(n) = a) V_{n-1}(\tilde{b} + a)$$
$$v_n(b,g) = \min_{w \in W(b,g)} \tilde{V}_n(b - w)$$
$$V_n(b) = \sum_g \mathbb{P}(G(n) = g) v_n(b,g).$$

Lemma 3: The functions V_n and $\tilde{V}_n$ belong to $\mathcal{F}$ (convex, nondecreasing, and permutation invariant), for all n. The function $b \mapsto v_n(b,g)$ is nondecreasing and convex for every g. Finally,

$$v_n(b_{\sigma(1)}, \ldots, b_{\sigma(N)}, g_{\sigma(1)}, \ldots g_{\sigma(N)}) = v_n(b,g)$$

for every (b,g) and every permutation σ.

Proof: Monotonicity is immediate from the structure of the above recursion, and the monotonicity of the function $V_0 = f \in \mathcal{F}$ that starts the recursion. Permutation invariance is also an easy consequence of this recursion, the permutation invariance of the function $V_0 = f \in \mathcal{F}$, and the permutation invariance of the mappings $a \mapsto \mathbb{P}(A(n) = a)$ and $g \mapsto \mathbb{P}(G(n) = g)$.

It remains to establish convexity, which is done by induction. The function $V_0 = f$ is convex, by definition. Assuming that V_{n-1} is convex, $V_{n-1}(\tilde{b} + a)$ is a convex function of $\tilde{b}$, for any a, from which it follows that $\tilde{V}_n$ is convex. Let us now fix some g, and using the notation $l = b - w$, note that $v_n(b, g)$ is the minimum of $\tilde{V}_n(l)$ subject to the constraints

$$\sum_{i=1}^{N} l_i \geq \sum_{i=1}^{N} b_i - C; \quad l_i \geq 0; \qquad \text{if } g_i = 0 \text{ then } l_i = b_i.$$

This is a convex optimization problem and by a standard argument, the optimal value $v_n(b, g)$ is a convex function of the parameter vector b appearing in the right-hand side of the linear constraints. Thus, $v_n(b, g)$ is a convex function of b. Finally, V_n is a weighted average of convex functions and is therefore convex. **q.e.d.**

Fix some (b, g), and let $c = b_1 + \cdots + b_N - C, l_i = b_i - w_i$. Consider the vector l^* obtained by the MB policy, which is an optimal solution to the optimization problem

$$\begin{aligned} \text{minimize} \quad & \max_{i:g_i=1} l_i \\ \text{subject to} \quad & 0 \leq l_i \leq b_i \\ & l_1 + \cdots + l_N \geq c \\ & l_i = 0 \text{ if } g_i = 0. \end{aligned}$$

We will now show that l^* also minimizes $\tilde{V}_n(l)$, subject to the same constraints, which will then imply that $w = b - l^*$ is an optimal decision when $(B(n), G(n)) = (b, g)$, and therefore the MB policy is optimal. For simplicity, and without loss of generality, let us assume that $g_i = 1$ for all i. If $\sum_{i=1}^{N} b_i \leq C$, then $c \leq 0$ and $l^* = 0$. Since $\tilde{V}_n(l)$ is a nondecreasing function of l, it follows that l^* minimizes $\tilde{V}_n(l)$ as well. We can therefore assume that $c > 0$, in which case we can replace the constraint $l_1 + \cdots + l_N \geq c$ by the equality constraint $l_1 + \cdots + l_N = c$.

Consider the compact and convex set S of optimal solutions to the problem of minimizing $\tilde{V}_n(l)$ subject to the constraints $0 \leq l_i \leq b_i$ and $l_1 + \cdots + l_N = c$. We will show that the MB vector in S must be equal to l^*, implying that l^* indeed minimizes $\tilde{V}_n(l)$.

