Transmission Scheduling for Multi-Channel Satellite and Wireless Networks *

Anand Ganti, Eytan Modiano, John N. Tsitsiklis
Massachusetts Institute of Technology
e-mail: aganti@mit.edu, modiano@mit.edu, jnt@mit.edu

Abstract

We consider a slotted system with N queues, and i.i.d. Bernoulli arrivals at each queue during each slot. Each queue is associated with a beam and 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_0 packets from a queue associated with an "on" channel. We show that when K is arbitrary and $C_0 = 1$, as well as when $K = 1$ and C_0 is arbitrary, a policy that assigns the K servers to the "on" channels associated with the K longest queues is optimal. We also consider a "fluid" model under which fractional packets can be served, for the case $K = N$, and subject to a constraint that at most C packets can be served in total over all of the N queues. We show that there is an optimal policy which serves the queues so that the resulting vector of queue lengths is "Most Balanced."

1 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 one of its neighboring satellites, and are stored in on-board buffers. In this context, there is often only a limited number of transmitters on-board the satellite, 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. Similarly, 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. This scheduling problem has received attention recently in the context of next generation wireless data systems [2, 3, 4, 5, 6, 7, 8, 10].

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

*This research is supported in part by NSF Grant No. NCR-9627610 and by DARPA under the Next Generation Internet Initiative

the numbers of arrivals to the ith queue at time n, 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 at time n, denoted by $G_i(n)$, can only take one of two values, namely, 0 or 1. We designate 0 as an "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. Again, we assume that the $G_i(n)$ are independent and identically distributed 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.

The satellite has K transmitters. Each transmitter can only serve one queue, and can only be assigned to a queue whose channel is "on." At each slot, each transmitter can transmit up to C_0 packets, and the total number of transmitted packets cannot exceed C. The parameter C_0 corresponds to the power limitation on an individual transmitter and C corresponds to the power limitation on the entire satellite.

Such queueing systems, with multiple queues and stochastically varying service rates, have been studied in [2] where the authors show that, when $K = 1$ (single transmitter), a policy called the Longest Connected Queue (LCQ) maximizes the stability region of the system, and also results in optimal average queue lenghts. Furthermore, [3] shows that the LCQ policy results in a maximal stability region under more general assumptions on the arrival and channel state processes. The models considered in the present work extend this previous work in various directions.

2 The case of multiple servers and $C_0 = 1$

We first consider the case where $C = K$ and $C_0 = 1$. That is, there are K servers, and each server can serve at most one packet during each time slot. This is a generalization of the system studied in [2], which dealt with the case where $K = 1$. We now describe a policy that will be shown to be optimal in the current scenario, as well as in the case where there is a single server that can serve upto C_0 packets (see Section 3).

The **Longest Connected Queues (LCQ)** policy 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 $\min\{B_i(n), C_0\}$ packets from each one of them. Note that fewer than K queues will be selected 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 transmitter, the policy can choose between them arbitrarily.

Our main result states that the LCQ policy is optimal in the stochastic ordering sense [1]. The intuitive reason is that this policy tries to keep the queued packets spread over multiple queues and has a better chance of avoiding idling when some channels are off.

Theorem 1 *Assume that $C_0 = 1$ and $C = K$. Let $Q(n) = \sum_{i=1}^{N} B_i(n)$ be the total number of packets, starting from a given initial state, under some policy π. Let $Q^L(n)$ be the total number of packets when the LCQ policy is used. Then, $\{Q^L(n)\} \overset{st}{\leq} \{Q(n)\}$, i.e., the process $\{Q^L(n)\}$ stochastically dominates the process $\{Q(n)\}$.*

A result of this type was proved in [2], for the case $K = 1$, using a coupling argument. Our proof for the case of general K also uses a coupling argument. For brevity, we only provide an outline here and refer the reader to [9] for complete proofs.

Let us note that optimality in the stochastic ordering sense is a rather strong notion. It implies the optimality of the LCQ policy over either a finite or an infinite horizon, involving one-step costs of the form $E[f(Q(n))]$, where f is any nondecreasing function. The same comment applies to Theorem 2 later in this paper.

