Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures

SIMON S. LAM, MEMBER, IEEE, AND LEONARD KLEINROCK, FELLOW, IEEE

Abstract—**In a companion paper [1], the rationale for multiaccess broadcast packet communication using satellite and ground radio channels has been discussed. Analytic tools for the performance evaluation and design of uncontrolled slotted ALOHA systems have been presented. In this paper, a Markovian decision model is formulated for the dynamic control of unstable slotted ALOHA systems and optimum decision rules are found. Numerical results on the performance of controlled channels are shown for three specific dynamic channel control procedures. Several practical control schemes are also proposed and their performance compared through simulation. These dynamic control procedures have been found to be not only capable of preventing channel saturation for unstable channels but also capable of achieving a throughput-delay channel performance close to the theoretical optimum.**

Paper approved by the Associate Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received November 21, 1974; revised March 24, 1975. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC 15-73-C-0368.
S. S. Lam is with the IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y. 10598.
L. Kleinrock is with the Computer Science Department, University of California, Los Angeles, Calif. 90024.

I. INTRODUCTION

TRENDS in the growth of computer-communication networks seem to indicate that the next generation of networks will be at least an order of magnitude larger than existing designs. Present implementations, however, are not directly applicable to very large networks. New techniques are needed which can provide cost-effective high-speed communications for large populations of (potentially mobile) users scattered over wide geographical areas. Under these circumstances, packet switched satellite and ground radio systems are emerging as attractive solutions to the design of computer-communication networks and terminal access networks, respectively [1]–[6]. The rationale for packet switching using satellite and ground radio channels in a multiaccess broadcast mode has been examined in [1].

A multiaccess broadcast packet switching technique which has attracted considerable attention is the slotted ALOHA random access scheme [1]–[10]. This paper is a sequel to [1] in which an analytic model and a methodology for the performance evaluation of (uncontrolled) slotted ALOHA channels were developed. Specifically, a theory was proposed in [1] to explain the dynamic and stochastic channel behavior. Stable and unstable channels have been characterized. The trading relations among channel stability, channel throughput, and average packet delay were also shown.

In this paper, we study dynamic control procedures for unstable slotted ALOHA channels. The Markovian model formulated in [1], [6], [8] for a slotted ALOHA system is first introduced. Assuming that all channel users have exact knowledge of the instantaneous state of the system, three dynamic channel control procedures are described. A general Markovian decision model is next formulated by injecting two classes of control actions into the above model. State transition costs are defined so that the channel performance measures, namely, the stationary channel throughput rate S_{out} and the stationary average packet delay D, can be expressed in terms of the cost rates of the resulting Markovian decision processes. It is then shown that for the given model an optimal control policy exists which is a stationary policy and which maximizes S_{out} and minimizes D simultaneously. An efficient computational algorithm for finding the optimal control policy based upon Howard's policy-iteration method [11], [12] is next presented. The three specific control procedures, namely, the input control procedure (ICP), the retransmission control procedure (RCP), and the input-retransmission control procedure (IRCP), are then studied in more detail. Both numerical and simulation results are given for the throughput-delay performance of such controlled slotted ALOHA channels. In all cases considered, the optimal control policies were found to be of the control limit type. Next, we consider the fact that the exact instantaneous channel state is not generally known to individual channel users. A scheme is proposed which estimates the channel state and applies the above optimal control policies using this estimate. Another retransmission control procedure which circumvents the state estimation problem is also suggested. These practical control procedures are then tested through simulation and have been found to be capable of achieving a throughput-delay performance close to the theoretical optimum, as well as capable of preventing channel saturation under temporary overload conditions.

II. PRELIMINARIES

In this section, we first present the Markovian model formulated in [1], [6], [8] for a slotted ALOHA system. The stability behavior of uncontrolled slotted ALOHA channels is then discussed. Following that, some definitions are given for Markov decision processes. The three specific dynamic channel control procedures are then described. A general formulation of the problem as a Markovian decision model is given in Section III.

A. The Markovian Model for a Slotted ALOHA System

In a slotted ALOHA system, all users transmit packets into channel time slots independently. If two or more packet transmissions overlap in time at the multiaccessed radio receiver, it is assumed that none is received correctly. This event is referred to as a channel collision. We consider a slotted ALOHA channel with a user population consisting of M users. Each such user can be in one of two states: blocked or thinking.[1] In the thinking state, a user generates and transmits a new packet in a time slot with probability σ. A packet which had a channel collision and is waiting for retransmission is said to be backlogged. The retransmission delay RD of each backlogged packet is assumed to be geometrically distributed, i.e., each backlogged packet retransmits in the current time slot with probability p. Assuming bursty users, we must have $p \gg \sigma$. From the time a user generates a packet until that packet is successfully received, the user is blocked in the sense that he cannot generate (or accept from his input source) a new packet for transmission.

Let N^t be a random variable (called *channel backlog*) representing the total number of backlogged packets at time t. The channel input rate at time t is $S^t = (M - N^t)\sigma$. Assuming M and σ to be time-invariant, N^t is a Markov process (chain) with stationary transition probabilities and serves as the state description for the system. As in [1], [6], [8], we assume that p is given by

$$p = \frac{1}{R + (K+1)/2} \tag{1}$$

where R is the number of time slots in a round-trip channel propagation delay and the parameter K corresponds to the uniform retransmission randomization interval in [4].

B. Channel Stability

Consider (N^t,S^t) in the two-dimensional (n,S) plane. The trajectory of (N^t,S^t) is constrained to lie on the straight line $S = (M - n)\sigma$ which we refer to as the *channel load line*. For a fixed value of K (or p), there is an *equilibrium contour* in the (n,S) plane defined as the locus of points for which the channel input rate S is exactly equal to the expected channel throughput $S_{out}(n,S)$ in a time slot [1], [6], [8]. One such contour is illustrated in Fig. 1. Note that within the shaded region enclosed by the equilibrium contour, $S_{out}(n,S)$ is greater than S; elsewhere, S exceeds $S_{out}(n,S)$. Three channel load lines are also shown in Fig. 1 corresponding to the channel user population sizes M, M', and M'', and an average user think time of $1/\sigma$ slots. Arrows on the channel load lines point in the direction of "drift" of N^t.

A channel load line may intersect (nontangentially) the equilibrium contour one or more times, and we refer to

[1] This model is similar to the one studied by Metcalfe through a steady-state analysis [7]. He has also recognized the need for channel control and proposed a method for controlling the transmission probability of "ready" packets.