Let $\tilde{l}$ be an optimal solution to the problem of minimizing $\max_i l_i$ within the set S. If $\max_i \tilde{l}_i = \max_i l_i^*$, then $\tilde{l} = l^*$ (this is because the optimization problem defining l^* has a unique solution). Let us therefore assume that $\max_i \tilde{l}_i \neq \max_i l_i^*$. In particular, there exists some i such that $\tilde{l}_i > l_k^*$ for all k. Furthermore, since $\tilde{l}_1 + \cdots + \tilde{l}_N = c = l_1^* + \cdots + l_N^*$, there also exists some j such that $\tilde{l}_j < l_j^*$, which also implies that $\tilde{l}_j < b_j$. Consider a new vector $\hat{l}$ obtained by making components i and j of $\tilde{l}$ more balanced. More precisely, let ϵ be a small positive number, and let

$$\hat{l}_i = \tilde{l}_i - \epsilon, \quad \hat{l}_j = \tilde{l}_j + \epsilon, \quad \hat{l}_k = \tilde{l}_k, \qquad \text{for } k \neq i, j.$$

Note that $\hat{l}$ satisfies the required constraints $l_1 + \cdots + l_N = c$ and $0 \leq l_i \leq b_i$ when ϵ is chosen small enough, e.g., $\epsilon = \min\{\tilde{l}_i, b_j - \tilde{l}_j\}$. Furthermore, we restrict ϵ to satisfy $\tilde{l}_i - \epsilon > \tilde{l}_j + \epsilon$.

Lemma 4: We have $\tilde{V}_n(\hat{l}) \leq \tilde{V}_n(\tilde{l})$, and therefore, $\hat{l} \in S$.

Proof: Since $\tilde{l}$ and $\hat{l}$ differ only in the ith and jth components, it suffices to consider a function $v \in \mathcal{F}$ of two variables, denoted by d_1 and d_2, and show that if $d_1 > d_2$, $\epsilon > 0$, and $d_1 - \epsilon > d_2 + \epsilon$, then $v(d_1 - \epsilon, d_2 + \epsilon) \leq v(d_1, d_2)$. Note that $(d_1 - \epsilon, d_2 + \epsilon)$ lies on the interval joining (d_1, d_2) and (d_2, d_1). In particular, for some $\gamma \in [0, 1]$, we have $(d_1 - \epsilon, d_2 + \epsilon) = \gamma(d_1, d_2) + (1 - \gamma)(d_2, d_1)$. Using the convexity of v, we obtain

$$\begin{aligned} v(d_1 - \epsilon, d_2 + \epsilon) &\leq \gamma v(d_1, d_2) + (1 - \gamma) v(d_2, d_1) \\ &= v(d_1, d_2) \end{aligned}$$

where the last equality made use of the permutation-invariance of v. **q.e.d.**

We have thus constructed a new element of S with one less component equal to $\max_k \tilde{l}_k$. By repeating this procedure a number of times, we obtain a new vector $\bar{l}$ which is in S, but with $\max_k \bar{l}_k < \max_k \tilde{l}_k$. But this contradicts the definition of $\tilde{l}$, establishes that $\tilde{l} = l^*$, and completes the proof. **q.e.d.**

Using Theorem 4, it is easily seen that the most balanced policy is optimal for a wide variety of performance criteria, such as a discounted sum of the $\mathbb{E}[f(B(n))]$ over a finite or infinite horizon, or an undiscounted sum over a finite horizon. Furthermore, the theorem covers the important special case of $f(b) = b_1 + \cdots + b_N$ (total queue length).

VI. Concluding Remarks

We have studied symmetric on–off queueing systems, which form a special case of queueing systems with time-varying service rates, with the objective of finding policies that minimize buffer occupancies (or equivalently, delays). In general, very few results exist on minimum delay scheduling over time-varying channels. Under the assumption of i.i.d. Bernoulli arrivals and connectivity variables, we showed that the "Longest Connected Queue" policy (LCQ) is optimal for the case of K servers each able to serve C packets per slot from a connected queue. Using a fluid service model, we also showed that the "Most Balanced" (MB) policy is optimal for the case of N servers and a total capacity, between all servers, of C units. We established these results using stochastic coupling techniques and dynamic programming, respectively.

We have argued that our results also hold for certain non-i.i.d. models, as long as the arrival and connectivity distributions remain permutation-invariant. However, the use of stochastic coupling techniques relies heavily on symmetry between the queues and cannot be applied in the absence of such symmetry.