Let $\mathcal{Z}_+^N$ be the set of nonnegative integers. We use $\Pi_d : \Re^N \mapsto \Re^N$ to denote the function that sorts its arguments in decreasing order. We also let

$$\Delta(N) \triangleq \{\delta \in \{-1, 0, 1\}^N \mid \delta_i > 0 \Rightarrow [\delta_j \geq 0, \ \forall j < i]\}$$

be the set of vectors whose components are either minus one, zero, or one, with all the ones appearing before all the negative ones. For example, $[0, 1, 0, 1, 1, 0, -1, 0, -1]$ belongs to $\Delta(9)$. Finally, we use

$$Z(N) \triangleq \{b \in \mathcal{Z}_+^N : b_1 \geq b_2 \geq \cdots \geq b_N\}$$

to denote the set of nonnegative vectors whose components are nonincreasing. For example, $[9, 2, 2, 1]$ belongs to $Z(4)$.

Proof Outline: We consider a system (to be referred to as "system 0") that starts at an initial state $(b(0), g(0))$ and is controlled by policy π. We will construct a policy π^1 and a corresponding "system 1" on the same probability space, by appropriately coupling the arrival, service, and channel state processes, so that π^1 acts similar to LCQ at $n = 0$, and so that $Q^1(n) \leq Q(n)$ at all times n, for every sample path. We will use a superscript 1 to denote various quantities, such as total queue length or channel state, under policy π^1 in system 1.

Both systems start with the same initial state $(b(0), g(0))$. If π behaves like LCQ at time 0, we let π^1 be equal to π, and we couple the two systems so that they evolve in an identical fashion. In this case, the inequality $Q^1(n) \leq Q(n)$ holds with equality for all n.

Suppose now that at time 0, π behaves differently from LCQ. We construct system 1 and policy π^1 as follows.

At time 0, we have the same channel states $g(0)$ in both systems. We apply an LCQ policy to system 1. After subtracting the packet withdrawals at time 0, we arrange the queue lengths of system 0 and system 1 in decreasing order We then couple any arrival to the ith longest queue of system 0 to an arrival to the ith longest queue of system 1.

Let $\beta(n)$ be the vector of queue lengths in system 0, arranged in decreasing order, so that $\beta_1(n) \geq \beta_2(n) \geq \cdots \geq \beta_N(n)$. Similarly define $\beta^1(n)$ for system 1. Let $\delta(n) = \beta(n) - \beta^1(n)$

The following lemma (whose proof we omit) shows that $\delta(1) \in \Delta(N)$, a result we will use in describing the coupling procedure for $n \geq 1$. The first part of the lemma shows that if we first subtract the packet withdrawals under either policy, and then sort the resulting queue length vectors, the difference belongs to the set $\Delta(N)$. The second part establishes that after coupling the arrivals at time 0, as described earlier, the vector $\delta(1)$ belongs to $\Delta(N)$.

Lemma 1 *(a) Let $b \in Z(N)$ and $g \in \{0, 1\}^N$. Let r be a feasible packet withdrawal vector, and let $\tilde{r}$ be the LCQ control, in state (b, g). Then $\Pi_d(b - r) - \Pi_d(b - \tilde{r}) \in \Delta(N)$.*
(b) Let $e \in \{0, 1\}^N, b, \tilde{b} \in Z(N), b - \tilde{b} \in \Delta(N)$. Then $\Pi_d(b + e) - \Pi_d(\tilde{b} + e) \in \Delta(N)$.

We now describe the coupling and the policy π^1 at positive times n.

Case I: Suppose that there exists some i such that $\delta_i(1) = 1$ and $\beta_i^1(1) = 0$. Since $\delta(1) \in \Delta(N)$ (Lemma 1), we must have $\delta(1) \geq 0$, i.e., the jth longest queue in system 0 has no fewer packets than the jth longest queue in system 1, for every j. We then let the channel states, controls, and arrivals for the jth longest queue in system 1 be the same as for the jth longest queue in system 1, at all positive times n. We obtain $Q^1(1) \leq Q(1)$, and the coupling also preserves this inequality for all times n.

