Scheduling with Asynchronous Service Opportunities with Applications to Multiple Satellite Systems

Michael Carr and Bruce Hajek

Abstract—**A single server is assigned to M parallel queues with independent Poisson arrivals. Service times are constant, but the server has the opportunity to initiate service at a given queue only at times forming a Poisson process. Four related scheduling policies are investigated. a) A simple first-come, first-serve policy for which the stability region is determined, b) A policy with maximum throughput, but requiring the server to have advance knowledge of service opportunities, c) A policy of threshold type, which is shown to be optimal among nonlookahead policies with preemption, and d) An adaptive policy, which when $M = 2$ is shown to provide stability for all arrival rate vectors for which stability is possible under any nonlookahead policy with preemption. The work is motivated by the problem of transmission scheduling for a packet-switched, low-altitude, multiple satellite system.**

I. Introduction

THE scheduling problem addressed in this paper arose in satellite communications, but the problem and the solutions proposed here may be of interest for other problems. In this section the particular application is described, the mathematical formulation is presented, and an overview of the rest of the paper is given.

Packet-switched, low-altitude, multiple satellite systems (MSS) have been proposed as a cost effective, robust means of providing continuous world-wide connectivity for integrated voice/data communications [2], [4]. Up to 240 low-altitude satellites with intelligent switching capabilities and steerable antennae would be used. Due to power limitations, it is desirable to schedule transmissions, rather than to allow collisions as in the classical ALOHA system. In the general MSS model, satellites travel in random but predictable orbits relative to each other, and crosslinks between neighboring satellites last for five to six minutes on average. The propagation delay between neighboring satellites would be on the order of hundreds of packet durations, and would be continuously changing. The distances of a satellite to its neighbors would change at different rates, and each satellite would know the locations of all satellites.

An approach to scheduling transmissions in this scenario, called Pseudo-Random Scheduling (PRS), was introduced by Binder *et al.* [2]. A related protocol called the Adaptive Receive Node Scheduling (ARNS) Protocol was introduced by Kosowsky *et al.* [4]. The basic idea of these protocols is the following. Each satellite uses a pseudo-random sequence which dictates when it listens for packet transmissions from each of its neighboring satellites. Moreover, the times when a satellite is in listen mode are composed of nonoverlapping periods, the length of each being the time needed to receive (or transmit) a packet. During each period the satellite is assigned exactly one neighbor to listen to, according to its pseudo-random sequence. This provides a capture effect to guard against potential collision interference.

Each satellite knows the pseudo-random sequences of all neighboring satellites. Some provision is made for a small amount of high order control information over specially dedicated slots. Such information is used to allow a given satellite to partially change its pseudo-random sequence in response to traffic demands, and the relevant nearby satellites are notified.

When a satellite is not listening for transmissions from neighboring satellites, it can schedule transmissions to these neighbors. Its transmissions cannot overlap in time, and they must occur at proper times so that the transmitted packets are being listened for. When a satellite is neither transmitting, nor listening to other satellites, it can communicate with terminals on the ground or seek new neighbors. The purpose of this paper is to consider methods for a satellite to schedule packet transmissions to neighboring satellites, that maximize either the total or weighted throughput to those neighbors.

Consider the time periods during which a particular satellite is not in the listening mode. It may have packets queued for sending to each of several neighboring satellites. Since the satellite knows the times that each of its neighbors will be listening to it, and can calculate the propagation delay to each of its neighbors, it can easily determine when it has opportunities to send to each of its

Manuscript received September 14, 1992; revised March 2, 1993. This research was supported in part by the Joint Services Electronics Program under Grant N00014-90-J-1270. Preliminary version presented at the IEEE International Communications Conference, June 1992, Chicago.

Michael Carr is with Raytheon Co., Missile Systems Division, Tewksbury, Massachusetts.

Bruce Hajek is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois.

IEEE Log Number 9213190.

0018-9286/93$03.00

neighbors. The opportunities to transmit to a particular neighbor may appear on a slotted time scale. However, the opportunities for different neighbors are superimposed, and no attempt to synchronize the transmission opportunities from different neighbors is made because the propagation delays to different neighbors change at different rates. Thus, the transmission opportunities occur at the satellite in a largely asynchronous fashion.

We refer to the control unit in the satellite responsible for transmission scheduling as the *server*. In view of the above discussion, we adopt the following mathematical formulation of the problem faced by the server in a particular satellite in transmit mode (see Fig. 1). The satellite has M neighbors. *Opportunities* to transmit to neighbor i are assumed to arrive according to a Poisson process of rate μ_i. *Packets* destined for neighbor i are assumed to arrive to the satellite's transmission queues according to a Poisson process with rate λ_i. The packet arrival processes and the streams of transmit opportunities are assumed to be mutually independent. Finally, the time it takes to transmit a packet is denoted by τ.

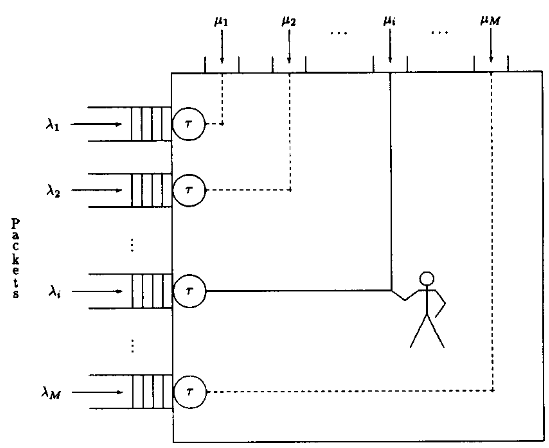


Fig. 1. Single-server queuing station model. Type i transmit opportunities arrive to the station at rate μ_i, and do not queue. Type i packets arrive at the station at rate λ_i, each requiring τ time units of service time for successful delivery to neighbor i.

Fig. 2 shows the transmit opportunities that might be offered to a particular satellite with three neighbors. A triangular "pulse" with label i represents an opportunity to transmit a packet to neighbor i. The initial height of a pulse is τ so the value at time t is equal to the residual transmission time of the packet, should the opportunity be used. Since some of the opportunities overlap, they cannot all be accepted.

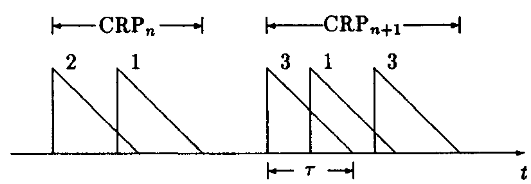


Fig. 2. Typical transmission opportunity sequence for $M = 3$ neighbors.

The scheduling policy we discuss first is essentially that used in earlier work [2], [4]. We call it the *simple* nonpreemptive scheduling policy, and denote it by π^S. Consider a given satellite during a period of time in which it is not listening for packet transmissions from other satellites. Under policy π^S the server assigns the queued packets to transmission opportunities by never passing up the opportunity to send a packet, as long as the transmission of a packet does not overlap the transmission of another packet already assigned to transmit. The throughput achievable by this simple scheduling policy is evaluated in Section II. In Section III we describe two different scheduling policies, π^P and π^T, which are optimal in maximizing the weighted saturated-system throughput to all neighbors, over certain classes of scheduling policies.

The policy π^P maximizes a reward *pathwise*. It is based on dynamic programming, and requires that the server look ahead at all transmit opportunities before deciding whether to use a given one. To obtain a simpler policy, we consider the class of nonlookahead with preemption policies, in which the server considers the transmit opportunities in sequential fashion. Upon considering each opportunity, the server can decide to tentatively use the opportunity or not. If a transmit opportunity is tentatively used, and if it would interfere with an opportunity previously tentatively accepted, then the previous one is dropped from consideration and is said to be preempted. If a transmit opportunity is tentatively used and not later preempted, then it will actually be used.

The optimal nonlookahead with preemption policy, denoted as π^T, is found to be of *threshold* type—a previously accepted opportunity will be preempted if and only if the time needed for it to be completed is above a threshold. The optimal threshold policy is also shown to be optimal within a larger class: the class of τ-lookahead policies. A *dynamic* scheduling policy, denoted by π^D, is introduced in Section IV. It dynamically adjusts the thresholds so that the packet arrival rates can vary with time and needn't be known in advance. The throughputs achievable by these policies are compared in Section V.

The fact that transmit opportunities arrive in an asynchronous fashion is a key feature of our model. If our model is mapped into a discrete time formulation, this key feature should still be preserved. In contrast, interesting recent work of Tassiulas and Ephremides [6], [7] investigates a system with time slots in which service opportunities either completely overlap or do not overlap at all. The difference between the models, and hence the results obtained, is substantial. For example, the threshold policies considered here make no sense for the model of [6], [7], while a policy of [6], [7], to accept the opportunity with the highest backlog, makes no sense for our model.

II. Performance of the Simple Nonpreemptive Policy

Let $N(t) = (N_i(t))_{1 \le i \le M}$, where $N_i(t)$ denotes the number of packets of type i in the system at time t, and let $|N(t)|$ denote the total number of packets in the system at time t. For convenience we assume that the system is empty at time 0.

Definition 2.1: The system is weakly stable if for any $\epsilon > 0$ there is a constant c such that $P[|N(t)| \le c] \ge 1 - \epsilon$ for all t. The system is exponentially stable if there are positive constants A and α so that $P[|N(t)| > c] \le A\exp(-\alpha c)$ for all $c, t > 0$.

Given (λ_i) and (μ_i) we can, except for "borderline" cases, determine whether the system is stable. Consider the two sets of inequalities, where $\lambda = \sum_{i=1}^{M} \lambda_i$:

$$\lambda_i \le \mu_i(1 - \tau\lambda) \tag{2.1}$$

$$\lambda_i < \mu_i(1 - \tau\lambda). \tag{2.2}$$

Theorem 2.1: Suppose the simple nonpreemptive policy π^S is used. a) If the system is weakly stable then (2.1) holds for $1 \le i \le M$. b) If (2.2) holds for $1 \le i \le M$ then the system is exponentially stable.

The theorem is proved in Appendix A, but a heuristic explanation for the theorem is the following. Work arrives at average rate $\lambda\tau$, so if the system is stable then the transmitter is idle a fraction $(1 - \lambda\tau)$ of the time. A type i transmit opportunity can only be used when the transmitter is idle, so the right-hand side of (2.1) is the arrival rate of usable type i transmit opportunities. The conditions thus state that the rate of arrivals should be less than the rate of usable transmission opportunities. This heuristic explanation of Theorem 2.1 readily translates into a proof of part a), but it only suggests a flawed circular argument for part b). The tact taken in Appendix A is to first show that (2.2) implies a stronger inequality if the indexes $\{1,\cdots,M\}$ are suitably permuted, and then the new concept of E-sequences is used to analyze the system averaged over appropriate time intervals.