Even in the symmetric case, many problems remain open. For example, consider the simple extension where we have K servers, with $1 < K < N$, that can serve a total of C packets from K of the connected queues (notice that here we do not impose the limit on the number of packets that can be served from each queue). In this case, it is not difficult to show that a "Most Balanced" policy is not optimal, even if all the queues are always connected. In fact, we suspect that the problem of minimizing the total queue length is NP-hard, even for the deterministic and static special case where all channels are always on and there are no arrivals.

We finally note that a closed-form description of the optimal policy under general conditions on the arrival and channel state processes is not feasible. However, it would be interesting to examine simple suboptimal policies and bound their performance for more general models of time-varying queueing systems.

Appendix
Proofs of Lemmas 1 and 2

Proof of Lemma 1: This lemma, as well as the next one, are proved using the dynamic coupling method described in Section II-B.

Let us fix a policy $\pi \in \mathcal{L}_\tau^\ell \cap \Phi_\infty$, and a sample path ω consisting of the values $\omega = (b(1), g(1), a(1), \ldots)$ of the random variables $(B(1), G(1), A(1), \ldots)$ (the "original" system). We construct a new sample path $\tilde{\omega}$ and policy $\tilde{\pi}$ (the "new" system), with the same initial condition $\tilde{b}(1) = b(1)$. We use a tilde to denote quantities associated with the new system. Before time τ, we let arrivals and channel states be the same in the two systems ($\tilde{a}(n) = a(n)$ and $\tilde{g}(n) = g(n)$, for $n < \tau$), and let $\tilde{\pi}$ be identical to π. As a consequence, $\tilde{b}(\tau) = b(\tau)$. At time τ, we let the channel states of the two systems be the same ($\tilde{g}(\tau) = g(\tau)$).

Step A: Policy $\tilde{\pi}$ at time τ.

If π (at time τ and for that particular sample path) deviates from LCQ by less than ℓ, we let $\tilde{\pi}$ choose the same controls as π, and set $\tilde{a}(\tau) = a(\tau), \tilde{h}(\tau) = 0$, resulting in $\tilde{b}(\tau+1) = b(\tau+1)$.

Suppose now that π (at time τ and for that particular sample path) deviates from LCQ by ℓ. Let i be such that $b_i = \max\{b_k \mid g_k = 1, w_k = 0\}$, that is, i corresponds to a longest connected queue that is not served by π. Similarly, let j be such that $b_j = \min\{b_k \mid g_k = 1, w_k > 0\}$, that is, j corresponds to a shortest connected queue that is served by π. It can be seen that $b_i > b_j$ (otherwise, the connected queues that are served are at least as long as the connected queues that are not served, and π would have the LCQ property). We now let $\tilde{\pi}$ serve the same queues as π, except that queue j is replaced by queue i. It can be seen that such a $\tilde{\pi}$ will deviate from LCQ by at most $\ell - 1$.

We now proceed to describe the action of $\tilde{\pi}$ in detail. Without loss of generality, we assume that $i = 1$ and $j = 2$. Thus, π serves queue 2, and $b(\tau)$ is of the form

$$b(\tau) = \begin{bmatrix} M \\ m \\ s \end{bmatrix}$$

for some $s \in \mathcal{Z}_+^{N-2}$ and $M > m$. We distinguish three cases.

(i) Suppose that $C \le m < M$. Policy π removes C packets from queue 2. We let policy $\tilde{\pi}$ remove C packets from queue 1. We also let $\tilde{h}(\tau) = 0$ and $\tilde{a}(\tau) = a(\tau)$. The resulting configurations are of the form

$$b(\tau+1) = \begin{bmatrix} M \\ m - C \\ s \end{bmatrix} + a(\tau)$$

$$\tilde{b}(\tau+1) = \begin{bmatrix} M - C \\ m \\ s \end{bmatrix} + a(\tau).$$

(ii) Suppose that $m < M \le C$. In this case, π drives m down to zero. We let $\tilde{\pi}$ serve queue 1, also driving it down to zero. Furthermore, $\tilde{\pi}$ adds packets to queue 2 to drive it up to M, i.e., $\tilde{h}_2(\tau) = M - m$. We also let

$$\tilde{a}_i(\tau) = \begin{cases} a_2(\tau), & \text{if } i = 1, \\ a_1(\tau), & \text{if } i = 2, \\ a_i(\tau), & \text{otherwise.} \end{cases}$$

The resulting configurations are of the form

$$b(\tau+1) = \begin{bmatrix} M \\ 0 \\ s \end{bmatrix} + a(\tau), \quad \tilde{b}(\tau+1) = \begin{bmatrix} 0 \\ M \\ s \end{bmatrix} + \tilde{a}(\tau).$$

Thus, $\tilde{b}(\tau+1)$ and $b(\tau+1)$ are permutations of each other.