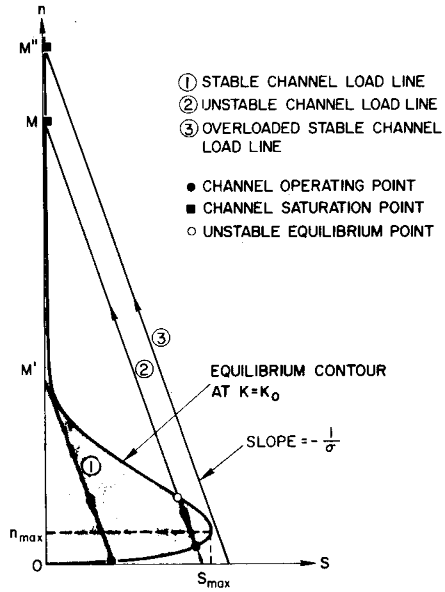

Fig. 1. Stable and unstable channels.

these as equilibrium points denoted by (n_e,S_e). An equilibrium point on a load line is said to be a *stable equilibrium point* if it acts as a "sink" with respect to the drift of N^t; it is said to be an *unstable equilibrium point* if it acts as a "source." A stable equilibrium point is said to be a *channel operating point* if $n_e \leq n_{\max}$ as shown in Fig. 1; it is said to be a *channel saturation point* if $n_e > n_{\max}$. (We shall use (n_o,S_o) instead of (n_e,S_e) to distinguish a channel operating point from other equilibrium points.) A channel load line is said to be *stable* if it has exactly one stable equilibrium point; otherwise, it is said to be *unstable*. Thus, the load lines labeled 1 and 3 in Fig. 1 are stable by definition; the load line labeled 2 is unstable. In a stable channel, the equilibrium point (n_e,S_e) determines the steady-state throughput-delay performance of the channel over an infinite time horizon. On the other hand, an unstable channel exhibits "bistable" behavior; the throughput-delay performance given by the channel operating point is achievable only for a finite time period before the channel drifts towards the channel saturation point. When this happens, the channel performance degrades rapidly as the channel throughput rate decreases and the average packet delay increases. The channel load line labeled 3 in Fig. 1 has a channel saturation point as its only stable equilibrium point. It is overloaded in the sense that M'' is too big for the given σ and K. From now on, a stable channel load line will always refer to 1 instead of 3 [1], [6], [8].

In Fig. 1, $S_{\max}$ represents the maximum possible throughput rate of the slotted ALOHA channel. For the infinite population model, $S_{\max}$ was shown to be $1/e \cong 0.368$ [3], [4].

Given a channel load line, suppose K_{opt} is the optimum K which minimizes n_o and maximizes S_o at the channel operating point. For this value of K, the channel may be unstable, in which case the optimum channel performance given by (n_o,S_o) is achievable only for a finite time period. To render the channel stable, two obvious solutions are available [1], [6], [8]: 1) use a larger value for K, and 2) allow a smaller user population size M. The first solution gives rise to a larger n_o; the corresponding average packet delay may then be too large to be acceptable [1]. In the second solution, a small M implies that $S_o \ll S_{\max}$ since $\sigma \ll 1$ under the assumption of bursty users. This results in a waste of channel capacity. A third solution is the use of dynamic channel control which constitutes the subject matter of this paper.

C. *Markov Decision Processes [11]–[13]*

Consider the Markov process (chain) N^t which is observed at time points $t = 0,1,2,\cdots$ to be in one of a finite number of states. The set of states $\mathcal{S}$ is labeled by the nonnegative integers $\{0,1,2,\cdots,M\}$. Let $\mathcal{A}$ be a finite set of possible actions such that corresponding to each action $\alpha \in \mathcal{A}$, a set of state transition probabilities $\{p_{ij}(\alpha)\}$ and a set of expected immediate costs $\{C_i(\alpha)\}$ are uniquely specified. We define a *policy* f to be any rule for choosing actions and $\mathcal{P}$ to be the class of all policies. The action chosen by a policy at time t may, for instance, depend on the complete history of the process up to that point.

Suppose the action α^t is given by the policy f at time t, which in turn specifies the state transition probabilities and costs at that time. Thus, f determines both the evolution in time of the Markov process N^t and the sequence of costs it incurs. For a policy f which generates the following sequence of actions in time $\{\alpha^0,\alpha^1,\alpha^2,\cdots,\alpha^t,\cdots\}$ we define the expected cost per unit time for N^t which was initially in state i as

$$\phi_i(f) \triangleq \lim_{\tau\to\infty} \frac{1}{\tau+1} E_f\left[\sum_{t=0}^{\tau} C_{N^t}(\alpha^t) \mid N^0 = i\right] \qquad (2)$$

where the limit always exists, since the costs are assumed to be bounded; the expectation is taken conditioning on the policy f.

An important subclass of all policies is the class of stationary policies $\mathcal{P}_s$. A *stationary policy* is defined to be one which chooses an action at time t depending only on the state of the process at time t. Thus, a stationary policy f is a function $f(\cdot):\mathcal{S}\to\mathcal{A}$. A Markov decision process employing a stationary policy f is a Markov process with stationary transition probabilities.

We now state several well-known results for finite-state Markov decision processes employing stationary policies.

Result 1: Given a stationary policy f such that N^t is irreducible we have

$$\phi_i(f) = \sum_{j=0}^{M} \pi_j(f)C_j(f) \triangleq g(f), \qquad \forall i = 0,1,\cdots,M \qquad (3)$$

where $\{\pi_j(f)\}$ is the unique stationary probability distribution of N^t such that

$$\pi_j(f) = \sum_{i=0}^{M} \pi_i(f) p_{ij}(f), \qquad j = 0,1,\cdots,M$$

$$\pi_i(f) \geq 0, \qquad i = 0,1,\cdots,M$$

and

$$\sum_{i=0}^{M} \pi_i(f) = 1.$$

$g(f)$ is said to be the *cost rate* or expected cost per unit time of the process N^t using policy f.

Result 2: If every stationary policy gives rise to an irreducible Markov chain, then there *exists* a stationary policy f^* which is optimal over the class of all policies such that

$$g(f^*) = \min_{f \in \mathcal{P}} \phi_i(f), \qquad \forall i = 0,1,\cdots,M.$$

Thus, by the above results, we may limit our attention only to the class of stationary policies. We shall present a computational procedure (to be described below) based upon Howard's policy-iteration method [11], [12] which evaluates the cost rate $g(f)$ given a stationary policy f and always leads to an optimal stationary policy within a finite number of computational steps.

Within the class of stationary policies, a subclass of policies known as control limit policies can be described as follows for a two-action space $\mathcal{A}$. Either the policy specifies the same action for all states in $\mathcal{S}$ or there is a critical state $\hat{n}$ $(=0,1,2,\cdots,$ or $M-1)$ such that if the policy specifies one action for states 0 to $\hat{n}$, the other action is specified for states $\hat{n}+1$ to M. $\hat{n}$ is said to be the *control limit.*

Finally, we assume that at any time t all channel users have perfect knowledge of the instantaneous channel state (perfect channel state information). This assumption is necessary in the mathematical model, but will be relaxed when we consider practical control procedures based upon insights gained from the analysis.

D. Dynamic Control Procedures

By a channel control procedure we mean the set of actions in the action space $\mathcal{A}$. Here we introduce three channel control procedures which will be studied below in more detail. They are special cases of the Markovian decision model in the next section.

The input control procedure (ICP): This control procedure corresponds to $\mathcal{A} = \{\text{accept, reject}\} \triangleq \{a,r\}$. Thus, in any channel state, the possible actions are: accept (action $= a$) or reject (action $= r$) *all* new packets that arrive[2] in the current time slot.