Case II: Suppose now that for every i satisfying $\beta_i^1(1) = 0$, we must also have $\delta_i(1) = 0$, which implies $\beta_i(1) = 0$. We map the channel state of the ith longest queue in system 0 to the the ith longest queue in system 1. If π serves a packet from the ith longest queue in system 0, we have $\beta_i(1) > 0$, which implies that $\beta_i^1(1) > 0$, and we let π^1 serve a packet from the ith longest queue in system 1. To couple the arrivals, we again subtract the packet withdrawals, arrange the queue lengths in system 0 and 1, respectively, in decreasing order, and couple any arrival to the ith longest queue of system 0 to an arrival to the ith longest queue of system 1.

The next lemma (whose proof is omitted) states that $\delta(2) \in \Delta(N)$, which we can then use to repeat the coupling at time 1, for times $n = 2, 3, \ldots$.

Lemma 2 *Let* $e \in \{0,1\}^N, b, \tilde{b} \in Z(N), b-\tilde{b} \in \Delta(N)$. *Then* $\Pi_d(b-e)-\Pi_d(\tilde{b}-e) \in \Delta(N)$.

For $n = 2, 3, \ldots$, either the coupling has already been determined (if at some prior time the condition for Case I was satisfied), or else we repeat the above coupling procedure and use the fact

$$\delta(n-1) \in \Delta(N) \Rightarrow \delta(n) \in \Delta(N)$$

(Lemmas 1 and 2).

We now show that $Q^1(n) \leq Q(n)$ for all n. This statement clearly holds when $n = 0$. We have also established it for Case I. For Case II, we have already noted that we can always withdraw as many packets in system 1 as in system 0, which implies our assertion for all n.

We have established that for every policy π, there exists a policy π^1 that agrees with the LCQ policy at time 0, and such that $Q^1(n) \leq Q(n)$, for all n. By repeating the construction, we obtain policies $\pi^1, \pi^2, \ldots$ such that the policy π^k agrees with LCQ until time $k-1$, and such that $Q^k(n) \leq Q^{k-1}(n-1) \leq \cdots \leq Q^1(n) \leq Q(n)$, for all n. This statement, in conjunction with a standard characterization of stochastic dominance [1] completes the proof. □

3 The single server case

We now consider a variant of the problem in which there is a single server. This is another extension of the problem studied in [2], where the server could only serve one packet per slot. In our generalization, the server can serve up to C_0 packets during the same slot, but they must all be taken from the same queue. Without loss of generality, we assume that if a queue with $B_i(n)$ packets is served, then the number of packets withdrawn is equal to $\min\{B_i(n), C_0\}$. We have the following result, which is again proved using a coupling argument.

Theorem 2 *Assume that* $K = 1$. *Let* $Q(n) = \sum_{i=1}^{N} B_i(n)$ *be the total number of packets, starting from a given initial state, under some policy* π. *Let* $Q^L(n)$ *be the total number of packets when the LCQ policy is used. Then,* $\{Q^L(n)\} \overset{st}{\leq} \{Q(n)\}$.

For the purposes of the proof of this theorem, we expand the set of policies to include policies that may add extra packets to queues. We let Φ_a be the expanded set of policies. We let Φ_{na} ("no additions") be the set of all policies that never add extra packets. We say that a policy has the *LCQ property* at time t if at that time it can only serve packets from a longest connected queue. We let $\mathcal{L}_\tau$ be the set of all policies that have the LCQ property for all times $t \leq \tau$, and which do not add any packets before time τ. We will use $a(n)$ to denote arrivals, $g(n)$ to denote channel states, $b(n)$ to denote queue sizes, and $h_i(n)$ to denote packets added to queue i, at time n.
Proof Outline: The core of the proof consists of the following two steps.

0. We start with a policy π in $\mathcal{L}_{\tau-1} \cap \Phi_{na}$ acting on a queueing system with initial state $(b(0), g(0))$. We call this system 0. Such a policy has the LCQ property until time $\tau - 1$, and never introduces additional packets.