(iii) Suppose finally that $m < C < M$. In this case, π drives m down to zero. We let $\tilde{\pi}$ serve queue 1 and remove C packets. We also let $\tilde{\pi}$ add $C - m$ packets to queue 2, i.e., $\tilde{h}_2(\tau) = C - m$, driving it up to C, and $\tilde{a}(\tau) = a(\tau)$. The resulting configurations are of the form

$$b(\tau+1) = \begin{bmatrix} M \\ 0 \\ s \end{bmatrix} + a(\tau)$$

$$\tilde{b}(\tau+1) = \begin{bmatrix} M - C \\ C \\ s \end{bmatrix} + a(\tau).$$

This completes the description of policy $\tilde{\pi}$ at time τ. Note that our construction of $\tilde{\pi}$ at time τ guarantees that $\tilde{b}(\tau+1) \sqsubseteq b(\tau+1)$.

Step B: Policy $\tilde{\pi}$ at times $n > \tau$.

We now construct the policy $\tilde{\pi}$ for times $n > \tau$. We proceed recursively. Suppose that $\tilde{\pi}$ has been defined up to some time $n - 1$ and that $\tilde{b}(n) \sqsubseteq b(n)$. (For $n = \tau + 1$, this has already been accomplished, in Step A, which starts the recursion.) We consider three cases, which correspond to the three cases in the definition of the relation $\sqsubseteq$.

Case (i) If $\tilde{b}(n) = b(n)$, we let the channel states, arrivals, and controls be the same for both systems, which ensures that $\tilde{b}(n+1) = b(n+1)$ and $\tilde{b}(n+1) \sqsubseteq b(n+1)$.

Case (ii) Suppose that $\tilde{b}(n)$ is obtained from $b(n)$ by permuting components i and j. Without loss of generality, we assume that $i = 1, j = 2$. For queues $l \notin \{1, 2\}$ we let the channel states, arrivals, and controls be the same for both systems. For queues 1 and 2, we let channel states, arrivals, and controls for queue 1 in the new system be the same as for queue 2 in the original system, and *vice versa*. Then, the last $N - 2$ components of $\tilde{b}(n+1)$ and $b(n+1)$ are equal, whereas the first two remain permutations of each other. In particular, $\tilde{b}(n+1) \sqsubseteq b(n+1)$.

Case (iii) We finally consider the remaining case (iii) in the definition of $\sqsubseteq$. Without loss of generality, we assume that $i = 1$ and $j = 2$. In particular, for some m and M, for some positive integer k with with $m \leq M - kC$, and for some $s \in \mathcal{Z}_+^{N-2}$, we have

$$b(n) = \begin{bmatrix} M \\ m \\ s \end{bmatrix}, \quad \tilde{b}(n) = \begin{bmatrix} M - kC \\ m + kC \\ s \end{bmatrix}.$$

Note that at the start of the recursion (time τ), the above condition would hold with $k = 1$. The rest of the argument will be different, depending on whether we have $m + kC \leq M - kC$ ("Type I"), $C \leq M - kC < m + kC$ ("Type II"), or $M - kC < C \leq m + kC$ ("Type III").

Type I Suppose that $m + kC \leq M - kC$. We couple the channel states and arrivals by letting $\tilde{g}(n) = g(n)$ and $\tilde{a}(n) = a(n)$. For queues $l \neq 1, 2$, we let $\tilde{\pi}$ take the same action as π, that is, $\tilde{w}_l(n) = w_l(n)$ and $\tilde{h}_l(n) = h_l(n) = 0$, resulting in $\tilde{b}_l(n+1) = b_l(n+1)$. If π serves queue 1, bringing it down to $M - w_1(n) = M - C$, policy $\tilde{\pi}$ also removes C packets from queue 1. (This is possible because $M - kC \geq m + kC \geq C$.) If π removes $w_2(n)$ packets from queue 2 (note that either $w_2(n) = C$ if $m \geq C$, or $w_2(n) = m$ otherwise), then $\tilde{\pi}$ effectively removes the same number of packets from queue 2. (This is done by removing C packets and then adding $\tilde{h}_2(n) = C - w_2(n)$ packets.)