In view of Theorem 2.1, given $\underline{\mu} = (\mu_1,\cdots,\mu_M)$, we call $\mathscr{S}^S$ the stability region for π^S, where $\mathscr{S}^S = \{\underline{\lambda}: 0 \le \lambda_i < \mu_i(1 - \lambda\tau)\}$. Another performance characteristic of the policy π^S is the *saturated-system throughput vector* for given $\underline{\mu}$. It is defined as the vector of throughputs, $\underline{\eta}^S = (\eta_1^S,\cdots,\eta_M^S)$, under the assumption that the queues at the stations are initially, and hence always, infinite. Thus, the saturated-system throughput vector $\underline{\eta}^S$ depends on $\underline{\mu}$, but not on the packet arrival rates. The concept of saturated-system throughput vector is mainly of interest as an aid in determining stability regions, a role illustrated in Section IV. Saturated-system throughput vectors are also used in Appendix A to give another representation of the stability region $\mathscr{S}^S$.

The vector $\underline{\eta}^S$ is calculated as follows. Each time the saturated system becomes idle, the transmitter must wait on average $1/\mu$ time units to begin transmitting the next packet, where $\mu = \sum_{i=1}^{M} \mu_i$. Furthermore, each transmission takes τ time units. Thus, a cycle consisting of one packet transmission and one idle period has mean length $\tau + 1/\mu$. The *sum* of the throughputs of the saturated system, called the total throughput and denoted by η_{tot}^S, is therefore given by $\eta_{tot}^S = 1/(\tau + 1/\mu) = \mu/(\mu\tau + 1)$. Finally, since a typical transmission opportunity is type i with probability μ_i/μ, the throughput of type i packets for the saturated system is given by

$$\eta_i^S = \frac{\mu_i}{\mu\tau + 1}. \tag{2.3}$$

We caution the reader that the condition (2.3) does not imply that $\lambda_i \le (\mu_i/\mu\tau + 1)$ is necessary for stability in a nonsaturated system. The reason is that, in a nonsaturated system if a type j transmission opportunity arrives while the transmitter is idle and there are no type j packets present, a subsequent arrival of a type i transmission opportunity less than τ time units later can be utilized, whereas it cannot in the saturated system. This condition is reflected in the correct stability condition (2.2).

An important property of the policy π^S is that no other policy has a larger value of η_{tot}^S. Indeed, for a saturated system, the policy π^S maximizes the number of transmitted packets during $[0, t]$ for all t with probability one.

III. Weighted Throughput Scheduling

The goal in searching for a better scheduling policy should not be to find one that achieves a saturated-system total throughput greater than η_{tot}^S—no such policy exists under our system model. Rather, policies are sought with saturated-system throughput vectors other than the single vector $\underline{\eta}^S$ corresponding to π^S. By reducing the throughput to one neighbor, the throughput to another can be increased. This leads to transmission policies with larger stability regions.

We focus in this section on a saturated system in spite of the cautionary note to the reader three paragraphs above, for the following reasons. First, if we consider a rich enough class of transmission policies (rather than just the simple nonpreemptive policy), it is *a priori* true that any vector of average throughputs achievable by a nonsaturated system can also be achieved by a saturated system —in effect, the occasional lack of packets in a nonsaturated system can be simply viewed as a control mechanism for a saturated system. On the other hand, if a vector of arrival rates is dominated by a saturated-system throughput vector and if dummy packets are sent when no real packets of a specified type are available, then the unsaturated system, under some additional mild conditions, is stable. We thus expect the region of stable arrival rate vectors (union over all policies) to coincide with the region of saturated system throughput vectors (union over all policies), except possibly for points on the boundaries of the regions.

Denote by *conflict resolution period* (CRP) a maximal interval of time completely covered by transmit opportunities. Consider the first CRP in Fig. 2. Under the simple nonpreemptive policy, the first opportunity is taken, so the type 2 packet is transmitted. Under a different policy, the server might wait and opt for the second opportunity, so the first type 1 packet is transmitted. For this particular example, the total throughput is not diminished under the other policy, but the packets transmitted have different types.

More generally, the server can break ties of any equal number of type 2 packets and type 1 packets in favor of the latter. As long as the simple nonpreemptive policy is followed for all cases not resulting in a tie, the total sum of throughputs is still η^S_{tot}. The type 1 throughput is higher than that corresponding to the simple nonpreemptive policy, and the type 2 throughput is reduced by the amount of type 1 gain. If the server employs this tie-breaking scheme at every opportunity to do so, a new achievable throughput vector (which we denote by $\eta^{2\to 1}$, where the superscript represents the reduction of type 2 throughput in favor of type 1 throughput) is obtained which still has a combined throughput equal to η^S_{tot}, but has a larger throughput of type 1 packets at the expense of a smaller throughput for type 2 packets. If at each occurrence of a tie, the server applies this tie-breaking scheme with some probability, and otherwise breaks ties using the simple nonpreemptive policy, any point on the line segment between $\underline{\eta}^S$ and $\eta^{2\to 1}$ can be achieved.

Pathwise-Max Search Policy

To bias the throughput even more towards one or more neighbors at the expense of throughput to other neighbors, a loss in total throughput must be suffered. This is a fundamental difference between the scheduling problem we address here and the problem of priority scheduling for a work-conserving queue. Associate a reward r_i with each packet transmitted by the host satellite to neighbor i. We refer to an *admissible service path* through a CRP as any sequence of nonoverlapping transmission opportunities. Denote by π^P a policy which, for each CRP, selects an admissible service path that maximizes the total reward of the path. For example, one admissible service path through the second CRP of Fig. 2 includes the two opportunities for neighbor 3, while another path includes only the opportunity for neighbor 1. Policy π^P transmits two packets to neighbor 3 if $2r_3 > r_1$, or transmits one packet to neighbor 1 if $2r_3 < r_1$. When two or more paths have the same reward, the tie must be broken. There are different versions of π^P corresponding to different tie-breaking schemes, though the case of ties becomes insignificant when the rewards have no common multiples.

By definition, when the policy π^P is employed over any number of CRP's, it maximizes the weighted throughput. Therefore, if $r_i > 0$ for each i, then no other policy can yield a vector of throughputs which strictly dominates the throughput vector achieved by π^P.

We will briefly describe how a version of the policy π^P can be based on an algorithm fashioned using dynamic programming in a way similar to that of a Viterbi decoder. The server is given the following information about a CRP at some time before the CRP begins:

N = number of transmit opportunities

t_k = arrival time of the kth opportunity, $k = 1, 2, \cdots, N$

T_k = type of the kth opportunity, $k = 1, 2, \cdots, N$.

For convenience, we also define $t_{N+1} = +\infty$. The server considers the transmit opportunities sequentially, computing an optimal path leading up to each opportunity.

Define:

$S(k)$ = the set of transmit opportunities *ending* during $(t_{k-1}, t_k]$.

$P(k)$ = an optimal path using transmit opportunities *ending* before or at time t_k.

$P'(k)$ = the path obtained by appending k to the end of $P(k)$.

The sequence $P(1), \cdots, P(N)$ can be computed recursively as follows. The initial value is the null path: $P(1) = (\ )$. For $2 \le k \le N$, the path $P(k)$ is taken to be a path in the set $\{P'(i): i \in S(k)\} \cup \{P(k-1)\}$ with the greatest reward. The output of the algorithm can be taken to be a path in the set $\{P'(i): i \in S(N)\}$ with the greatest reward.

As an example of this procedure, consider the CRP shown in Fig. 3 and suppose $(r_1, r_2, r_3) = (5, 4, 3)$. Table I indicates the outcome after each step of the algorithm. The resultant path contains a single type 1 opportunity, and two type 2 opportunities, whereas a server under the simple nonpreemptive policy elects to transmit two type 3 packets, and one type 2 packet. Similarly, a server giving total priority to type 1 packets elects to transmit both such packets in the CRP, at the expense of a reduction in total throughput.

The usual induction proof of dynamic programming implies that the algorithm described here maximizes the total reward. A pathwise-max policy π^P is thus obtained by using the above algorithm during each CRP. Given $\underline{\mu}$, let $\mathscr{S}^P$ denote the region bounded by saturated-system throughput vectors for pathwise-max policies generated by varying the reward values and tie-breaking rules. The region $\mathscr{S}^P$ contains the saturated-system throughput vector of any policy.

In practice, we envision that the system will typically not be saturated, and the arrival rates of new packets may not be known. That suggests that the reward parameters be changed as a function of packet backlogs, or as a function of some estimate of arrival rates. Since π^P requires extensive lookahead, it is not clear how the reward values should be updated, even in an adaptive fashion. The policy π^P may also be somewhat more complex than desirable. For these reasons, we next consider certain scheduling policies which do not look more than τ time units into the future. These policies achieve near optimal weighted throughput scheduling with negligible control processing.

Time-Threshold Preemption Policy

Consider a policy π^T under which type i packets are *vulnerable* to preemption by type j packets up to a threshold T_{ij}, for all ordered pairs of packet types (i, j). By this

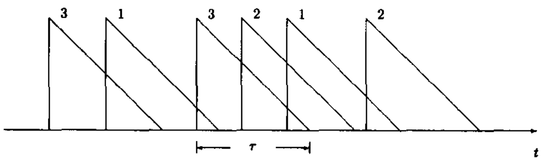


Fig. 3. Sample CRP for demonstration of the pathwise-max algorithm. Each transmit opportunity is represented by a triangular pulse of width τ, where the height at any given time is the residual service time for that packet.

TABLE I
PATHWISE-MAX ALGORITHM RESULTS FOR THE CRP OF FIG. 3. START TIMES ARE IN TERMS OF SERVICE DURATIONS, τ, AND ARE RELATIVE TO THE BEGINNING OF THE PERIOD.

Step $\#k$	t_k	T_k	$P(k)$	reward of $P(k)$
1	0.0	3	( )	0
2	0.5	1	( )	0
3	1.3	3	(1)	3
4	1.7	2	(2)	5
5	2.1	1	(2)	5
6	2.8	2	(2, 4)	9
END	—	—	(2, 4, 6)	13

we mean that if the server has previously scheduled a type i packet, if a type j transmission opportunity arrives and a type j packet is in the queue, then the arrival will preempt the type i packet if the arrival occurs within T_{ij} seconds after the start of the type i transmission. Though we say a packet is "preempted" here, it is only preempted in a logical sense—in the satellite system described in Section I such a preempted packet never even begins transmission. The class of nonlookahead policies with preemption is a subset of the class of *τ-lookahead* policies, which are those policies that require knowledge of service opportunities at most τ time units in the future. The following theorem is proved in Appendix B.