[2] A new packet is said to arrive in the current time slot only after it has been generated by the channel user (or its external source), processed, and ready for transmission over the channel in the current time slot. In the mathematical model, a rejected arrival is lost and the channel user generates a "new" packet in the next time slot with probability σ. In a practical system, this new packet must actually be the previously rejected packet! We shall elaborate upon this interpretation further below.

The retransmission control procedure (RCP): Under this control procedure, the action space $\mathcal{A} = \{p_o,p_c\} \triangleq \{o,c\}$ where p_o and p_c are said to be the *operating* and *control* values of the retransmission probability p. (Through (1), p_o corresponds to K_o which optimizes the channel operating point and p_c corresponds to K_c which is large enough to render the channel stable [1].) Obviously, we must have $p_c < p_o$. Thus, in any channel state the possible actions are: every backlogged packet is retransmitted in the current time slot with probability p_o (action $= o$) or with probability p_c (action $= c$).

The input-retransmission control procedure (IRCP): This control procedure is a combination of ICP and RCP with the action space $\mathcal{A} = \{(\text{accept}, p_o), (\text{accept}, p_c), (\text{reject}, p_o), (\text{reject}, p_c)\} \triangleq \{ao,ac,ro,rc\}$. Thus, for example, when the action rc is taken, both new and backlogged packets are delayed.

In Fig. 2(a) and (b), we show channel load lines corresponding to channels under ICP and RCP, respectively. We find it easier to illustrate both cases with control limit policies. In Fig. 2(a), $\hat{n}$ is the ICP control limit. When $N^t \leq \hat{n}$, the channel input rate is $S^t = (M - N^t)\sigma$; when $N^t > \hat{n}$, $S^t = 0$. Similarly, suppose $\hat{n}$ is the RCP control limit in Fig. 2(b). When $N^t \leq \hat{n}$, $K = K_o$, but as soon as N^t exceeds $\hat{n}$, $K = K_c$ is used. Note that both controlled channels are stable in the sense that the channel saturation point as shown in Fig. 1 no longer exists.

III. THE MARKOVIAN DECISION MODEL

In this section, a Markovian decision model is formulated which includes as special cases ICP, RCP, and IRCP introduced above. Expected immediate costs are defined so that the fundamental channel performance measures, namely, the stationary channel throughput rate S_{out} and the stationary average packet delay D, can be expressed in terms of the cost rates of the resulting Markov decision processes. Finally, it will be shown that an optimal stationary policy maximizes S_{out} and minimizes D simultaneously.

A. The Control Action Space

Consider the action space $\mathcal{A}_1 = \{\beta_1,\beta_2,\cdots,\beta_m\}$ where $0 \leq \beta_1 < \beta_2 < \cdots < \beta_m \leq 1$, and the action space $\mathcal{A}_2 = \{\gamma_1,\gamma_2,\cdots,\gamma_k\}$ where $0 < \gamma_1 < \cdots < \gamma_k < 1$. Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ such that each element in $\mathcal{A}$ is a two-dimensional vector (β,γ). As before, the Markov decision process N^t has the finite state space $\mathcal{S} = \{0,1,2,\cdots,M\}$. A stationary control policy f maps $\mathcal{S}$ into $\mathcal{A}$. Given a stationary control policy f, $f(i) = (\beta,\gamma)$ means that whenever $N^t = i$, each (new) packet arrival is accepted with probability β (and rejected with probability $1-\beta$) while each backlogged packet is retransmitted with probability γ in the tth time slot. Thus, ICP corresponds to the special case $\mathcal{A} = \{0,1\} \times \{p_o\}$; RCP corresponds to the special case $\mathcal{A} = \{1\} \times \{p_o,p_c\}$; IRCP corresponds to the special case $\mathcal{A} = \{0,1\} \times \{p_o,p_c\}$.

Suppose N^t is in state i and the stationary control policy

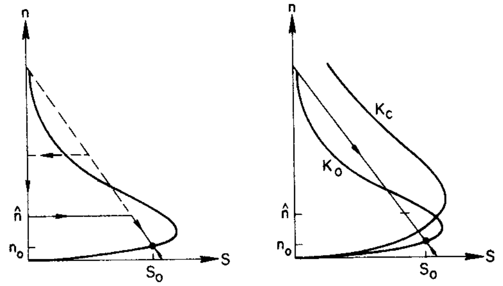

Fig. 2. (a) ICP control limit policy example. (b) RCP control limit policy example.

$f(i) = (\beta,\gamma)$, it is easy to show that the one-step state transition probabilities are given by

$$p_{ij}(f) = \begin{cases} 0, & j \leq i-2 \\ i\gamma(1-\gamma)^{i-1}(1-\beta\sigma)^{M-i}, & j = i-1 \\ (1-\gamma)^i(M-i)\beta\sigma(1-\beta\sigma)^{M-i-1} + [1-i\gamma(1-\gamma)^{i-1}](1-\beta\sigma)^{M-i}, & j = i \\ [1-(1-\gamma)^i](M-i)\beta\sigma(1-\beta\sigma)^{M-i-1}, & j = i+1 \\ \binom{M-i}{j-i}(\beta\sigma)^{j-i}(1-\beta\sigma)^{M-j}, & j \geq i+2 \end{cases} \quad (4)$$

for $0 \leq i, j \leq M$.

B. Cost Rates and Performance Measures

Suppose N^t is in state i and $f(i) = (\beta,\gamma)$. We define the expected immediate cost $C_i(f)$ to be

$$\begin{aligned} C_i(f) &= -\bar{S}_{\text{out}}(i,f) \\ &= -[i\gamma(1-\gamma)^{i-1}(1-\beta\sigma)^{M-i} \\ &\quad + (1-\gamma)^i(M-i)\beta\sigma(1-\beta\sigma)^{M-i-1}] \end{aligned} \quad (5)$$

where $\bar{S}_{\text{out}}(i,f)$ is the expected channel throughput[3] in the tth time slot. By (3) the cost rate of N^t is

$$g_s(f) = -\sum_{i=0}^{M} \pi_i(f)\bar{S}_{\text{out}}(i,f).$$

Thus, the stationary channel throughput rate is given by

$$S_{\text{out}} = -g_s(f). \quad (6)$$

Next, we show how to compute the stationary average packet delay D. Suppose N^t is in state i and $f(i) = (\beta,\gamma)$. For D, we define the expected immediate cost to be

$$C_i(f) = i + (M-i)(1-\beta)\sigma d_r \quad (7)$$

where the first term i represents a "holding cost" which accounts for the waiting cost of i packets incurred in the tth time slot; the second term represents a "rejection cost" which accounts for the additional delays incurred by rejected packets. Note that $(M-i)(1-\beta)\sigma$ is the expected number of packets rejected in the tth time slot. d_r is the expected cost in units of delay per packet arrival rejected. We shall assume that d_r is equal to an average user think time, i.e.,

$$d_r = \frac{1}{\sigma}. \quad (8)$$

This assumption is necessitated by our Markovian model formulation in which each thinking user is assumed to transmit a new packet (which may be a previously rejected new packet) with probability σ in a time slot. In an actual system, this random delay may be machine-introduced if needed.