The resulting configurations are of the form

$$b(n+1) = \begin{bmatrix} M - w_1(n) \\ m - w_2(n) \\ s' \end{bmatrix} + a(n),$$

$$\tilde{b}(n+1) = \begin{bmatrix} M - w_1(n) - kC \\ m - w_2(n) + kC \\ s' \end{bmatrix} + a(n).$$

Note that $b_1(n+1) \geq \tilde{b}_1(n+1)$, and also $b_1(n+1) = M - w_1(n) \geq M - C \geq M - kC \geq m + kC \geq m - w_2(n) + kC = \tilde{b}_2(n+1)$. Thus, $\tilde{b}(n+1) \sqsubseteq b(n+1)$, as desired.

Type II Suppose now that $C \leq M - kC < m + kC$. We let $\tilde{g}_1(n) = g_2(n)$, $\tilde{g}_2(n) = g_1(n)$, and $\tilde{g}_l(n) = g_l(n)$ for $l > 2$. That is, we "couple" the channel state for queue 1 in the new system with the channel state for queue 2 in the original system, and *vice versa*. For all other queues, the channel states are the same in the two systems. We also let $\tilde{a}(n) = a(n)$.

For queues $l \neq 1, 2$, we let $\tilde{\pi}$ take the same action as π, that is, $\tilde{w}_l(n) = w_l(n)$ and $\tilde{h}_l(n) = h_l(n) = 0$, resulting in $\tilde{b}_l(n+1) = b_l(n+1)$. For $i = 1, 2$, let $\alpha_i = 1$ if π serves queue i, and $\alpha_i = 0$, otherwise.

Regarding queue 1, policy π brings it down to $M - \alpha_1 C$. Accordingly, we let $\tilde{\pi}$ take the same action for queue 2, bringing it down to $m + (k - \alpha_1)C$. This is possible because $\tilde{g}_2(t) = g_1(t)$ (if π can serve queue 1, then $\tilde{\pi}$ can serve queue 2) and $k \geq 1$.

We now have two subcases.

(a) If π serves queue 2 and removes $w_2(n) = C$ packets (this happens when $\alpha_2 = 1$ and $m \geq C$), or if π does not serve queue 2 ($\alpha_2 = 0$), we let $\tilde{\pi}$ remove the same number of packets from queue 1. The resulting configurations are of the form

$$b(n+1) = \begin{bmatrix} M - \alpha_1 C \\ m - \alpha_2 C \\ s' \end{bmatrix} + a(n)$$

$$\tilde{b}(n+1) = \begin{bmatrix} M - \alpha_2 C - kC \\ m - \alpha_1 C + kC \\ s' \end{bmatrix} + a(n).$$

For all four possible values of (α_1, α_2), we have $m - \alpha_2 C \leq M - \alpha_2 C - kC$ (this is because $m \leq M - kC$) and $m - \alpha_2 C \leq m \leq m - \alpha_1 C + kC$ (because $\alpha_1 \leq 1$ and $k \geq 1$). It follows that $\tilde{b}(n+1) \sqsubseteq b(n+1)$. (To see this, use the definition of $\sqsubseteq$ but with k replaced by $k + \alpha_2 - \alpha_1$.)

(b) If π serves queue 2 and removes $w_2(n) = m \neq C$ packets (this happens when $m < C$), then $\tilde{\pi}$ removes C packets from queue 1, and adds $\tilde{h}_2(n) = C - m$ packets to queue 2. The resulting configurations are of the form

$$b(n+1) = \begin{bmatrix} M - \alpha_1 C \\ 0 \\ s' \end{bmatrix} + a(n)$$

$$\tilde{b}(n+1) = \begin{bmatrix} M - C - kC \\ m - \alpha_1 C + kC + (C - m) \\ s' \end{bmatrix} + a(n),$$

where $\alpha_1 \in \{0, 1\}$ indicates whether queue 1 is served by π. It follows that $\tilde{b}(n+1) \sqsubseteq b(n+1)$, with k replaced by $k + 1 - \alpha_1$.