Theorem 3.1: Given arrival rates of transmission opportunities, $\underline{\mu}$, and service rewards, $\underline{r}$, there are thresholds $(T_{i,j})$ so that the threshold policy π^T maximizes the weighted saturated-system throughput over all τ-lookahead policies. Furthermore, if $r_1 \geq r_2 \geq \cdots \geq r_M$ then the thresholds can be taken to satisfy $0 \leq T_{i,j+1} \leq T_{i,j} \leq T_{i+1,j} \leq \tau$ for $i > j$ and $T_{i,j} = 0$ for $i \leq j$.

Also given in Appendix B is an expression for the saturated-system throughput vector η^T for τ^T in terms of the arrival rates of transmission opportunities and the threshold values. If the satellite has two neighbors ($M = 2$) and if the packet types are ordered so that $r_2 \geq r_1$, then we take $T_{1,2} = 0$ without loss of optimality and (7.15) reduces to

$$\eta_1^T = \frac{\mu_1 + \mu_2(1 - e^{-\mu_1 T_{2,1}})}{1 + (\mu_1 + \mu_2)\tau + \dfrac{\mu_2}{\mu_1}(1 - e^{-\mu_1 T_{2,1}}(1 + \mu_1 T_{2,1}))}$$

$$\eta_2^T = \frac{\mu_2 e^{-\mu_1 T_{2,1}}}{1 + (\mu_1 + \mu_2)\tau + \dfrac{\mu_2}{\mu_1}(1 - e^{-\mu_1 T_{2,1}}(1 + \mu_1 T_{2,1}))}. \tag{3.1}$$

IV. DYNAMIC SCHEDULING

If the packet arrival rate vector $\underline{\lambda}$ is known then we can search for a threshold matrix $\boldsymbol{T}$, such that $\underline{\lambda} < \underline{\eta}^T$. If such a $\boldsymbol{T}$ can be found, then the system with arrival rate vector $\underline{\lambda}$ is stable when the scheduling policy is π^T modified by inserting dummy packets when queues are empty in order to simulate saturated-system traffic. However, this method is not satisfactory since the parameter matrix $\boldsymbol{T}$ must be adjusted according to the input traffic vector $\underline{\lambda}$. To remedy the situation, we present a dynamic policy π^D in which $\boldsymbol{T}$ is a function only of the backlog vector $\underline{N}$. Specifically, the basic idea of the policy is that for any backlog vector $\underline{N}$, the server adjusts $\boldsymbol{T}$ in an attempt to make $\underline{\eta}^T$ proportional to $\underline{N}$.

Two Receivers ($M = 2$)

According to our basic idea, we wish to solve for the value of $\boldsymbol{T}$ such that $(\eta_2^T/\eta_1^T) = (N_2/N_1)$. Following Theorem 3.1, we take $T_{0,0} = T_{1,1} = 0$, and at most one of $T_{1,2}$ and $T_{2,1}$ should be nonzero. If it happens that $N_1/\mu_1 = N_2/\mu_2$, then we can simply take all the thresholds to be zero. If $N_1/\mu_1 > N_2/\mu_2$ then the choice of threshold matrix should favor type 1 packets, so we take $T_{1,2} = 0$ and $T_{2,1} > 0$. In this case, (3.1) yields

$$\frac{\eta_2^T}{\eta_1^T} = \frac{\mu_2 e^{-\mu_1 T_{2,1}}}{\mu_1 + \mu_2(1 - e^{-\mu_1 T_{2,1}})}.$$

Equating this with N_2/N_1 and solving for $T_{2,1}$ results in:

$$T_{2,1} = \frac{1}{\mu_1} \ln\left(\frac{1 + N_1/N_2}{1 + \mu_1/\mu_2}\right). \tag{4.1}$$

By symmetry, if $N_1/\mu_1 < N_2/\mu_2$ then we set

$$T_{1,2} = \frac{1}{\mu_2} \ln\left(\frac{1 + N_2/N_1}{1 + \mu_2/\mu_1}\right). \tag{4.2}$$

We can hard-limit these thresholds between zero and τ with no change in operation, so that (4.1) and (4.2) can both be used for any value of (N_1, N_2). When a transmit opportunity arrives for which a packet exists, the server computes these thresholds based on the current backlog. The server compares the time invested in the current transmission (if any) with the appropriate threshold to decide whether to preempt.

Define the region $\mathscr{S}^D$ for given $\underline{\mu}$ by

$$\mathscr{S}^D \triangleq \left\{\underline{\lambda}: \underline{\lambda} < \underline{\eta}^T \text{ for some } \boldsymbol{T}\right\}. \tag{4.3}$$

Theorem 4.1: If $\underline{\lambda} \in \mathscr{S}^D$ then the system is exponentially stable under π^D.

Thus, ignoring the situation when $\underline{\lambda}$ is on the upper boundary of $\mathscr{S}^D$, our dynamic policy yields a stable system whenever any nonlookahead with preemption scheduling policy (in particular, one depending on $\underline{\lambda}$) yields a stable system.

Theorem 4.1 is proved in Appendix C. The rough idea of the proof is as follows. We first consider the backlog (N_1, N_2) of the system sampled periodically, where the spacing between samples is large enough to provide sufficient averaging so we can speak of the drift of the backlog. We then use the idea, used by Mikhailov [5] for a similar problem, of expressing the backlog vector (N_1, N_2) in polar coordinates and using a Lyapunov function that is linear along rays through the origin in $\mathbb{R}_+^2$.

Under π^D, the drift when the backlog is $\underline{N}$ is roughly proportional to the vector difference $\underline{\lambda} - \underline{\eta}^T$, where $\boldsymbol{T}$ is given by (4.1) and (4.2). The thresholds depend only on the ratio of the two backlogs, so if we let $\phi = \tan^{-1} N_2/N_1$ then we can express the thresholds in terms of ϕ. The tendency of the dynamic policy is to push the angular component of the backlog ϕ to an angle at which the angular drift is zero (or to one of the coordinate axes), and at that angle we verify that the radial component of the drift is negative. The angular drift is zero when ϕ is also the angle of the arrival rate vector. Thus, roughly speaking, when the backlogs are large the ratio of the backlogs tends towards the ratio of the arrival rate vectors. Fortuitously, by Little's law, this means that π^D tends to equalize the delays experienced by type 1 and type 2 packets.

Fig. 4 indicates an example of the drift vector field of the backlog for some $\lambda \in \mathcal{S}^D$. A dotted line is drawn through the arrival rate vector $\underline{\lambda}$. Also pictured in the figure is the boundary of a scaled version of the stability region $\mathcal{S}^D$.

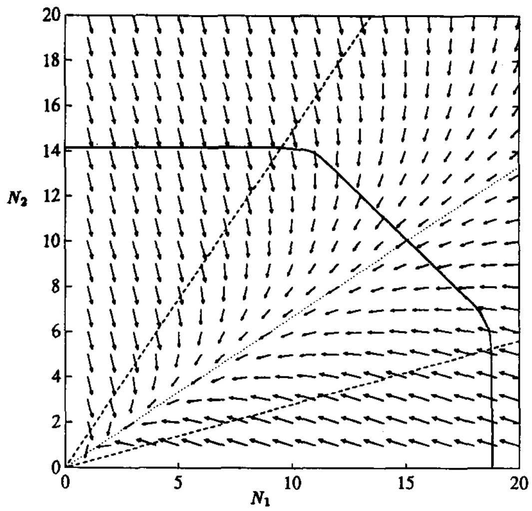


Fig. 4. Example of drift field for system under the dynamic policy for stable $\underline{\lambda}$ with $\mu_1 = 6000$, $\mu_2 = 4000$, and $\tau = 100\mu s$.

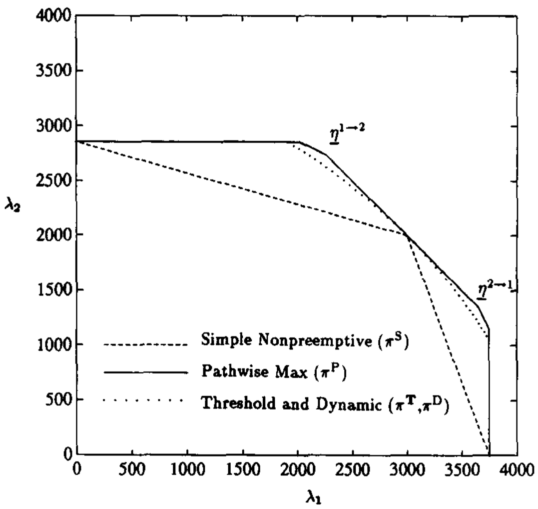


Fig. 5. Achievable throughput regions of Simple Nonpreemptive, Pathwise-Max, and Time-Threshold Preemption policies for $M = 2$, $\mu_1 = 6000$, $\mu_2 = 4000$, and $\tau = 100$ μs.

Three or More Receivers ($M \geq 3$)

A heuristic policy for $M \geq 3$, motivated in part by the policy for $M = 2$, operates as follows. If the server scheduled the transmission of a type i packet, and then a type j transmit opportunity arrives which overlaps the type i transmission, the server computes the threshold:

$$T_{i,j} = \frac{1}{\mu_j} \ln\left(\frac{1 + N_j/N_i}{1 + \mu_j/\mu_i}\right).$$

If the type i transmission opportunity precedes the type j transmission opportunity by less than $T_{i,j}$ units of time, then the server will preempt the transmission of the type i packet in favor of the type j packet; otherwise, the server will disregard the type j transmission opportunity and keep the type i packet scheduled for transmission.

Monte Carlo simulations suggest (and we conjecture) that this policy enjoys for all M the stability property of the case $M = 2$: it appears to yield a stable system whenever any nonlookahead with preemption policy does.

V. Comparison and Conclusion

Fig. 5 displays the stability regions for two neighboring satellites ($M = 2$) operating at 10 Mbps, with a packet length of 1000 bits or $\tau = 100$ μs, and opportunity arrival rates $\mu_1 = 6000$, $\mu_2 = 4000$ opportunities/s. The inner region bounded by two straight lines is the stability region $\mathcal{S}^S$ of the simple nonpreemptive scheduling policy π^S. The point where the lines intersect is $\underline{\eta}^S = (3000, 2000)$. At this point, the maximum total throughput $\eta_{tot}^S = 5000$ packets/s is obtained.