Next, let $\mathcal{S} = \cup_{l=1}^{m} \mathcal{S}_l$ where $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_m$ are nonintersecting sets induced by the stationary control policy f such that

$$f(i) = (\beta_l,\gamma) \text{ if and only if } i \in \mathcal{S}_l$$

where $l = 1,2,\cdots,m$ and γ is any action in $\mathcal{A}_2$.

By (3), the cost rate of N^t is

$$\begin{aligned} g_d(f) &= \sum_{i=0}^{M} C_i(f)\pi_i(f) \\ &= \sum_{i=0}^{M} i\pi_i(f) + \sum_{l=1}^{m}\sum_{i\in\mathcal{S}_l} (M-i)(1-\beta_l)\sigma d_r\pi_i(f) \\ &= \bar{N} + \lambda_r d_r \\ &= \bar{N} + \bar{N}_r \end{aligned} \quad (9)$$

where

$$\lambda_r \triangleq \sum_{l=1}^{m}\sum_{i\in\mathcal{S}_l} (M-i)(1-\beta_l)\sigma\pi_i(f) \quad (10)$$

is the stationary packet rejection rate; $\bar{N}$ is the average channel backlog size, and $\bar{N}_r$ is the average number of rejected packets in the system by Little's result [14].

Applying Little's result once more, the average "wasted time" of a packet is

$$D_w = \frac{g_d(f)}{S_{\text{out}}} = -\frac{g_d(f)}{g_s(f)}. \quad (11)$$

[3] Expected number of successful packet transmissions in a time slot.

The average packet delay (in number of time slots) is given by

$$D = -\frac{g_d(f)}{g_s(f)} + R + 1 \tag{12}$$

where $R + 1$ represents the transmission and channel propagation delays incurred by each successful packet transmission.

Lemma: Given any stationary control policy $f: \mathcal{S} \rightarrow \mathcal{A}$,

$$g_d(f) = \frac{g_s(f)}{\sigma} + M. \tag{13}$$

Proof: From (3), (7), and (8)

$$g_d(f) = \sum_{i=0}^{M} i\pi_i(f) + \sum_{i=0}^{M} (M - i)\pi_i(f)$$

$$- \frac{1}{\sigma}\sum_{l=1}^{m} \sum_{i \in \mathcal{S}_l} (M - i)\beta_l\sigma\pi_i(f)$$

$$= M - \frac{1}{\sigma}\sum_{l=1}^{m} \sum_{i \in \mathcal{S}_l} (M - i)\beta_l\sigma\pi_i(f).$$

Note that $\sum_{l=1}^{m} \sum^{i \in \mathcal{S}_l} (M - i)\beta_l\sigma\pi_i(f)$ is just the stationary channel input rate and is thus equal to the stationary channel throughput rate $S_{\text{out}} = -g_s(f)$. Hence,

$$g_d(f) = \frac{g_s(f)}{\sigma} + M.$$

Q.E.D.

Theorem: For the above Markovian decision model: 1) there exists a stationary policy $\hat{f}$ such that

$$g_d(\hat{f}) = \min_{f \in \mathcal{P}_s} g_d(f)$$

if and only if

$$g_s(\hat{f}) = \min_{f \in \mathcal{P}_s} g_s(f);$$

and 2) if $\hat{f}$ is a stationary policy satisfying the preceding condition, then $\hat{f}$ minimizes D and at the same time maximizes S_{out} over the class $\mathcal{P}$ of all policies.

Proof: 1) This is a direct consequence of the above lemma and Result 2. 2) By (6) and (12), $\hat{f}$ minimizes D and maximizes S_{out} over the class of all stationary policies. The generalization to the class $\mathcal{P}$ of all policies is a consequence of Result 2. Q.E.D.

C. Optimum Channel Performance

Applying (6) and (13) to substitute for $g_d(f)$ and $g_s(f)$ in (12), we have

$$D = R + 1 + \left(\frac{M}{S_{\text{out}}} - \frac{1}{\sigma}\right) \tag{14}$$

which relates D as a one-to-one function of S_{out} given fixed values of R, M, and σ. Note that this function is monotonically decreasing with S_{out}.

Assuming a fixed R, we show in Fig. 3 a family of curves each of which depicts D as a function of S_{out} given by (14). The parameters M and σ, which determine the channel load line, also define a curve in the two-dimensional space of the performance measures D and S_{out}. We may consider a given control procedure as a mathematical operator which maps $\mathcal{P}_s$ (the space of all stationary policies) into one such curve. Each f in $\mathcal{P}_s$ is mapped into one point on the curve. The range space of the operator must be a proper subset of points on the curve. Otherwise, it is possible that $D = R + 1$ and $S_{\text{out}} = M\sigma$ (i.e., no congestion at all!). The optimization problem thus corresponds to finding the extreme points (maximum S_{out} and minimum D) of this range space. Since the curve under consideration is monotonically decreasing, these extreme points coincide. Thus, the same control policy f must maximize S_{out} and minimize D simultaneously.

Given a family of channel load lines (e.g., M varying from 0 to ∞ at fixed σ or σ varying from 0 to 1 at fixed M), each channel control procedure gives rise to an infeasible region such as shown in Fig. 3. The boundary of this region represents the optimum channel throughput-delay tradeoff under the above constraints.

IV. AN EFFICIENT COMPUTATIONAL ALGORITHM

Given a channel control procedure and a channel load line, we must determine the optimal control policy and the associated optimum values for S_{out} and D. Howard's policy-iteration method described in [11], [12] enables us to find an optimal policy usually in a small number of iterations. The method is composed of two parts: a value-determination operation and a policy-improvement routine. The difficulty now arises in the solution of the following $(M + 1)$ linear simultaneous equations in the value-determination operation for the cost rate g and the "relative values" v_i (setting $v_0 = 0$) when M is large (say, a few hundred, which is our range of interest).

$$g + v_i = C_i + \sum_{j=0}^{M} p_{ij}v_j, \qquad i = 0,1,2,\cdots,M. \tag{15}$$

If we take advantage of the fact that the state transition probabilities $p_{ij} = 0$ for $j \leq i - 2$ in our model, we may solve the above set of equations recursively using the following algorithm. The derivation of the algorithm is straightforward and is given in [6].