1. We construct a new policy $\tilde{\pi}$ in $\mathcal{L}_\tau$ (i.e., $\tilde{\pi}$ has the LCQ property until time τ, and does not introduce additional packets before time τ), and a corresponding system 1 on the same probability space, so that $\tilde{\pi}$ "dominates" π, in the sense that $Q^1(n) \leq Q(n)$ for all n.

(We use a superscript of 1 to denote various quantites in system 1 and no superscript to denote related quantities in system 0.)

We now describe the construction of $\tilde{\pi}$. Before time τ, we let $\tilde{\pi}$ be identical to π, and let arrivals and channel states be the same for both systems. In particular, $b(\tau) = b^1(\tau)$. At time τ, we let the state of each channel be the same for both systems.

Step 1.1. Policy $\tilde{\pi}$ at time τ.

If π serves a longest connected queue at time τ, we let $\tilde{\pi}$ do the same, and set $a^1(\tau) = a(\tau)$, $h^1(\tau) = 0$. In this case, we have $b^1(\tau + 1) = b(\tau + 1)$.

Suppose now that π chooses to serve at time τ some queue which is not a longest connected queue. Without loss of generality, let us assume that π serves queue 2 and that queue 1 is a longest connected queue. Thus, $b(\tau)$ 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 $C_0 \leq m < M$. Policy π removes C_0 packets from queue 2. We let policy $\tilde{\pi}$ remove C_0 packets from queue 1. We also let $h^1(\tau) = 0$ and $a^1(\tau) = a(\tau)$. The resulting configurations are of the form

$$b(\tau + 1) = \begin{bmatrix} M \\ m - C_0 \\ s \end{bmatrix} + a(\tau), \qquad b^1(\tau + 1) = \begin{bmatrix} M - C_0 \\ m \\ s \end{bmatrix} + a(\tau).$$

(ii) Suppose that $m < M \leq C_0$. 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., $h_2^1(\tau) = M - m$. We also let

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

The resulting configurations are of the form

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

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

(iii) Suppose finally that $m < C_0 < M$. In this case, π drives m down to zero. We let $\tilde{\pi}$ serve queue 1 and remove C_0 packets. We also let $\tilde{\pi}$ add $C_0 - m$ packets to queue 2, i.e., $h_2^1(\tau) = C_0 - m$, driving it up to C_0. The resulting configurations are of the form

$$b(\tau+1) = \begin{bmatrix} M \\ 0 \\ s \end{bmatrix} + a(\tau), \qquad b^1(\tau+1) = \begin{bmatrix} M - C_0 \\ C_0 \\ s \end{bmatrix} + a(\tau).$$

This completes the description of policy $\tilde{\pi}$ at time τ. We will now construct $\tilde{\pi}$, for times $t > \tau$, so that at any time one of the following three relationships holds:

(i) $b(t) = b^1(t)$.

(ii) $b(t)$ and $b^1(t)$ differ only in their first two components, and

$$b_1(t) = b_2^1(t), \qquad b_2(t) = b_1^1(t).$$

(iii) $b(t)$ and $b^1(t)$ differ only in their first two components, and there exists a positive integer k such that for either $(i,j) = (1,2)$ or for $(i,j) = (2,1)$, we have

$$b_j(t) \le \min\{b_1^1(t), b_2^1(t)\} \le \max\{b_1^1(t), b_2^1(t)\} \le b_i(t),$$

$$b_i^1(t) = b_i(t) - kC_0, \qquad b_j^1(t) = b_j(t) + kC_0,$$

We will use $b^1(t) \sqsubseteq b(t)$ to indicate that $b(t)$ and $b^1(t)$ are related in one of the above three ways. Note that our construction of $\tilde{\pi}$ at time τ guarantees that $b^1(\tau+1) \sqsubseteq b(\tau+1)$.
Step 1.2. Policy $\tilde{\pi}$ at times $t > \tau$.

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

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

Case (ii): Suppose that $b^1(t)$ is obtained from $b(t)$ by permuting the first two components. For queues $i \notin \{1,2\}$ 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 system 0 be the same as for queue 2 in system 1, and vice versa. Then, the last $N-2$ components of $b^1(t+1)$ and $b(t+1)$ are equal, whereas the first two remain permutations of each other. In particular, $b^1(t+1) \sqsubseteq b(t+1)$.