The outermost curve in Fig. 5 bounds the region $\mathcal{S}^P$, which is the region bounded by the saturated-system throughput vectors for versions of the pathwise-max policy, π^P. We see that for some values of the ratio λ_2/λ_1, policy π^P offers roughly a 10% increase in throughput over that of π^S.

No policy can achieve throughput points above the line with slope -1, passing through η^S. As mentioned in Section III, however, the points on this line between $\eta^{1\to 2}$ and $\eta^{2\to 1}$ are saturated-system throughput vectors for biased versions of π^P. These points form part of the boundary of $\mathcal{S}^P$. We found by Monte-Carlo simulation that $\eta^{1\to 2}$ and $\eta^{2\to 1}$ are approximately (2260, 2740) and (3650, 1350), respectively. Another point of interest under policy π^P is that where $r_1 > 0$, $r_2 = 0$, and all equal reward paths are broken in favor of those with the highest actual throughput. This will result in transmitting type 2 packets only when they do not interfere with the transmission of a type 1 packet. Thus, we expect a type 1 throughput equal to $\eta_1^* = 3750$ packets/s. (found by treating type 1 packet as the only packet type in the system), which is what we find by simulation. The corresponding type 2 throughput at this point is approximately 1150 packets/s. By reversing the rewards, the point (2020, 2857) is achieved. The straight line segments connecting these four points nearly encompass the region $\mathcal{S}^P$.

The stability region $\mathcal{S}^D$ for the dynamic policy π^D is seen to be just slightly smaller than $\mathcal{S}^P$. (Recall that the region $\mathcal{S}^D$ coincides with the region of saturated-system throughput vectors for threshold policies, and it is also the region of all saturated-system throughput vectors achievable by arbitrary τ-lookahead policies.) Thus the dynamic threshold policy π^D, which is nearly as easy to implement as the simple policy π^S, yields almost as large a throughput region as policies that use extensive lookahead. The example shown is typical of other comparisons we have made, indicating that the dynamic policy π^D offers a good design point on the tradeoff between complexity and throughput.

Appendix A
Stability Region for Simple Policy

Theorem 2.1 is proved in this appendix, and an alternative characterization of the stability region $\mathcal{S}^S$ is also given. First the concept of E-sequence is introduced, which is a useful complement to the methods for proving stability given in [3].

E-Sequences

A sequence $(T(k), \mathcal{F}(k))_{k\geq 0}$ is called a (D, α, δ) E-sequence (or, simply an E-sequence if the parameter values are understood) if for each $k \geq 0$

a) $T(k)$ is an $\mathcal{F}(k)$ measurable random variable,
b) $P[T(k+1) \geq T(k) + \delta] = 1$, and
c) For any $c > 0$, $P[T(k+1) - T(k) \geq c | \mathcal{F}(k)] \leq D\exp(-\alpha c)$ *a.s.*

An E-sequence should be thought of as a point process in which successive points are spaced apart by at least δ, and each interarrival time is exponentially bounded, given all past interarrival times. Note that if R is a random index, which is a stopping time relative to $(\mathcal{F}(k))_{k\geq 0}$, then property c) in the definition implies that

$$P[T(R+1) - T(R) \geq c | \mathcal{F}(R)] \leq D\exp(-\alpha c) \ a.s. \tag{6.1}$$

and therefore also that

$$P[T(R+1) - T(R) \geq c] \leq D\exp(-\alpha c). \tag{6.2}$$

Given a constant t such that $P[T(0) \leq t] = 1$, the residual lifetime at time t for the E-sequence is defined by

$$\gamma_t = T(S) - t \tag{6.3}$$

where S is the stopping time defined by $S = \min\{k: T(k) \geq t\}$.

Lemma 6.1: Let $D' = D/(1 - \exp(-\alpha\delta))$. Then $P[\gamma_t \geq c] \leq D'\exp(-\alpha c)$ for $c \geq 0$, and $E[\gamma_t] \leq D'/\alpha$.

Proof: For $l \geq 0$, define $R_l = \min\{k: T(k) \geq t - (l+1)\delta\}$. Then

$$P[\gamma_t \geq c] = \sum_{l=0}^{\infty} P[\{\gamma_t \geq c\} \cap \{T(S-1) \in [t - (l+1)\delta, t - l\delta)\}] \tag{6.4}$$

$$\leq \sum_{l=0}^{\infty} P[T(1+R_l) - T(R_l) \geq l\delta + c] \tag{6.5}$$

$$\leq \sum_{l=0}^{\infty} D\exp(-\alpha(l\delta + c)) = D'\exp(-\alpha c). \tag{6.6}$$

This establishes the first conclusion of the lemma, which in turn implies the second. □

In applications of the lemma it will be important that D' does not depend on t. The lemma still holds when t is replaced by a random time of the form $T(0) + L$, where L is a positive constant (simply condition on the value of $T(0)$ for a proof).

Given a (D, α, δ) E-sequence $(T(k), \mathcal{F}(k))_{k\geq 0}$, we define a new sequence $(T^*(k), \mathcal{F}^*(k))_{k\geq 0}$ by "thinning" the given E-sequence so that successive variables exceed the previous ones by at least L each. To be precise, first inductively define $(S_k)_{k\geq 0}$ by $S_0 = 0$ and $S_{k+1} = \min\{j: T(j) \geq T(S_k) + L\}$, and then set $T^*(k) = T(S_k)$ and $\mathcal{F}^*(k) = \mathcal{F}(S_k)$. By induction on k and use of the previous lemma, we obtain the following lemma.

Lemma 6.2: The thinned sequence is a (D', α, L) E-sequence, where $D' = D/(1 - \exp(-\alpha\delta))$.

In applications of the lemma it will be important that the constant D' does not depend on L.

Proof of Theorem 2.1: Begin the proof by proving part a), so assume that the system is weakly stable. Given $\epsilon > 0$ arbitrarily small, there are constants c and T so that (6.7)–(6.11) below are true. First, the definition of weak stability and the central limit theorem yield

$$P[|N(T)| \le c] \ge 1 - \epsilon \tag{6.7}$$

$$P\left[(\text{number of type } i \text{ arrivals in } [0,T]) \ge \lambda_i T - c\sqrt{T}\right] \ge 1 - \epsilon. \tag{6.8}$$

Since $|N(0)| = 0$, the number of departures in $[0,T]$ is the number of arrivals minus $|N(T)|$, and each customer takes τ time units to serve. Thus, from (6.7) and (6.8) we conclude that

$$P[\text{at least } \lambda T - c(M\sqrt{T} + 1) \text{ departures in } [0,T]] \ge 1 - (M+1)\epsilon \tag{6.9}$$

$$P[\text{idle time in } [0,T] \text{ is at most } T - \tau(\lambda T - c(M\sqrt{T} + 1))] \ge 1 - (M+1)\epsilon \tag{6.10}$$

$$P\left[\text{at least } \lambda_i T - c(1 + \sqrt{T}) \text{ type } i \text{ departures in } [0,T]\right] \ge 1 - 2\epsilon. \tag{6.11}$$

Using elementary properties of Poisson processes, it is not difficult to show that the cumulative number of type i transmission opportunities which find the server idle, as a function of the cumulative server idle time, is a Poisson point process with rate μ_i. The strong law of large numbers and the independent increment property of Poisson processes thus implies that there exits a constant C, depending only on μ_i and ϵ, such that with probability at least $1 - \epsilon$ the following is true *at all times*: the cumulative number of type i transmission opportunities to arrive when the server is idle is less than C, plus $(\mu_i + \epsilon)$ times the cumulative amount of idle time. Combining this observation with (6.10) yields the following inequality:

$$P\left[\text{more than } C + (\mu_i + \epsilon)[T(1 - \tau\lambda) + c\tau(M\sqrt{T} + 1)] \text{ type } i \text{ service opportunities in } [0,T]\right] \le (M+2)\epsilon. \tag{6.12}$$

Comparing (6.11) and (6.12) for $\epsilon < 1/(M+4)$ and observing that there must be at least as many type i service opportunities as type i departures yields part a) of the theorem.

Turning to the proof of part b) of Theorem 2.1, assume that (2.2) holds for $i \in \{1, \cdots, M\}$. Without loss of generality (since we can re-index the packet types if necessary) assume that for the remainder of this proof that the following is true:

$$\frac{\lambda_1}{\mu_1} \le \frac{\lambda_2}{\mu_2} \le \cdots \le \frac{\lambda_M}{\mu_M}. \tag{6.13}$$

Starting with (2.2) we have

$$\begin{aligned}\lambda_i &< \mu_i - \mu_i\tau(\lambda_1 + \cdots + \lambda_{i-1}) - \tau(\mu_i\lambda_i + \cdots + \mu_i\lambda_M) \\ &< \mu_i - \mu_i\tau(\lambda_1 + \cdots + \lambda_{i-1}) - \tau(\lambda_i\mu_i + \cdots + \lambda_i\mu_M)\end{aligned}$$

where the second inequality follows from the reordering, i.e., $\mu_i\lambda_j \le \lambda_i\mu_j$ for $j \ge i$. After rearranging terms, we have the following:

$$\lambda_i < \frac{\mu_i}{1 + \tau(\mu_i + \cdots + \mu_M)}[1 - \tau(\lambda_1 + \cdots + \lambda_{i-1})]. \tag{6.14}$$

We will next show that the following statement, S_i, is true for $1 \le i \le M$.

Statement S_i: There exist constants $a_1, \cdots, a_i$ and an E-sequence $(T(k), \mathscr{F}(k))_{k \ge 0}$ with $T(0) = 0$ such that

$$N_j(T(k)) \le a_j \quad \text{for} \quad 1 \le j \le i \quad \text{and} \quad k \ge 0, \tag{6.15}$$

and $\mathscr{F}(k) = \mathscr{F}_S(T(k))$, where $\mathscr{F}_S(t)$ is the σ-algebra generated by the system up to time t.

Suppose $i \ge 1$ is fixed. For the sake of argument by induction, if $i \ge 2$ suppose that S_{i-1} is true. We shall establish that S_i is true. If $i = 1$ let $T(k) = k$, whereas if $i \ge 2$ let $(T(k))_{k \ge 0}$ denote the sequence associated with statement S_{i-1}. Let L be a constant (to be specified later) and let $(T^*(k), \mathscr{F}^*(k))_{k \ge 0}$ be the new E-sequence obtain by thinning $(T(k), \mathscr{F}(k))_{k \ge 0}$ so that successive variables exceed the previous ones by at least L each. We show next that if L and a_i are sufficiently large, then the random process $(N_i(T^*(k)), \mathscr{F}^*(k))$ satisfies [3, conditions C.1 and C.2] (for some ϵ_0, $a = a_{i+1}$ and Z).