Algorithm 1: This algorithm solves for g and $\{v_i\}_{i=1}^{M}$ in the following set of $(M + 1)$ linear simultaneous equations:

$$g = C_0 + \sum_{j=1}^{M} p_{0j}v_j$$

$$g + v_1 = C_1 + \sum_{j=1}^{M} p_{1j}v_j$$

$$g + v_i = C_i + \sum_{j=i-1}^{M} p_{ij}v_j, \qquad i = 2,3,\cdots,M$$

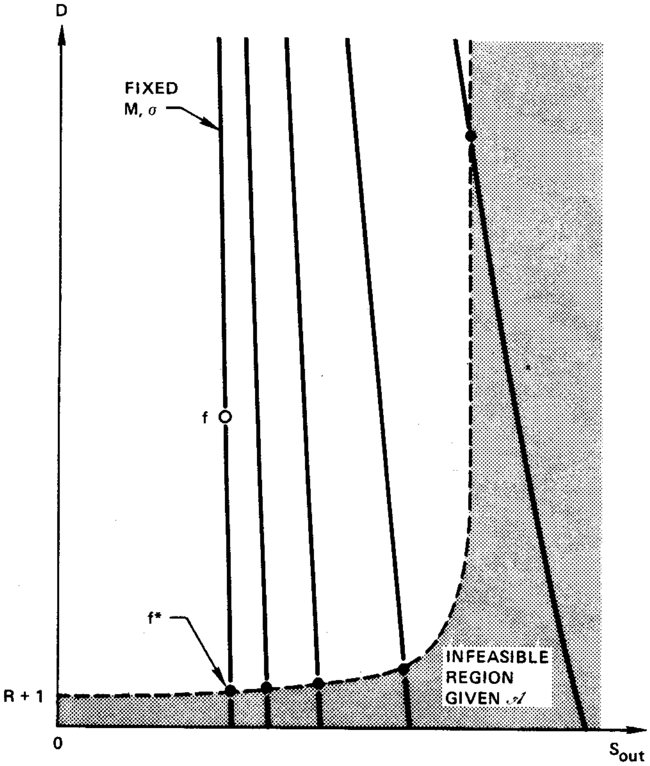

Fig. 3. Optimum performance of a channel control procedure.

where

$$\sum_{j=0}^{M} p_{0j} = \sum_{j=i-1}^{M} p_{ij} = 1, \qquad i = 1,2,\cdots,M.$$

The algorithm is as follows.

Step 1: Define

$$b_{M-1} = \frac{1}{p_{M,M-1}}$$

$$d_{M-1} = -\frac{C_M}{p_{M,M-1}}.$$

Step 2: For $i = M-1, M-2,\cdots,2$ solve recursively

$$b_{i-1} = \frac{1}{p_{i,i-1}}\left[b_i + 1 - \sum_{j=i}^{M-1} p_{ij}b_j\right]$$

$$d_{i-1} = \frac{1}{p_{i,i-1}}\left[d_i - C_i - \sum_{j=i}^{M-1} p_{ij}d_j\right].$$

Step 3: Define

$$u_M = -\frac{1}{p_{10}}\left[b_1 + 1 - \sum_{j=1}^{M-1} p_{1j}b_j\right]$$

$$w_M = -\frac{1}{p_{10}}\left[d_1 - C_1 - \sum_{j=1}^{M-1} p_{1j}d_j\right]$$

$$u_i = u_M + b_i$$

$$w_i = w_M + d_i, \qquad i = 1,2,\cdots,M-1.$$

Step 4: Finally,

$$g = \frac{C_0 + \sum_{j=1}^{M} p_{0j}w_j}{1 - \sum_{j=1}^{M} p_{0j}u_j}, \qquad v_i = u_i g + w_i,\ i = 1,2,\cdots,M.$$

Algorithm 1 has the advantage that the crucial variables b_i and d_i in the algorithm are computed recursively such that the state transition probabilities p_{ij} can be computed as needed. This eliminates the need for storing the $\{[(M+1)(M+2)]/2\} + M$ elements in the state transition matrix and virtually eliminates any machine storage constraint on the dimensionality of the optimization problem. The number of arithmetic operations required is comparable to that of a standard solution method such as Gauss elimination [15].

We present below our computational procedure (POLITE) for the Markovian decision model, which combines the policy-iteration method, Algorithm 1, and the above Lemma. Given a channel load line and a dynamic channel control procedure, POLITE finds the optimal control policy and evaluates the optimum channel performance measures.

Algorithm 2 (POLITE):

Step 1: Given a policy f, apply Algorithm 1 to obtain g and $\{v_i\}_{i=1}^{M}$; $p_{ij}(f)$ and $C_i(f)$ are computed when needed in the algorithm.

Step 2: For state $i = 0,1,\cdots,M$ define the test quantity

$$\text{Cost}(i,\alpha) = C_i(\alpha) + \sum_{j=1}^{M} p_{ij}(\alpha)v_j.$$

Find $\hat{\alpha}$ such that $\text{Cost}(i,\hat{\alpha}) = \min_{\alpha\epsilon\mathfrak{A}} \text{Cost}(i,\alpha)$. If $\text{Cost}(i,f(i)) = \text{Cost}(i,\hat{\alpha})$, then let $\hat{f}(i) = f(i)$; otherwise, let $\hat{f}(i) = \hat{\alpha}$.

Step 3: If $\hat{f}$ and f are identical, go to Step 4; otherwise, replace f by $\hat{f}$ and go to Step 1.

Step 4: f is an optimal control policy; $g = g_s(f)$ or $g_d(f)$ depending on the expected immediate costs $C_i(\alpha)$.

Step 5: Applying (6), (12), and (13), the optimum performance measures are

$$S_{out}^* = -g_s(f)$$

$$D^* = -\frac{g_d(f)}{g_s(f)} + R + 1.$$

V. NUMERICAL RESULTS

Numerical results have been obtained for the ICP, RCP, and IRCP control procedures using POLITE. In this section, we first discuss the optimality of control limit policies. The performance of controlled channels under ICP, RCP, and IRCP is then shown. More specific computational considerations are discussed in [6], [10].

A. Control Limit Policies

Consider ICP and RCP. The action space $\mathfrak{A}$ of both control procedures consists of two actions $\{\alpha_o,\alpha_c\}$. α_o is the *operating action,* designed to give near optimum channel throughput-delay performance under equilibrium conditions. α_o corresponds to "accept" in ICP and p_o in RCP. α_c is the *control action,* designed to prevent the channel from drifting into saturation. α_c corresponds to "reject" in ICP and p_c in RCP.

Our intuition suggests that a good control policy (for either ICP or RCP) must be such that the control action is applied whenever the channel backlog size N^t exceeds some threshold value to prevent it from drifting away. But as soon as N^t decreases below this threshold value, the control action should be replaced by the operating action for optimum performance. This intuition has been confirmed in all our numerical solutions for ICP and RCP. In each case, the optimal control policy given by POLITE is a control limit policy of the following form:

$$f(i) = \begin{cases} \alpha_o, & i \leq \hat{n} \\ \alpha_c, & i > \hat{n} \end{cases} \tag{16}$$

where $\hat{n}$ is said to be the control limit (CL) of the control limit policy f. A rigorous proof of the optimality of the control limit policy remains an open problem. Some difficulties in the pursuit of such a proof are discussed in [6].

B. Performance of Controlled Channels

In this section we show the throughput-delay performance of controlled slotted ALOHA channels under ICP, RCP, and IRCP. The following numerical constants corresponding to a satellite channel are assumed. Note that a satellite channel is characterized by a large round-trip channel propagation delay R (compared to ground radio). R will be taken to be 12 channel time slots and each time slot is 22.5 ms long, giving 44.4 slots/s. The above figures are computed from the assumptions of a 50 kbit/s satellite voice channel, 1125 bits/packet, and a round-trip channel propagation time of 0.27 s for all channel users. The duration of a channel time slot is assumed to be the same as a packet transmission time. From our discussion in the previous section, all control policies considered below for ICP and RCP are of the CL type.