Case (iii): We finally consider the remaining case (iii) in the definition of $\sqsubseteq$. Without loss of generality, we assume that the first component of $b(t)$ is at least as large as the second component. In particular, for some m and M, with $m < M$, for some positive integer k, and for some $s \in \mathcal{Z}_+^{N-2}$, we have

$$b(t) = \begin{bmatrix} M \\ m \\ s \end{bmatrix}, \qquad b^1(t) = \begin{bmatrix} M - kC_0 \\ m + kC_0 \\ s \end{bmatrix}.$$

The rest of the argument will be different, depending on whether we have $m + kC_0 \le M - kC_0$ ("Type I") or $m + kC_0 > M - kC_0$ ("Type II").

Type I: Suppose that $m+kC_0 \le M-kC_0$. We couple the channel states and arrivals by letting $g^1(t) = g(t)$ and $a^1(t) = a(t)$. (a) If π serves queue 1, bringing it down to $M - C_0$, policy $\tilde{\pi}$ also removes C_0 packets from queue 1. (This is posssible because $M - kC_0 \ge m + kC_0 \ge C_0$. The resulting configuration is

$$b(t+1) = \begin{bmatrix} M - C_0 \\ m \\ s \end{bmatrix} + a(t), \qquad b^1(t+1) = \begin{bmatrix} M - (k+1)C_0 \\ m + kC_0 \\ s \end{bmatrix} + a(t),$$

and we have $b^1(t+1) \sqsubseteq b(t+1)$. (b) If π removes u packets from queue 2 (note that either $u = C_0$ if $m \ge C_0$, or $u = m$ otherwise), then $\tilde{\pi}$ removes the same number of packets from queue 2. (This is done by removing C_0 packets and then adding $h_2^1(t) = C_0 - u$ packets.) The resulting configuration is

$$b(t+1) = \begin{bmatrix} M \\ m-u \\ s \end{bmatrix} + a(t), \qquad b^1(t+1) = \begin{bmatrix} M - kC_0 \\ m - u + kC_0 \\ s \end{bmatrix} + a(t).$$

(c) Finally, if π serves some queue $j > 2$, we let $\tilde{\pi}$ do the same. In all subcases, we have $b^1(t+1) \sqsubseteq b(t+1)$.

Type II: Suppose now that $m + kC_0 > M - kC_0$. We let $g_1^1(t) = g_2(t)$, $g_2^1(t) = g_1(t)$, and $g_j^1(t) = g_j(t)$ for $j > 2$. That is, we "couple" the channel state for queue 1 under policy $\tilde{\pi}$ to that for queue 2 under policy π, and vice versa. For all other queues, channel states coincide under the two policies.

(a) Suppose that π serves some queue $j > 2$. Then, $\tilde{\pi}$ removes the same number of packets from the same queue, which is possible because $b_j^1(t) = b_j(t)$ and $g_j^1(t) = g_j(t)$. We then let $a^1(t) = a(t)$.

(b) Suppose that π serves queue 1 (in particular, $g_1(t) = 1$), bringing it down to $M - C_0$. We then let $\tilde{\pi}$ serve queue 2, bringing it down to $m + (k-1)C_0$. This is possible because $g_2^1(t) = g_1(t) = 1$ and $b_2^1(t) = m + kC_0 \ge C_0$. We then let $a^1(t) = a(t)$. The resulting configuration is

$$b(t+1) = \begin{bmatrix} M - C_0 \\ m \\ s \end{bmatrix} + a(t), \qquad b^1(t+1) = \begin{bmatrix} M - kC_0 \\ m + (k-1)C_0 \\ s \end{bmatrix} + a(t).$$

Notice that $m \le m + (k-1)C_0$, because k is positive. In particular,

$$m \le \min\{M - kC_0, m + (k-1)C_0\} \le \max\{M - kC_0, m + (k-1)C_0\} \le M - C_0.$$

We then see that $b^1(t+1) \sqsubseteq b(t+1)$. (If $k = 1$, we have case (i) in the definition of $\sqsubseteq$; if $k > 1$, we have case (iii).)