[3, condition C.1] is a drift condition. In the notation here, it states that for fixed k,

$$E[N_i(T^*(k+1)) - N_i(T^*(k)) | \mathscr{F}^*(k)] \le -\epsilon_0 \tag{6.16}$$

on the event $\{N_i(T^*(k)) > a_i\}$. The difference $N_i(T^*(k+1)) - N_i(T^*(k))$ is upper bounded by the number of type i arrivals during the full interval $[T^*(k), T^*(k+1)]$ minus the number of type i departures during the shorter interval $[T^*(k), T^*(k) + L]$. We consider the two terms in turn. By Lemma 6.1, the conditional length of the full interval is less than $L + o(1)$, so that the expected number of type i arrivals during the interval is less than $\lambda_i L + o(1)$. We now turn to the other term. Since the number of packets of lower types, i.e., of types $1, 2, \cdots$, or $i - 1$, in the system at time $T^*(k)$ is bounded above by the constant $a_1 + a_2 + \cdots + a_{i-1}$, the expected amount of time during the period $[T^*(k), T^*(k) + L]$ that the server is occupied with packets of lower types is at most $L(\lambda_1 + \cdots + \lambda_{i-1})\tau + o(1)$. When the server is not serving packets of lower types it serves packets of type i at a rate of at least $\mu_i/(1 + \tau(\mu_i + \cdots + \mu_M))$, as long as there are type i packets in the system. By choosing a_i sufficiently large (depending on L), the effect of running out of packets of type i can be controlled on the event $\{N_i(T^*(k)) > a_{i+1}\}$. The net result is that, given $\{N_i(T^*(k)) > a_{i+1}\}$, the mean number of type i departures during the interval $[T^*(k), T^*(k) + L]$ is at least

$$\frac{L(1 - (\lambda_1 + \cdots + \lambda_{i-1})\tau)\mu_i}{1 + (\mu_i + \cdots + \mu_M)\tau} + o(1). \tag{6.17}$$

Since by (6.14) this exceeds $\lambda_i L + o(1)$ for large enough L, Condition C.1 indeed holds.

[3, condition C.2] is the requirement that the increments of the process $(N_i(T^*(k)), \mathscr{F}^*(k))$ be uniformly conditionally exponentially bounded. Since the length of the interval $[T^*(k), T^*(k+1)]$ is uniformly exponentially bounded, this condition is easily seen to be true.

Since [3, conditions C.1 and C.2] are satisfied, Theorem 2.3 there applies yielding that $(T^{**}(k), \mathcal{F}^{**}(k))_{k \geq 0}$ is an E-sequence, where $T^{**}(0) = 0$, $T^{**}(k) = T^*(R(k))$, and $\mathcal{F}^{**}(k) = \mathcal{F}^*(R(k))$, with

$$R(k) = \min\{l: l > R(k-1) \quad \text{and} \quad N_i(T^*(l)) < a_i\}. \tag{6.18}$$

Therefore statement S_i is true.

By the induction argument we therefore have that S_M is true. Let now $(T(k), \mathcal{F}(k))_{k \geq 0}$ denote the E-sequence for statement S_M and let γ_t denote the residual lifetime sequence of the process. Since at most one departure can occur per τ time units,

$$|N(t)| \leq \gamma_t/\tau + 1 + a_1 + \cdots + a_M \tag{6.19}$$

for any t. By Lemma 6.1 it follows that the system is exponentially stable. Theorem 2.1 is proved.

Structure of Stability Region for Simple Policy

A different representation of the stability region $\mathcal{S}^S$ for the simple algorithm π^S will be given in this section. Let $\bar{\mathcal{S}}^S$ denote the closure of the region $\mathcal{S}^S$. Thus, $\bar{\mathcal{S}}^S = \{\underline{\lambda}: 0 \leq \lambda_i \leq \mu_i(1 - \lambda\tau)\}$ where $\lambda = \lambda_1 + \cdots + \lambda_M$. We will prove that $\bar{\mathcal{S}}^S$ is the convex hull of the 2^M saturated throughput points that result from considering subsets of flows. That is, let $Q \subset \{1, \cdots, M\}$ represent a subset of the packet types of the original system. If the server only transmits packets of those types in Q, then the saturated-system throughput vector of the new system is

$$\eta_i^Q = \frac{\mu_i I_{\{i \in Q\}}}{1 + \tau \sum_{i \in Q} \mu_i}.$$

Theorem 6.1: The set $\bar{\mathcal{S}}^S$ is the convex hull of $\{\eta_i^Q: Q \subset \{1, \cdots, M\}\}$.

Proof: First, note that each of the 2^M vectors $\underline{\eta}^Q$ is in $\bar{\mathcal{S}}^S$. It thus remains to show that any point in $\bar{\mathcal{S}}^S$ can be represented as a convex combination of vectors of the form η^Q. So suppose $\underline{\lambda} \in \bar{\mathcal{S}}^S$. Then,

$$\begin{aligned}\underline{\lambda} &= \alpha\underline{\eta}^Q + \underline{\lambda} - \alpha\underline{\eta}^Q \\ &= \alpha\underline{\eta}^Q + (1-\alpha)\frac{\underline{\lambda} - \alpha\underline{\eta}^Q}{1-\alpha} \\ &= \alpha\underline{\eta}^Q + (1-\alpha)\tilde{\underline{\lambda}}\end{aligned}$$

where we define $\tilde{\underline{\lambda}} = (\underline{\lambda} - \alpha\eta^Q)/(1-\alpha)$.

Let $Q = \{i: \lambda_i > 0\}$, and $\alpha = \min_{j \in Q}(\lambda_j/\mu_j)(1 + \tau\sum_{i \in Q}\mu_i)$. Note that $0 < \alpha \leq 1$ by our definition of Q and $\bar{\mathcal{S}}^S$, and that $\tilde{\lambda}_i = 0$ if $i \notin Q$ or $(\lambda_i/\mu_i) = \min_{j \in Q}(\lambda_j/\mu_j)$. It is not difficult to verify that $\tilde{\underline{\lambda}} \in \bar{\mathcal{S}}^S$.

Therefore, any $\underline{\lambda}$ in the region defined by $\bar{\mathcal{S}}^S$ can be expressed as a convex combination of one of the 2^M subsystem saturated-system throughput vectors and some other point in $\bar{\mathcal{S}}^S$ with fewer nonzero entries, which implies that $\underline{\lambda}$ is a convex combination of a set of at most $M+1$ vectors of the form $\underline{\eta}^Q$. □

Appendix B
Threshold Policies

Proof of Theorem 3.1—Optimality of Threshold Policies: The main step in the following proof of Theorem 3.1 is to show that threshold policies are optimal within the class of nonlookahead with preemption policies for maximizing a discounted infinite horizon average reward. We then outline how the result can be strengthened to show optimality within the larger class of τ-lookahead policies, and how to extend the result to the long run average reward criteria (weighted saturated-system throughput). The last part of this appendix gives expressions for computing the saturated-system throughput vector $\underline{\eta}^T$ for a threshold policy.

Define the state space $\Sigma = \{1, \cdots, M\} \times (0, \tau] \cup \{\Delta\}$. We take the topology of Σ to be the product topology of $\{1, 2, \cdots, M\} \times (0, \tau]$, with Δ added as an isolated point. We will represent the state of the server at any time t by X_t, where $X_t \in \Sigma$. The state (i, δ) denotes that a type i packet is being transmitted, and that the residual transmission time for the packet is δ. The state Δ denotes that no packet is being transmitted. For notational convenience, we will identify all pairs of the form (i, δ) for $\delta \geq 0$ with the isolated point Δ, although such pairs do not exist in the state space Σ. Hence, at any time t, X_t has the form $X_t = (i, \delta)$ where $\delta < 0$ means that no packet is being transmitted.

A (nonlookahead with preemption) policy π consists of a sequence $(\pi_1, \pi_2, \cdots)$ where $\pi_k: \Sigma \times \{1, 2, \cdots, M\} \mapsto \{0, 1\}$ is a Borel measurable function for each k. Given an initial state x_0, and a policy π, we construct a controlled process as follows. Let $(T_k, \theta_k)_{k \geq 1}$ be a marked Poisson process with total rate $\mu = \mu_1 + \cdots + \mu_M$, mark times $(T_k)_{k \geq 1}$, and marks $(\theta_k)_{k \geq 1}$ where $\Pr[\theta_k = 1] = \mu_i/\mu$.

Set

$$X_t = x_0 - (0, t) \qquad \text{for } t \in [0, T_1)$$

and for $t \in [T_k, T_{k+1})$, $k \geq 1$, set

$$X_t = \begin{cases} X_{T_k^-} - (0, t - T_k) & \text{if } \pi_k(X_{T_k^-}, \theta_k) = 0 \\ (\theta_k, \tau) - (0, t - T_k) & \text{if } \pi_k(X_{T_k^-}, \theta_k) = 1. \end{cases}$$

Thus, given π and an arbitrary initial state $x_0 \in \Sigma$, we have constructed a Σ-valued random process $X = (X_t)_{t \geq 0}$, the sample paths of which are right continuous with left-hand limits.

Let $\alpha > 0$ be a discount rate, and recall that r_i is the reward for completion of a type i transmission. The total discounted reward (over an infinite-horizon) is then defined by

$$\Phi(X) = \sum_{t \geq 0} \sum_{i=1}^{M} e^{-\alpha t} r_i I_{\{X_t = (i, 0)\}}.$$

Define the maximum expected discounted reward for initial state x_0 by

$$V^*(x_0) = \sup_{\pi} E_{x_0}^{\pi}[\Phi(X)].$$

The dynamic programming operator H is defined as follows. Let $C_+(\Sigma)$ denote the space of nonnegative, bounded, continuous functions on Σ. For $f \in C_+(\Sigma)$, define Hf by

$$Hf(i, \delta) = \int_0^{\delta} e^{-(\alpha+\mu)s} \left\{ \sum_{j=1}^{M} \mu_j [f(i, \delta - s) \vee f(j, \tau)] \right\} ds + e^{-(\mu+\alpha)\delta} \left[r_i + \frac{1}{\alpha + \mu} \sum_{j=1}^{M} \mu_j f(j, \tau) \right]$$

$$Hf(\Delta) = \frac{1}{\alpha + \mu} \sum_{j=1}^{M} \mu_j f(j, \tau).$$

The operator H maps an expected reward function at the time of the next state transition into a present expected reward function by considering all possible state transitions, along with the probability of occurrence of each possible transition. The integral term corresponds to the discounted reward at the time of the next transmit opportunity, provided the opportunity arrives before the current transmission is complete. The server will switch over if the reward for doing so is higher than that for staying the course. We then average over all possible state transitions and opportunity arrival instances, discounting all future values appropriately. The terms in square brackets account for the reward for the current packet if no service opportunities arrive until after the current packet is transmitted, plus the expected reward for beginning another transmission.