Choosing K_o: Given an unstable channel, the throughput-delay performance at the operating point (n_o, S_o) is what we strive to achieve through dynamic channel control. Thus, it is essential to choose the operating value of K to be K_{opt} or some K_o which yields an operating point close to the optimum. For the numerical constants given above, $K = 10$ is an excellent choice and will be used throughout this paper as the operating value K_o [1], [6].

Specifying a channel load line: The channel load line is a straight line uniquely specified by its intercept on the vertical axis, M, and its slope $-1/\sigma$. Alternatively, it may be specified by M and the operating point (n_o, S_o) on the equilibrium contour (instead of σ). Thus, different load lines specified by the same channel operating point can be compared by showing how well they approach the throughput-delay performance at the operating point.

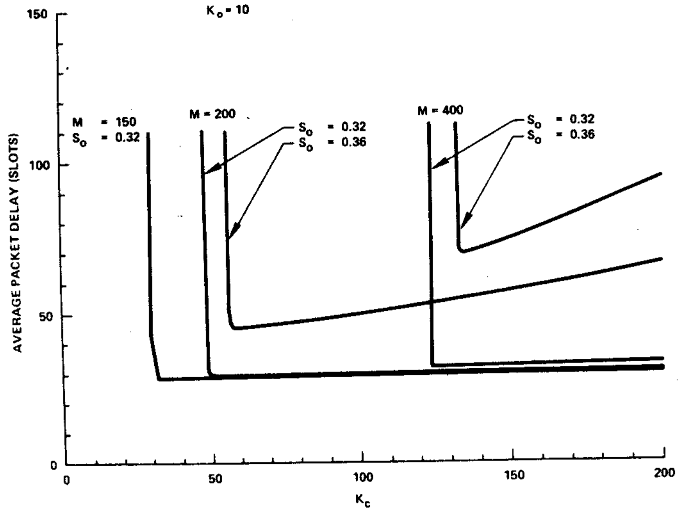

Fig. 4. RCP channel performance versus K_c.

Choosing K_c for RCP: In RCP the control action is to use a large enough value of K, namely, K_c which renders the channel load line stable [1], [6], [8]. We illustrate this last statement in Fig. 4 in which the average packet delay D given by an optimal RCP policy is shown as a function of K_c. Observe that for a sufficiently large K_c, D is quite insensitive to its exact value except when $S_o \geq 0.36$, in which case D increases slowly with K_c. Note that for the same S_o, a much larger K_c is required for a larger M. In the limit as $M \rightarrow \infty$, RCP becomes ineffective since no sufficiently large value of K can be used for K_c. Given a channel load line, a suitable value of K_c may be determined graphically from a family of equilibrium contours for different K [1], [6], [8].

Channel performance under ICP and RCP control limit policies: We show in Figs. 5 and 6 the channel performance measures S_{out} and D over a range of ICP and RCP control limits for $M = 200$ and $S_o = 0.32, 0.36$. Observe in Fig. 5 that a single control limit minimizes D and maximizes S_{out} as predicted by the theory. Note the amazing flatness of S_{out} and D near the optimum point, especially when $S_o = 0.32$. The consequence is that even if a nonoptimal control policy is used (due to, for example, not knowing the exact instantaneous backlog size such as in most practical systems), it is still possible to achieve a throughput-delay performance close to the optimum. However, such flatness of S_{out} and D is not as pronounced when S_o is 0.36. The optimum values of S_{out} and D given by ICP and RCP are approximately the same, but RCP gives less severe degradation in channel performance when the control limit policy is different from the optimal. However, recall from Fig. 4 the potential disastrous channel performance if K_c is not sufficiently large. This must be taken into consideration in any system design using RCP since in a practical system both the parameters M and σ may change with time. To provide the necessary design safety margin, a much bigger value of K_c than deemed necessary may have to be adopted. In Fig. 6, we show the degradation in channel performance when $K_c = 200$ is

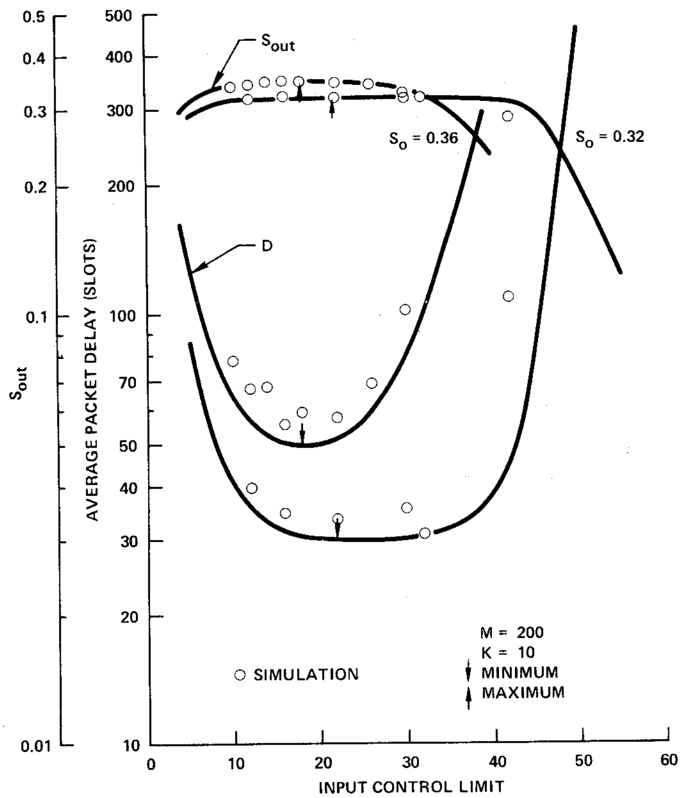

Fig. 5. Channel performance versus ICP control limit for $M = 200$.