(c) Suppose, finally, that π serves queue 2 and removes $u = \min\{m, C_0\}$ packets, bringing it down to $m - u$. In particular, $g_1^1(t) = g_2(t) = 1$. If $M - kC_0 \geq C_0$, then, $\tilde{\pi}$ removes C_0 packets from queue 1, bringing it to $M - (k+1)C_0$, and adds $h_2^1(t) = C_0 - u$ packets to queue 2, bringing it to $m + (k+1)C_0 - u$. We then let $a^1(t) = a(t)$. The resulting configuration is

$$b(t+1) = \begin{bmatrix} M \\ m-u \\ s \end{bmatrix} + a(t), \qquad b^1(t+1) = \begin{bmatrix} M-(k+1)C_0 \\ m-u+(k+1)C_0 \\ s \end{bmatrix} + a(t).$$

We then have $b^1(t+1) \sqsubseteq b(t+1)$ (case (iii) in the definition of $\sqsubseteq$). It remains to consider the case where $M - kC_0 < C_0$. In that case, we have $m \leq M - kC_0 \leq C_0$, so that π drives queue 2 down to zero. Policy $\tilde{\pi}$, drives queue 1 from $M - kC_0$ down to zero, and it adds enough packets to queue 2 to drive it up to M. We then let $a_1^1(t) = a_2(t), a_2^1(t) = a_1(t)$ and $a_j^1(t) = a_j(t)$, for $j > 2$. The resulting configuration is

$$b(t+1) = \begin{bmatrix} M \\ 0 \\ s \end{bmatrix} + \begin{bmatrix} a_1(t) \\ a_2(t) \\ [a_3(t), \ldots, a_N(t)]^T \end{bmatrix}, \qquad b^1(t+1) = \begin{bmatrix} 0 \\ M \\ s \end{bmatrix} + \begin{bmatrix} a_2(t) \\ a_1(t) \\ [a_3(t), \ldots, a_N(t)]^T \end{bmatrix},$$

and we have $b^1(t+1) \sqsubseteq b(t+1)$ (case (ii) in the definition of $\sqsubseteq$).

At this point, we have completed the recursive construction of $\tilde{\pi}$. The new policy $\tilde{\pi}$ dominates π, has the LCQ property until time τ, and does not add any packets before time τ.

The remainder of the proof of Theorem 2, which we omit, proceeds as follows. We modify the policy $\tilde{\pi}$ to obtain yet another policy $\hat{\pi}$ in $\mathcal{L}_\tau \cap \Phi_{na}$ (in particular, $\hat{\pi}$ never adds extra packets), and which dominates $\tilde{\pi}$. We then repeat the preceding steps for $\tau = 0, 1, \ldots$, to obtain a policy with the LCQ property at all times and with no added packets, and which dominates the policy that we started with. □

4 The case of one server per queue ($K = N$)

In this section, we assume that packets can be served from all connected queues simultaneously, subject to a bound C on the total number of packets that can be served during one time slot. If we further assume that $C = 1$, we are back to the model in [2], and the LCQ policy is optimal. The case of general C is an open problem. However, if we slightly modify the model to allow serving a noninteger number of packets from each queue, we can show that a generalization of LCQ, which we call the "Most Balanced" policy is optimal.

Suppose that at time n we have $(B(n), G(n)) = (b, g)$, for some vectors b and g. Let $U(b, g)$ denote the set of feasible vectors of packet withdrawals, when the system is in state (b, g), i.e,

$$U(b,g) = \left\{ u \in \Re_+^N \;\middle|\; \text{if } g_i = 0 \text{ then } u_i = 0;\ \sum_{i=1}^N u_i \leq C \right\}.$$

The **Most Balanced** policy chooses a $u \in U(b, g)$ so that $\sum_i u_i$ is as large as possible, and in addition, so that it minimizes

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

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 $u = [1.5, 0.5, 0, 0, 0, 0]$, resulting in the configuration $b - u = [3.5, 3.5, 3, 2, 6, 1]$. It is not hard to show that the most balanced policy is uniquely defined. Finally, we let $\mathcal{F}$ be the set of all functions $f : \Re_+^N \mapsto \Re$ that are convex, nondecreasing, and symmetric (permutation invariant).