It is not difficult to verify that $Hf \in C_+(\Sigma)$, and that H has the contraction property:

$$\|Hf - Hg\| \le \frac{\mu}{\alpha + \mu} \|f - g\|$$

where $\|\cdot\|$ is the supremum norm on $C_+(\Sigma)$. By the Contraction Mapping Fixed-Point Theorem [1, Section 5.3], there exists a unique function $V^\infty \in C_+(\Sigma)$ such that $V^\infty = HV^\infty$, and

$$\lim_{n \to \infty} \|V^\infty - H^n V_0\| = 0 \text{ for any } V_0 \in C_+(\Sigma).$$

Let π^* be the policy defined in terms of V^∞:

$$\pi^*[(i, \delta), j] = \begin{cases} 0 & \text{if } V^\infty(i, \delta) \ge V^\infty(j, \tau) \\ 1 & \text{otherwise} \end{cases}$$

and let π' be an arbitrary policy. In the following, the term "step" refers to an interarrival period of the marked Poisson process $(T_k, \theta_k)_{k \ge 0}$. Suppose our server starts in some initial state x_0, and proceeds for n steps under policy π^*. At the end of the nth step, the server may be given a *stopping* reward, equal to V^∞ evaluated at the server's final state. Then,

$$\begin{aligned} &E^{\pi^*}(\text{reward for } n \text{ steps} + \text{no stopping reward}) \\ &\quad \ge E^{\pi^*}(\text{reward for } n \text{ steps} + \text{stopping reward } V^\infty) - \left(\frac{\mu}{\alpha + \mu}\right)^n \|V^\infty\| \\ &\quad \ge E^{\pi'}(\text{reward for } n \text{ steps} + \text{stopping reward } V^\infty) - \left(\frac{\mu}{\alpha + \mu}\right)^n \|V^\infty\| \\ &\quad \ge E^{\pi'}(\text{reward for } n \text{ steps} + \text{no stopping reward}) - \left(\frac{\mu}{\alpha + \mu}\right)^n \|V^\infty\|. \end{aligned}$$

The first inequality follows from the fact that the expected stopping reward for an arbitrary final state X_{T_n} cannot be greater than the discounted maximum stopping reward. The second inequality is based on induction on the number of steps. From the dynamic programming equations, we know that if the server can only proceed for one step, any arbitrary policy will achieve an expected reward less than or equal to that of π^*. Also, for general n, the server maximizes the one-step expected reward by adhering to π^*. Thus, an arbitrary policy π' cannot do any better than π^* in maximizing the expected n-step reward.

Since the rewards are nonnegative, and the total discounted reward is bounded, we can let $n \to \infty$ to obtain:

$$E^{\pi^*}[\Phi(X)] \ge E^{\pi'}[\Phi(X)].$$

Hence, π^* is optimal. V^∞ is the maximum expected reward the server can receive after proceeding for an infinite number of steps, for a given initial state. Therefore,

$$\begin{aligned} V^\infty &\ge E^{\pi^*}(\text{reward for } n \text{ steps} + \text{no stopping reward}) \\ &\ge E^{\pi^*}(\text{reward for } n \text{ steps} + \text{stopping reward } V^\infty) - \left(\frac{\mu}{\alpha + \mu}\right)^n \|V^\infty\|. \end{aligned}$$

Note that by definition, E^{π^*} (reward after n steps + stopping reward V^∞) is simply V^∞. As we let $n \to \infty$ the above expression reduces to

$$V^\infty \ge E^{\pi^*}[\Phi(X)] \ge V^\infty$$

but since π^* is optimal, $E^{\pi^*}[\Phi(X)] = V^*$ by definition. Therefore, $V^\infty \ge V^* \ge V^\infty$ so that $V^\infty = V^*$.

Hence, $V^* \in C^+(\Sigma)$, $V^* = HV^*$, and the optimal stationary policy is given by $\underline{\pi}^* = (\pi^*, \pi^*, \cdots)$, where

$$\pi^*[(i, \delta), j] = \begin{cases} 0 & \text{if } V^*(i, \delta) \ge V^*(j, \tau) \\ 1 & \text{otherwise.} \end{cases}$$

To better characterize the optimal policy, we will prove that the reward $V^*(i, \delta)$ decreases with the amount of residual work δ of the initial state and increases with the reward offered for the initial packet type. Let $\mathscr{M}$ denote the subset of $C^+(\Sigma)$ such that

$$f(\Delta) + r_i \ge f(i, \delta') \ge f(i, \delta) \ge f(\Delta) \quad \text{whenever } 0 < \delta' < \delta \le \tau \tag{7.1}$$

$$f(i, \delta) \ge f(i', \delta) \text{ if } r_i > r_{i'} \text{ and } 0 < \delta \le \tau. \tag{7.2}$$

Claim 7.1: If $f \in \mathcal{M}$ then $Hf \in \mathcal{M}$.

Proof: Suppose $f \in \mathcal{M}$ and $0 < \delta' < \delta \le \tau$. Then

$$Hf(i,\delta) \ge \int_0^\delta e^{-(\alpha+\mu)s} \sum_{j=1}^M \mu_j f(j,\tau)\, ds + e^{-\delta(\alpha+\mu)}\left\{ r_i + \frac{1}{\alpha+\mu}\sum_{j=1}^M \mu_j f(j,\tau)\right\} \tag{7.3}$$

$$= e^{-\delta(\alpha+\mu)} r_i + \frac{1}{\alpha+\mu}\sum_{j=1}^M \mu_j f(j,\tau) \tag{7.4}$$

$$\ge Hf(\Delta). \tag{7.5}$$

Since $f(i, \delta - s) \le r_i + f(\Delta) \le r_i + f(j,\tau)$, we have $Hf(i,\delta) \le x_1 + x_2 + x_3$ where

$$x_1 = \int_0^{\delta'} e^{-(\alpha+\mu)s} \sum_{j=1}^M \mu_j [f(j, \delta - s) \vee f(j,\tau)]\, dx \tag{7.6}$$

$$\le \int_0^{\delta'} e^{-(\alpha+\mu)s} \sum_{j=1}^M \mu_j [f(j, \delta' - s) \vee f(j,\tau)]\, ds \tag{7.7}$$

$$x_2 = \int_{\delta'}^{\delta} e^{-(\alpha+\mu)s} \sum_{j=1}^M \mu_j [r_i + f(j,\tau)]\, ds \tag{7.8}$$

$$= \frac{\mu}{\alpha+\mu}[e^{-(\alpha+\mu)\delta'} - e^{-(\alpha+\mu)\delta}] r_i + \frac{1}{\alpha+\mu}[e^{-(\alpha+\mu)\delta'} - e^{-(\alpha+\mu)\delta}] \sum_{j=1}^M \mu_j f(j,\tau) \tag{7.9}$$

$$x_3 = e^{-\delta(\alpha+\mu)}\left\{ r_i + \frac{1}{\alpha+\mu}\sum_{j=1}^M \mu_j f(j,\tau)\right\}. \tag{7.10}$$

Combining these bounds yields $Hf(i,\delta) \le Hf(i,\delta')$. In particular, $Hf(i,\delta) \le \lim_{\delta' \to 0} Hf(i,\delta') = r_i + Hf(\Delta)$. Finally, if $r_i > r_{i'}$ then $f(i, \delta - s) > f(i', \delta - s)$ for $0 < s < \delta$, so that $Hf(i,\delta) \ge Hf(i',\delta)$. We have established that $Hf \in \mathcal{M}$. □

Claim 7.2: $V^* \in \mathcal{M}$.

Proof: Let $V^0 \equiv 0$. Then clearly $V^0 \in \mathcal{M}$, and by induction, $H^n V^0 \in \mathcal{M}$. Since $\|H^n V^0 - V^*\| \to 0$ as $n \to \infty$, and since $\mathcal{M}$ is a closed subset of $C^+(\Sigma)$, it follows that $V^* \in \mathcal{M}$. □

Now, suppose that the packet types are ordered such that $r_1 \ge r_2 \ge \cdots \ge r_M$. Then the monotonicity properties of V^* imply that π^* can also be represented as

$$\pi^*[(i,\delta), j] = I_{\{\delta \ge T_{i,j}\}}$$

where $(T_{i,j})$ are thresholds given by

$$T_{i,j} = \begin{cases} \max\{\delta : V^*(i,\delta) \ge V^*(j,\tau)\} & i > j \\ 0 & i \le j. \end{cases}$$

Note that $T_{i,j}$ is decreasing in i and increasing in j, i.e., $0 \le T_{i,j+1} \le T_{i,j} \le T_{i+1,j} \le \tau$, which also follows from the monotonicity properties of V^*.

We have thus shown that there exists a threshold policy leading to maximum discounted reward, among all nonlookahead with preemption policies, and that the associated value function V^* has the monotonicity properties (7.1), (7.2), which lead to the ordering properties of the thresholds.

The next step in our proof of Theorem 3.1 is to show that the threshold policy is optimal (for the discounted reward problem) within the larger class of τ-lookahead policies. A τ-lookahead policy determines whether to accept a given transmission opportunity after observing the opportunity arrival process through the end of the given opportunity. There are dynamic programming equations which characterize optimal cost-to-go functions. A natural state for a τ-lookahead policy at a given time t has the form $\sigma = ((i_1, \delta_1), \cdots, (i_k, \delta_k))$ where $k \ge 0$ is a variable integer. The entry (i_j, δ_j) for $1 \le j \le k$ indicates that there is a service opportunity of type i that will end at time $t + \delta_i$. If $k = 0$ or $k = 1$ then such a state is also in the statespace Σ considered for nonlookahead preemptive policies ($k = 0$ corresponds to state Δ). By verifying the new dynamic programming equations it can be shown that the value of σ is given in terms of the values of V^* already found for states in Σ as follows:

$$V^*(\sigma) = \max\{V^*(\delta_1, i_1), \cdots, V^*(\delta_k, i_k)\}. \tag{7.11}$$

A key fact needed to verify the new dynamic equations is the following property of V^* on Σ (which can be proved by a sample path method): for any $\delta, \delta' > 0, i$ and j, the difference $V^*(i, \delta + x) - V^*(j, \delta' + x)$ has the same sign (i.e., it is positive, zero, or negative) for all $x \ge 0$. That is, whether a type i opportunity is preferred to a type j opportunity at a given time is a function only of i, j and the difference of their arrival times, so that the decision can be made as soon as both arrival times are known. Roughly speaking, there is no loss in sequentially eliminating transmission opportunities as soon as better opportunities arrive-showing that the threshold policy is indeed optimal within the class of τ-lookahead policies.