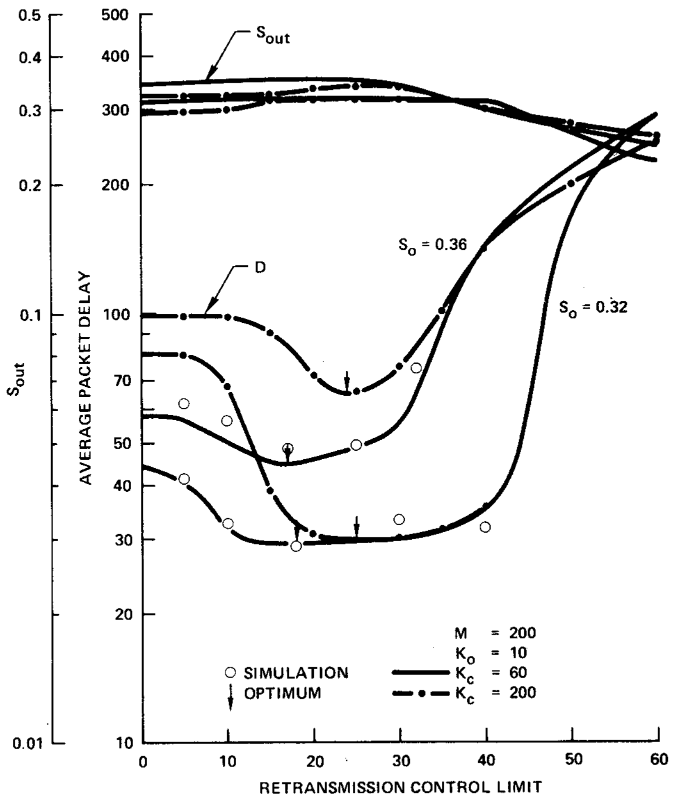

Fig. 6. Channel performance versus RCP control limit for $M = 200$.

used instead of $K_c = 60$. On the other hand, M has relatively little effect on the optimal ICP control limit as shown in Fig. 7(a). Thus, even if M fluctuates in time in a real system, the same ICP control limit policy is still near optimal. Of course, the optimum channel performance must deteriorate as M increases as shown in Fig. 7(b). In the same figure, the optimum D given by ICP and RCP are compared. RCP is found to be slightly better than ICP. However, as M becomes large, K_c must also be large, in which case the trend indicates that ICP is superior to RCP.

In Figs. 5 and 6, we have also indicated simulation results for throughput and delay. In these simulations, channel control policies are applied assuming that the exact instantaneous channel backlog size N^t is known to all channel users. However, contrary to the Markovian model, each collided packet is assumed to suffer the more realistic fixed delay of R slots and its retransmission randomized uniformly over the next K slots [4]. The Markovian model is idealized since R is assumed to be zero while each backlogged packet retransmits in a time slot with probability p. (In both cases, the same average retransmission delay was used.) This excellent agreement between simulation and analytic results presented here demonstrates the usefulness of the Markovian model for a slotted ALOHA system.

Optimum throughput-delay tradeoffs: Optimum throughput-delay tradeoffs at fixed M are shown in Fig. 8 for ICP. Note that S_{out} is maximized and D minimized by the optimal ICP control limit $\hat{n} = 22$ for a specific channel load

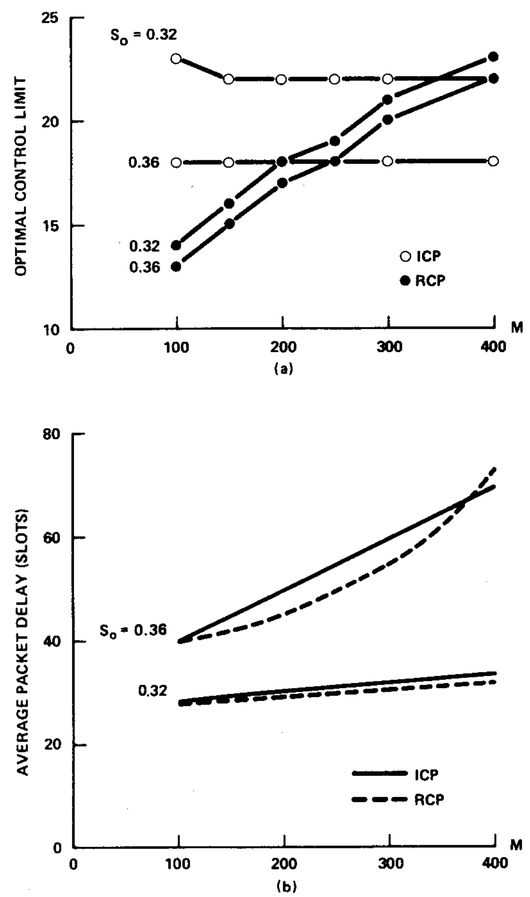

Fig. 7. ICP and RCP channel performance versus M.

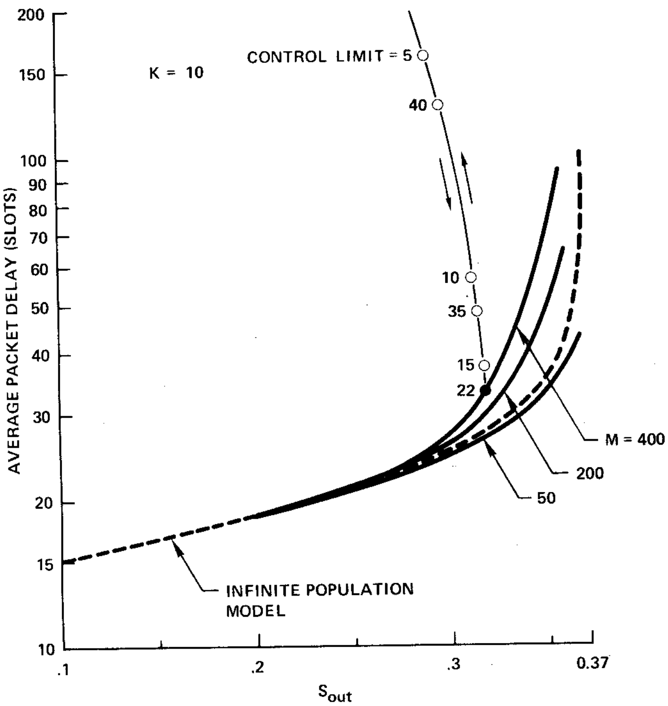

Fig. 8. ICP optimum throughput-delay tradeoffs at fixed M.

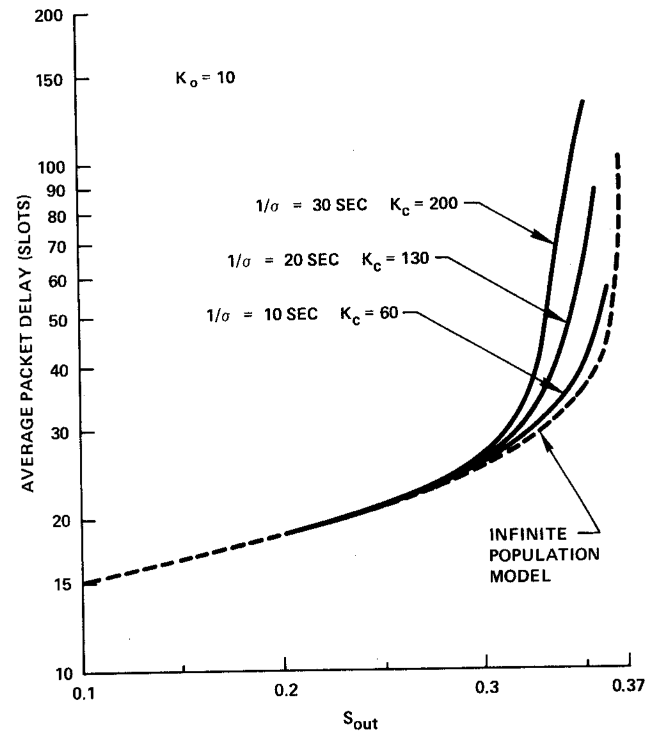

Fig. 9. RCP optimum throughput-delay tradeoffs at fixed σ.

line. Note how close the ICP throughput-delay tradeoff curves are to the optimum envelope obtained for the infinite population model in [4]. In fact, the $M = 50$ tradeoff curve lies a little below the optimum envelope. This is to be expected since $M = 50$ actually gives rise to a stable channel, in which case the throughput-delay performance at the optimum operating point is achieved.