Type III Suppose, finally, that $M - kC < C \leq m + kC$. Similar to the discussion of Type II, we let $\tilde{g}_1(n) = g_2(n)$, $\tilde{g}_2(n) = g_1(n)$, and $\tilde{g}_l(n) = g_l(n)$ for $l > 2$. For queues $l \neq 1, 2$, we let $\tilde{\pi}$ take the same action as π, that is, $\tilde{w}_l(n) = w_l(n)$ and $\tilde{h}_l(n) = h_l(n) = 0$, and let $\tilde{a}_l(n) = a_l(n)$, resulting in $\tilde{b}_l(n+1) = b_l(n+1)$. As before, let $\alpha_1 = 1$ if π serves queue 1, and $\alpha_1 = 0$, otherwise.

If π serves queue 1, bringing it down to $M - w_1(n) = M - C$, policy $\tilde{\pi}$ removes C packets from queue 2, bringing it down to $m + (k-1)C$. This is possible because $\tilde{g}_2(t) = g_1(t) = 1$ and $\tilde{b}_2(n) = m + kC \geq C$.

We now have two subcases:

(a) If π does not serve queue 2, we let $\tilde{a}_1(n) = a_1(n)$ and $\tilde{a}_2(n) = a_2(n)$, and the resulting configurations are

$$b(n+1) = \begin{bmatrix} M - \alpha_1 C \\ m \\ s' \end{bmatrix} + a(n)$$

$$\tilde{b}(n+1) = \begin{bmatrix} M - kC \\ m + (k - \alpha_1)C \\ s' \end{bmatrix} + a(n),$$

which shows that $\tilde{b}(n+1) \sqsubseteq b(n+1)$, with k replaced by $k - \alpha_1$.

(b) If π does serve queue 2, then it drives it to zero, because $m \leq M - kC < C$. Then, $\tilde{\pi}$ serves queue 1, driving it also to zero (since $\tilde{b}_2(n) = M - kC < C$), and adds $M - (m + kC)$ packets to queue 2 driving it to $M - \alpha_1 C$. We finally let $\tilde{a}_1(n) = a_2(n)$, $\tilde{a}_2(n) = a_1(n)$. The resulting configurations are of the form

$$b(n+1) = \begin{bmatrix} M - \alpha_1 C \\ 0 \\ s' \end{bmatrix} + \begin{bmatrix} a_1(n) \\ a_2(n) \\ a' \end{bmatrix}$$

$$\tilde{b}(n+1) = \begin{bmatrix} 0 \\ M - \alpha_1 C \\ s' \end{bmatrix} + \begin{bmatrix} a_2(n) \\ a_1(n) \\ a' \end{bmatrix}$$

where $a' = (a_3(n), \ldots, a_N(n))$. In this case, $\tilde{b}(n+1)$ is a permutation of $b(n+1)$, and we again have $\tilde{b}(n+1) \sqsubseteq b(n+1)$ (case (ii) in the definition of $\sqsubseteq$).

At this point, we have completed the recursive construction of $\tilde{\pi}$. Under the new policy, we have $\tilde{b}(n) \sqsubseteq b(n)$, which implies that $\tilde{b}(n) \preceq b(n)$, for all n, so that $\tilde{\pi}$ dominates π. Furthermore, by construction (cf. Step A), we have $\tilde{\pi} \in \mathcal{L}_\tau^{\ell-1}$. **q.e.d.**

Proof of Lemma 2: We use again a coupling argument. Suppose that $\tilde{\pi} \in \mathcal{L}_\tau^{\ell}$. We let $\hat{\pi}$ and $\tilde{\pi}$ coincide until time $\tau - 1$. In particular, $\hat{\pi}$ has the LCQ property until time $\tau-1$, and $\hat{b}_\tau = \tilde{b}_\tau$. From time τ onward, the process associated to $\tilde{\pi}$ evolves according to

$$\tilde{b}(n+1) = \tilde{b}(n) - \tilde{w}(n) + \tilde{h}(n) + \tilde{a}(n), \qquad n \geq \tau.$$