The final step in the proof of Theorem 3.1 is to show that the optimality of threshold policies for the infinite horizon discounted reward implies there is an optimal threshold policy for the long-run average reward criteria. As shown above, there are optimal threshold values for each given discount factor α. As α tends to zero there exists a convergent subsequence of the vector of thresholds, since the thresholds all take values in the compact set $[0, \tau]$. A useful technical point is that each time there is a gap of length at least τ between successive arrivals of transmission opportunities, the state Δ is reached. Moreover, the times between such gaps are exponentially bounded, and the maximum rate of reward per unit time is bounded. Consequently, for any bounded memory policy and uniformly over initial states, $\lim_{\alpha \to 0} \alpha \times$ (discounted expected reward) is equal to the long run average reward. These facts make it easy to show that the policy with the limiting threshold values is optimal for the long run average reward problem. Theorem 3.1 is proved.

Saturated-System Throughput Vector for Threshold Policies

For a given vector of transmission opportunity arrival rates $\underline{\mu}$ and matrix of thresholds $\boldsymbol{T}$, we will determine the saturation throughput vector $\underline{\eta}^T$ by examining a Markov chain embedded in the saturated-system model. The state space of the chain is $\{0,\cdots,M\}$, where state 0 denotes that no packet is being transmitted and i represents that a packet of type i is being transmitted. The embedded chain is obtained by sampling the original system when a packet is preempted or when the transmission of a packet is completed without preemption.

Define $\delta_i(t)$ as the instantaneous rate that a type i packet is preempted t seconds after the start of its transmission. This rate is simply the sum of the opportunity arrival rates of all packet types that can preempt a type i packet at t seconds, which is expressed as

$$\delta_i(t) = \sum_{j=1}^{M} \mu_j I_{\{T_{ij} > t\}}.$$

The probability that a type i packet transmission lasts at least t seconds for $t \in [0, \tau]$ is given by

$$r_i(t) = \exp\left(-\int_0^t \delta_i(s)\, ds\right).$$

For the embedded Markov chain to jump from state i to state j, a type j transmit opportunity must arrive within T_{ij} seconds of the start of the type i packet transmission, with no other preempting transmit opportunities arriving before the type j opportunity. Thus, we can express the one-step transition probabilities for the jump chain as follows:

$$\begin{aligned} P_{ij} &= \lim_{\epsilon \to 0} \sum_{k=1}^{\tau/\epsilon} \epsilon \mu_j I_{(k\epsilon < T_{ij})} r_i(k\epsilon) \\ &= \int_0^{T_{ij}} \mu_j r_i(t)\, dt \qquad i, j \geq 1. \end{aligned}$$

Jumps into the idle state occur only when a packet is transmitted without preemption, and jumps out of the idle state occur at the first transmit opportunity. Hence, we can complete the state transition matrix with the following:

$$\begin{aligned} P_{i,0} &= \Pr\,[\text{type } i \text{ not preempted}] \\ &= r_i(\tau) \end{aligned}$$

and

$$P_{0,i} = \frac{\mu_i}{\mu} \qquad i = 1, 2, \cdots, M$$

where we recall that μ is the sum of the transmission opportunity arrival rates.

Using the formula $E[X] = \int_0^\infty P[X > t]\, dt$, which is valid for any nonnegative random variable X, we see that the mean holding time of state i is given by:

$$h_i = \int_0^\tau r_i(t)\, dt. \tag{7.12}$$

We next find the equilibrium distribution of the embedded chain, $\underline{\pi}^J$, in order to compute the saturation throughput of this policy. To do so, we will solve the equation $\underline{\pi}^J = \underline{\pi}^J \boldsymbol{P}$. Let $\boldsymbol{P}$ and $\underline{\pi}^J$ be partitioned as follows, where $\underline{\alpha}$, $\underline{\beta}$, $\boldsymbol{C}$, and $\hat{\underline{\pi}}$ are implicitly defined.

$$\boldsymbol{P} = \left(\begin{array}{c|cccc} 0 & P_{0,1} & \cdots & P_{0,M} \\ \hline P_{1,0} & P_{1,1} & \cdots & P_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M,0} & P_{M,1} & \cdots & P_{M,M} \end{array}\right) = \left(\begin{array}{c|c} 0 & \underline{\alpha} \\ \hline \underline{\beta} & \boldsymbol{C} \end{array}\right) \tag{7.13}$$

$$\underline{\pi}^J = \left(\pi_0^J \;\middle|\; \pi_1^J \quad \pi_2^J \quad \cdots \quad \pi_M^J\right) = \left(\pi_0^J \;\middle|\; \hat{\underline{\pi}}\right). \tag{7.14}$$

With this partitioning, the vector equation, $\underline{\pi}^J = \underline{\pi}^J \boldsymbol{P}$, yields the scalar equation $\pi_0^J = \hat{\underline{\pi}}\underline{\beta}$, and the vector equation:

$$\hat{\underline{\pi}} = \pi_0^J \underline{\alpha} + \hat{\underline{\pi}} \boldsymbol{C}.$$

Solving this latter equation for $\hat{\underline{\pi}}$ yields

$$\hat{\underline{\pi}} = \pi_0^J \underline{\alpha}(\boldsymbol{I} - \boldsymbol{C})^{-1}$$

where $\boldsymbol{I}$ is an $M \times M$ identity matrix. The solution for $\underline{\pi}^J$ is then

$$\underline{\pi}^J = \gamma\left(1 \;\middle|\; \underline{\alpha}(\boldsymbol{I} - \boldsymbol{C})^{-1}\right)$$

where γ is the normalizing constant such that $\underline{\pi}^J$ is a valid probability distribution.

The throughput of type i packets is simply the fraction of transitions in the embedded Markov chain that are of the form $i \to 0$, divided by the mean time between jumps in the chain. The saturation throughput can thus be represented by

$$\underline{\eta}^T = \frac{\underline{\mu}(\boldsymbol{I} - \boldsymbol{C})^{-1} \operatorname{diag}[\underline{h}]}{1 + \underline{\mu}(\boldsymbol{I} - \boldsymbol{C})^{-1} \underline{h}} \tag{7.15}$$

where $\underline{h} = (h_1, \cdots, h_M)'$, and $\operatorname{diag}[\underline{h}]$ is the $M \times M$ diagonal matrix whose elements along the diagonal are the elements of the vector $\underline{h}$.

Equation (7.15) is our final expression for $\underline{\eta}^T$. Next we impose the assumption that $0 \leq T_{i,j+1} \leq T_{i,j} \leq T_{i+1,j} \leq \tau$ for $i > j$ and $T_{i,j} = 0$ for $i \leq j$, which allows us to simplify the expressions for $\boldsymbol{C}$ and $\underline{h}$. We introduce a vector $\underline{u}$ and matrix $\boldsymbol{A}$, whose elements are given by

$$u_i = \sum_{k=1}^{i} \mu_k$$

$$a_{ij} = \sum_{k=j}^{i-1} \mu_k T_{ik}.$$

The function $\delta_i(t)$ is a right-continuous, piecewise constant, decreasing function of t. Hence, the integral of $\delta_i(t)$ is a piecewise linear, continuous function of t. The first downward jump of $\delta_i(t)$ occurs at $T_{i,i-1}$, the second at $T_{i,i-2}$, and so on. In between these jumps, $r_i(t)$ is a continuous exponential function of t, with constant rate.

Therefore, we have

$$\delta_i(t) = u_{i-1} - \sum_{j=1}^{i-1} \mu_j I_{\{t \geq T_{ij}\}}$$

and

$$r_i(t) = \exp\left(-u_{i-1}t + \sum_{j=1}^{i-1} \mu_j (t - T_{ij}) I_{\{t \geq T_{ij}\}}\right).$$

The jump chain transition probability for jumps from state i to state j for $i > j \geq 1$ is then determined as follows:

$$\begin{aligned} C_{i,j} = P_{ij} &= \mu_j \int_0^{T_{ij}} r_i(t)\, dt \\ &= \mu_j \sum_{k=j}^{i-1} \int_{T_{i,k+1}}^{T_{ik}} r_i(t)\, dt \\ &= \mu_j \sum_{k=j}^{i-1} \left[\frac{e^{-u_k T_{i,k+1} - a_{i,k+1}} - e^{-u_k T_{ik} - a_{i,k+1}}}{u_k}\right] \quad i > j \geq 1. \end{aligned}$$

Of course $C_{i,j} = P_{i,j} = 0$ for $1 \leq i \leq j \leq M$. Since the matrix $\boldsymbol{C}$ is strictly lower triangular, the matrix inverse in (7.15) can be obtained as $(\boldsymbol{I} - \boldsymbol{C})^{-1} = \boldsymbol{I} + \boldsymbol{C} + \boldsymbol{C}^2 + \cdots + \boldsymbol{C}^{M-1}$.

Similarly, jumps to and from the idle state result in

$$\begin{aligned} P_{i,0} &= r_i(\tau) \\ &= \exp\left(-u_{i-1}\tau + \sum_{j=1}^{i-1} \mu_j (\tau - T_{ij})\right) \\ &= \exp\left(-\sum_{j=1}^{i-1} \mu_j T_{ij}\right) \\ &= e^{-a_{i,1}} \end{aligned}$$

and

$$P_{0,i} = \frac{\mu_i}{\mu}.$$

The expressions for the mean holding times are simplified in the same manner:

$$\begin{aligned} h_i &= \int_0^\tau r_i(t)\, dt \\ &= \int_0^{T_{i,1}} r_i(t)\, dt + \int_{T_{i,1}}^\tau r_i(t)\, dt \\ &= \frac{P_{i,1}}{\mu_1} + \int_{T_{i,1}}^\tau r_i(\tau)\, dt \\ &= \frac{P_{i,1}}{\mu_1} + (\tau - T_{i,1}) P_{i,0} \qquad i = 1, 2, \cdots, M \end{aligned}$$

and

$$h_0 = \frac{1}{\mu}.$$

Appendix C
Proof of Theorem 4.1—Stability of Dynamic Policy when $M = 2$

Let $\underline{\lambda} \in \mathscr{S}^D$, and let $(|n|, \phi)$ denote the polar coordinates of n, so $n = (|n|\cos(\phi), |n|\sin(\phi))$. Similarly, let $(|N(t)|, \phi_t)$ denote the polar coordinates of $N(t)$. The dynamic threshold matrix $\boldsymbol{T}$ depends only on $N_2(t)/N_1(t)$, and therefore only on ϕ_t. We thus write $\boldsymbol{T}(\phi_t)$ to denote the threshold matrix. Note that if $N_1(t)$ and $N_2(t)$ are large, then over a time interval beginning at time t that is long compared to τ but short compared to $N_1(t)$ and $N_2(t)$, the average drift vector for N over the interval is roughly approximately given by $\underline{\lambda} - \underline{\eta}^{T(\phi_t)}$.