Theorem 3 *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 Most Balanced policy minimizes $E[f(B(n))]$.*

Proof Outline: The proof uses a dynamic programming argument. Let $V_n^*(b, g)$ be the least possible value of $E[f(B(n))]$, starting from the initial state (b, g) at time zero. We then have $V_0^*(b, g) = f(b)$, and

$$V_n^*(b, g) = \min_{w \in U(b,g)} \left[\sum_a \sum_{\tilde{g}} p_A(a) p_G(\tilde{g}) V_{n-1}^*(b + a - w, \tilde{g}) \right],$$

for every positive n. Here, $p_G(\tilde{g})$ is the probability that the next vector of channel states is $\tilde{g}$, and $p_A(a)$ is the probability that the vector of queue arrivals is a.

We use the above equation and induction to show that the functions V_n^* belong to $\mathcal{F}$, for all n. The convexity and symmetry of $V^*(n)$, together with the form of the above dynamic programming equation, can be used to show that more balanced configurations are always preferable. □

Using Theorem 3, 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 $E[f(Q(n))]$ 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).

5 Discussion

Let us remark that Theorems 1 and 2 remain valid for certain generalizations of the model. For example, always assuming that the arrivals $A_i(n)$ are independent and identically distributed, the Bernoulli assumption can be relaxed. Instead, it suffices to assume that the random variable $A_i(n)$ can be expressed as a sum of independent Bernoulli random variables, with possibly different parameters. The special case of equal parameters allows the $A_i(n)$ to have a binomial distribution. For another generalization, we can relax the assumption that the distribution of the $A_i(n)$ is the same for different times n. For example, we can assume that it is Bernoulli with a parameter which is itself random and independent at different times (a deterministically changing parameter is a special case). In the same spirit, the probability that a channel is "on" can also vary with time.

It turns out that the results also remain valid in certain situations where the arrivals at different queues (respectively, the channel states) at a given time are dependent, as long as the dependence is "symmetric," in the sense that it leads to permutation-invariant distributions.

References

[1] D. Stoyan, *Comparison Methods for Queues and other Stochastic Models,* John Wiley, 1983.

[2] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity, *IEEE Transactions on Information Theory*, 1993, Vol. 39, No. 2, pp. 466-478.

[3] M. Andrews, A. Stoylar, K. Kumaran, R. Vijayakumar, K. Ramanan, and P. Whiting, "Scheduling in a Queueing System with Asynchronously Varying Service Rates", *IEEE Transactions on Communications,* Vol. 39, No. 2, 2001, pp. 466-478.

[4] R. A. Berry, and R. Gallager, "Communication Over Fading Channels with Delay Constraints," *IEEE Transactions on Information Theory,* Vol. 48, No. 5, May 2002, pp. 1135-1149.

[5] M. Neely, E. Modiano and C. Rohrs, "Power and Server Allocation in a Multi-Beam Satellite with Time Varying Channels," *IEEE INFOCOM 2002*, New York, June, 2002.

[6] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control,* Vol. 37, no. 12, Dec. 1992.

[7] N. Bambos and G. Michailidis, "On Parallel Queueing with Random Server Connectivity and Routing Constraints", *Probability in Engineering and Information Sciences,* 2002, in press.

[8] M. Armory and N. Bambos, "Queueing Dynamics and maximal Throughput Scheduling in Switched Processing Systems", technical report Net-Lab-2001-09.01, Stanford University, Stanford, CA 2001.

[9] A. Ganti, Transmission Scheduling for Multi-Beam Satellite Systems, doctoral thesis, dept. of EECS, MIT, Cambridge, MA, 2002.

[10] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, P. Whiting, "Scheduling in a Queueing System with Asynchronously Varying Service Rates, preprint, October 2000.