We let $\hat{a}(n) = \tilde{a}(n)$ and $\hat{g}(n) = \tilde{g}(n)$ for all times. We let the policy $\hat{\pi}$ correspond to the evolution equation

$$\hat{b}(n+1) = \max\{0, \hat{b}(n) - \tilde{w}(n)\} + \tilde{a}(n), \qquad n \geq \tau.$$

(The "$\max$" operation on vectors is defined componentwise.) Thus, $\hat{\pi}$ acts similar to $\tilde{\pi}$ but never adds any packets, so that $\hat{\pi} \in \Phi_\infty$. Furthermore, $\hat{\pi}$ serves the same queues as $\tilde{\pi}$ at time τ, and therefore has the same deviation ℓ from LCQ. It follows that $\hat{\pi} \in \mathcal{L}_\tau^{\ell} \cap \Phi_\infty$.

An easy inductive argument shows that $\hat{b}(n) \leq \tilde{b}(n)$, for all n. Indeed, assuming this to be true for some $n \geq \tau$, we have $\hat{b}(n) - \tilde{w}(n) \leq \tilde{b}(n) - \tilde{w}(n) + \tilde{h}(n)$, and also $\tilde{b}(n) - \tilde{w}(n) + \tilde{h}(n) \geq 0$, which imply that $\max\{0, \hat{b}(n) - \tilde{w}(n)\} \leq \tilde{b}(n) - \tilde{w}(n) + \tilde{h}(n)$, and $\hat{b}(n+1) \leq \tilde{b}(n+1)$.

Also, if $\hat{\pi}$ serves queue i at time n, it is seen that it removes $\min\{\hat{b}_i(n), C\}$ packets, the maximum possible number. Indeed, if $\tilde{b}_i(n) \geq C$, then $\tilde{w}_i(n) = C$, so that the number of packets removed by $\hat{\pi}$ is $\min\{\hat{b}_i(n), C\}$. Otherwise, if $\tilde{b}_i(n) < C$, then $\tilde{w}_i(n) = \tilde{b}_i(n) \geq \hat{b}_i(n)$, and $\hat{\pi}$ removes $\hat{b}_i(n)$ packets.

Since $\hat{\pi}$ is a legitimate policy in $\mathcal{L}_\tau^{\ell} \cap \Phi_\infty$, and $\hat{b}(n) \leq \tilde{b}(n)$ for all n, it follows that $\hat{b}(n) \preceq \tilde{b}(n)$ for all n and, therefore, $\hat{\pi} \preceq \tilde{\pi}$. **q.e.d.**

REFERENCES

[1] M. Andrews, A. Stoylar, K. Kumaran, R. Vijayakumar, K. Ramanan, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, pp. 191–217, 2004.

[2] M. Armory and N. Bambos, "Queueing dynamics and maximal throughput scheduling in switched processing systems," *Queueing Syst.*, vol. 44, no. 3, pp. 209–252, 2003.

[3] N. Bambos and G. Michailidis, "On parallel queueing with random server connectivity and routing constraints," *Probab. Eng. Inf. Sci.*, vol. 16, no. 3, Apr. 2002.

[4] S. Shakkottai, R. Srikant, and A. L. Stolyar, "Pathwise optimality of the exponential scheduling rule for wireless channels," in *Proc. The High-Speed Networking Workshop (HSN)*, New York, Jun. 2002.

[5] R. A. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[6] A. Ganti, "Transmission Scheduling for Multi-Beam Satellite Systems," Ph.D. dissertation, Dept. EECS, MIT, Cambridge, MA, 2002.

[7] M. Hazewinkel, *Encyclopaedia of Mathematics*. Norwell, MA: Kluwer Academic, 1988, vol. 6.

[8] M. Neely, E. Modiano, and C. Rohrs, "Power allocation and routing in multi-beam satellites with time varying channels," *IEEE/ACM Trans. Networking*, vol. 11, no. 3, pp. 138–152, Jun. 2003.

[9] D. Stoyan, *Comparison Methods for Queues and other Stochastic Models*. New York: Wiley, 1983.

[10] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[11] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.