Given a constant L, if $|N(t)|$ is sufficiently large then with high probability, $\phi_s \approx \phi_t$ for $s \in [t, t+L]$. If the thresholds were not variable the state of the server would be regenerative, where regeneration occurs whenever the server is idle. Furthermore, the time until the server reaches the idle state given it is serving a packet is at most 2τ time units. We can thus obtain the following. There is a constant c_1 such that given any L and δ, there exists a constant c_2, such that

$$|E[N(t+L) - N(t)|\mathscr{F}_t] - L(\underline{\lambda} - \underline{\eta}^{T(\phi_t)})| \leq c_1 \quad (8.1)$$

whenever $|N(t)| \geq c_2$ and $\phi_t \in [\delta, \pi/2 - \delta]$. If $N_2(t)$ is large but $N_1(t)$ is equal to or near zero, then the local throughput of type 1 packets may be less than $\eta_1(\phi_t)$, but the local throughput of type 2 packets is at least $\eta_2(\phi_t)$. Hence, there is a constant c_3 such that given any L and δ, there exists a constant c_4, such that

$$E[N_2(t+L) - N_2(t)|\mathscr{F}_t] - L(\lambda_2 - \eta_2^{T(\phi_t)}) \leq c_3 \quad (8.2)$$

whenever $|N_2(t)| \geq c_4$ and $\phi_t \in [\delta, \pi/2]$. Similarly, there is a constant c_5 such that given any L and δ, there exists a constant c_6, such that

$$E[N_1(t+L) - N_1(t)|\mathscr{F}_t] - L(\lambda_1 - \eta_1^{T(\phi_t)}) \leq c_5 \quad (8.3)$$

whenever $|N_1(t)| \geq c_6$ and $\phi_t \in [0, \pi/2 - \delta]$.

Let $\phi_o = \tan^{-1}(\lambda_2/\lambda_1)$. Our choice of $\boldsymbol{T}(\phi)$ insures that $\underline{\eta}^{T(\phi_o)}$ is a point on the boundary of $\mathscr{S}^D$ that dominates $\underline{\lambda}$. Thus, $\lambda_i - \eta_i^{T(\phi_o)} < 0$ for $i = 1, 2$. By continuity of $\underline{\eta}$ and $\boldsymbol{T}$, there is an $\epsilon > 0$ so that $\lambda_i - \eta_i^{T(\phi)} < 0$ for $\phi \in [\phi_o - \epsilon, \phi_o + \epsilon]$. Since $\eta_1^{T(\phi)}$ is decreasing in ϕ and $\eta_2^{T(\phi)}$ is increasing in ϕ, we have

$$\lambda_1 - \eta_1^{T(\phi)} < 0 \qquad \text{for } \phi \in [0, \phi_o + \epsilon] \quad (8.4)$$

$$\lambda_1 - \eta_2^{T(\phi)} < 0 \qquad \text{for } \phi \in [\phi_o - \epsilon, \pi/2]. \quad (8.5)$$

The next step of the proof is to use (8.2)–(8.5) to construct a function V so that $V(N_t)$ has negative drift whenever $|N(t)|$ is large.

Define a function $(a(\phi))_{0 \leq \phi \leq \pi/2}$ by

$$a(\phi) = \begin{cases} \dfrac{\cos(\phi)}{\lambda_1} & \phi \in [0, \phi_o - \epsilon] \\ \dfrac{\sin(\phi)}{\lambda_2} & \phi \in [\phi_o + \epsilon, \pi/2] \end{cases} \quad (8.6)$$

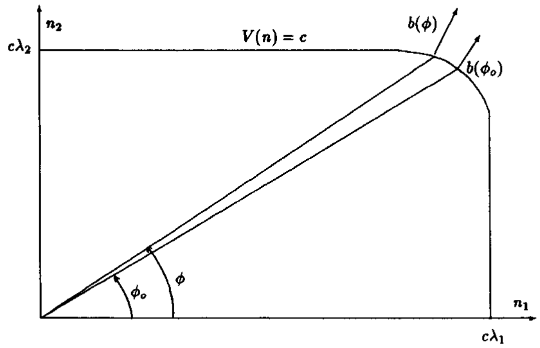


Fig. 6. A level curve of V with the gradient of V indicated at a typical point on the curve.

and take $a(\phi)$ for $\phi \in [\phi_o - \epsilon, \phi_o + \epsilon]$ so that $a(\phi)$ is concave on $[\phi_o - \epsilon, \phi_o + \epsilon]$ with continuous first and bounded second derivative on $[0, \pi/2]$. Then define the function V by $V(n) = |n|a(\phi)$. The function V is linear along rays starting at the origin, and the level curve $V(n) = c$ for a constant c is shown in Fig. 6. The gradient $\nabla V(n)$ is given by $\nabla V(n) = \underline{b}(\phi)$ where

$$\underline{b}(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} a(\phi) \\ a'(\phi) \end{pmatrix}. \tag{8.7}$$

The gradient vector $\underline{b}(\phi)$ is perpendicular to the level curve, as indicated in Fig. 6. Note that $\underline{b}(\phi)$ is nondegenerate and $b_i(\phi) \geq 0$ for $\phi \in [0, \pi/2]$ and $i = 1, 2$. Also,

$$b_1(\phi) = 0 \qquad \text{for } \phi \in [\phi_o + \epsilon, \pi/2] \tag{8.8}$$

$$b_2(\phi) = 0 \qquad \text{for } \phi \in [0, \phi_o - \epsilon]. \tag{8.9}$$

Combining (8.4), (8.5), (8.8), and (8.9), we obtain that

$$\underline{b}(\phi) \cdot \left(\underline{\lambda} - \underline{\eta}^{T(\phi)}\right) < 0 \qquad \text{for } \phi \in [0, \pi/2]. \tag{8.10}$$

Suppose $0 < \delta < \min(\phi_o, \pi/2 - \phi_o)$. Then (8.2), (8.3), (8.8), and (8.9) imply that there is a constant c_7 such that

$$E[(N(t + L) - N(t)) \cdot \underline{b}(\phi_t) | \mathcal{F}(t)] \leq \underline{b}(\phi_t)\left(\underline{\lambda} - \underline{\eta}^{T(\phi_t)}\right) + c_7 \tag{8.11}$$

whenever $|N(t)|$ is sufficiently large (how large depends on L). In view of (8.10), we can take L so large that $\underline{b}(\phi) \cdot (\underline{\lambda} - \underline{\eta}^{T(\phi)}) + c_7 \leq -2$ for $\phi \in [0, \pi/2]$. There is thus a constant c_8 so large that $E[(N(t + L) - N(t)) \cdot \underline{b}(\phi_t) | \mathcal{F}(t)] \leq -2$ whenever $|N(t)| \geq c_8$. Since V is asymptotically linear and lipschitz continuous, it follows that for some constant c_9, $E[V(N(t + L)) - V(N(t)) | \mathcal{F}(t)] \leq -2$ whenever $|N(t)| \geq c_8$. Consequently, the sequence $(V(N(kT)), \mathcal{F}(kT))_{k \geq 0}$ satisfies [3, conditions C.1 and C.2]. The sequence $V(N(kT))_{k \geq 0}$, and hence also the process $(N(t))_{t \geq 0}$, is exponentially stable. Theorem 4.1 is proved.

Acknowledgment

The authors are grateful for helpful comments from the anonymous reviewers. In particular, one reviewer suggested proving that threshold policies are optimal not only among nonlookahead preemptive policies, but also among all τ-lookahead policies.

References

[1] D. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[2] R. Binder, S. D. Huffman, I. Gurantz, and P. A. Vena, "Crosslink architectures for a multiple satellite system," *Proc. IEEE*, vol. 75, pp. 74–81, Jan. 1987.

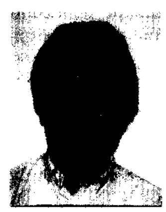
[3] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Advances in Applied Probability*, vol. 14, pp. 502–525, 1982.

[4] R. P. Kosowsky, I. M. Jacobs, and K. S. Gilhousen, "Arns: A new link layer protocol," in *IEEE Proc. MILCOM-88*, pp. 515–519, 1988.

[5] V. A. Mikhailov, "Geometrical analysis of the stability of Markov chains in R^n and its application to throughput evaluation of the adaptive random multiple access algorithm," *Problems Info. Transmission*, vol. 24, pp. 47–56, Jan.–Mar. 1988.

[6] L. Tassiulas and A. Ephremides, "Allocation of a single server to a set of parallel queues with time dependent demands," in *Proc. 1991 IEEE Int. Symp. Info. Theory*, p. 208, 1991.

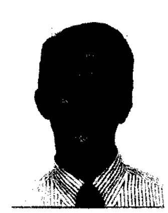
[7] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," preprint, Mar. 1992.



Bruce Hajek (M'79—SM'84—F'89) received the B.S. degree in mathematics and the M.S. degree in electrical engineering from the University of Illinois, Urbana Champaign, in 1976 and 1977, and the Ph. D. degree in electrical engineering from the University of California, Berkeley, in 1979.

He is currently a Professor in the Department of Electrical and Computer Engineering and in the Coordinated Science Laboratory at the University of Illinois, where he has been since 1979. His research interests include communication and computer networks, stochastic systems, combinatorial and nonlinear optimization and information theory.

Dr. Hajek has served as Editor-in-Chief (1989–1992) of the IEEE Transactions on Information Theory and is First Vice-President of the IEEE Information Theory Society.



Michael Carr received the B.S. degree in electrical engineering from the University of Lowell, MA (now the University of Massachusetts at Lowell) in 1989 and the M.S. degree from the University of Illinois, Urbana-Champaign, in 1991.

Since 1989, he has been a communications engineer at Raytheon, Missile Systems Division, where he was the recipient of a 1990 Aldo Miccioli Scholarship for Graduate Studies. His areas of interest include routing, IVHS, data fusion, and C^3 systems analysis.