In Fig. 9, optimum throughput-delay tradeoffs at fixed values of σ for RCP are shown. ($1/\sigma$ is the average think time of a channel user.) In this case, increasing S_{out} corresponds to increasing M. Note that the channel performance improves as the packet generation probability σ increases, since this implies that for the same S_{out}, the number of channel users is smaller and these users are also less "bursty."

IRCP channel performance: Recall that the ICP and RCP action spaces are both subspaces of the IRCP action space. Therefore, the channel performance given by IRCP must be better or at least as good as that given by ICP or RCP. In Table I, we compare these three control procedures for the four channel load lines involving $M = 200$, 400 and $(n_o,S_o) = (4,0.32),(7,0.36)$. Note that in every instance, IRCP gives the best performance. We also observed that the optimal control policy for IRCP is of the form

$$f(i) = \begin{cases} ao, & 0 \le i \le \hat{n}_1 \\ ac, & \hat{n}_1 < i \le \hat{n}_2 \\ rc, & \hat{n}_2 < i \end{cases} \tag{17}$$

which is uniquely specified by $(\hat{n}_1,\hat{n}_2)$. Also, $\hat{n}_1$ is either equal or very close to the optimal RCP control limit in

TABLE I
COMPARISON OF ICP, RCP, AND IRCP

		$M = 200$, $S_o = 0.32$ $(K_c = 60)$	$M = 200$, $S_o = 0.36$ $(K_c = 60)$	$M = 400$, $S_o = 0.32$ $(K_c = 150)$	$M = 400$, $S_o = 0.36$ $(K_c = 150)$
$\hat{n}$	ICP	22	18	22	18
$\hat{n}$	RCP	18	17	23	22
$(\hat{n}_1,\hat{n}_2)$	IRCP	(18,56)	(17,43)	(23,116)	(23,91)
S_{out}	ICP	0.31778	0.34925	0.31807	0.34846
	RCP	0.31817	0.35217	0.31844	0.34715
	IRCP	0.31817	0.35219	0.31844	0.34847
D	ICP	29.857	49.552	33.096	69.237
	RCP	29.085	44.802	31.608	73.588
	IRCP	29.085	44.772	31.608	69.215

each case and the use of $\hat{n}_2$ brings about only minor improvement in the channel performance except in the case of $M = 400$ and $S_o = 0.36$.

VI. PRACTICAL CONTROL SCHEMES

The optimum throughput-delay channel performance given in the last section is achievable over an infinite time horizon if the channel users have exact knowledge of the channel state at any time. In a practical system, the channel users often have no means of communication among themselves other than the multiaccess broadcast channel itself. Each channel user may individually estimate the channel state by observing the (broadcasted) outcome in each channel slot. However, whatever channel state information available to the channel users is at least as old as one round-trip propagation delay (R) which may introduce additional errors in the users' estimates if R is large (such as in a satellite channel). Thus, the control action applied based upon an estimate of the channel state may not necessarily be the optimal one at that time, which will then lead to some degradation in channel performance.

Below we first give a heuristic scheme for estimating the channel state assuming that the history (i.e., empty slots, successful transmissions or collisions) of the channel is available to all channel users. The optimal ICP, RCP, and IRCP control policies may be applied based upon this estimate. A heuristic control procedure is next proposed which circumvents the state estimation problem. These control procedures are then examined through simulation and compared with the theoretical optimum throughput-delay results in the previous section. The ability of these control procedures to handle time-varying inputs (with pulses) is also examined.

A. Channel Control-Estimation Algorithms

The *channel traffic* in a time slot is defined to be the number of packet transmissions (both new and previously collided packets) by all users in that time slot. Our heuristic scheme for estimating the channel state is based upon the observation that the channel traffic in a time slot is approximately Poisson distributed. (See [6, ch. 4 and Appendix A].) Algorithms which implement channel control procedures using the Poisson channel traffic estimate will be referred to as control-estimation (CONTEST) algorithms.

We illustrate below a procedure for implementing RCP. Similar algorithms for implementing ICP and IRCP are given in [6], [9], [10]. Let $\hat{n}$ be the RCP control limit. Define

$$\hat{G}_o = \hat{n}p_o + (M - \hat{n})\sigma \tag{18}$$

$$\hat{G}_c = \hat{n}p_c + (M - \hat{n})\sigma. \tag{19}$$

$\hat{G}_o$ and $\hat{G}_c$ are thus the channel traffic rates when the channel backlog size is $\hat{n}$ packets with K equal to K_o and K_c, respectively. Assuming that the channel traffic is Poisson distributed, we define the following critical values (corresponding to the probability of zero channel traffic in a time slot):

$$\hat{f}_o = \exp\left[-\hat{G}_o\right] \tag{20}$$

and

$$\hat{f}_c = \exp\left[-\hat{G}_c\right]. \tag{21}$$

Since $K_c > K_o$ we must have

$$\hat{f}_o < \hat{f}_c.$$

Suppose each channel user keeps track of the channel history (delayed by one round-trip propagation time) within a window frame of W slots. Let $\bar{f}^t$ be the fraction of empty slots within the history window for the tth time slot. $\bar{f}^t$ will closely approximate the probability of zero channel traffic in the tth time slot provided that 1) the channel traffic probability distribution does not change appreciably in $(W + R)$ time slots, 2) that $W \gg 1$, and 3) that the Poisson traffic assumption holds. We give the following CONTEST algorithm to be adopted by each channel user. d^t denotes the control decision at time t.

Algorithm 3 (RCP-CONTEST): This algorithm generates the decision $d^t = K_o, K_c$ at each time point based upon the channel state estimate $\bar{f}^t$ and the RCP control limit $\hat{n}$. Start at Step 1 or Step 4.

Step 1:

$$t \leftarrow t + 1$$

$$d^t = K_o.$$

Step 2: If $\bar{f}^t < \hat{f}_o$, go to Step 4.

Step 3: Go to Step 1.

Step 4:

$$t \leftarrow t + 1$$

$$d^t = K_c.$$

Step 5: If $\bar{f}^t > \hat{f}_c$, go to Step 1.

Step 6: Go to Step 4.

The channel history window: The size W of the channel history window kept by each channel user is very important for successful channel state estimation. If W is too large, we may lose information on the dynamic behavior of the channel such that the necessary actions are taken too late. If W is too small, we may get large errors in approximating the probability of zero channel traffic by the fraction of empty slots in the history window. A good initial estimate is that W should be bigger than R and of the same order of magnitude. Below we compare simulation results on channel performance for different values of W.

To implement the channel state estimation procedure, each channel user needs to maintain the channel history for W slots. Since it is only necessary to record whether or not a slot is empty, W bits of information suffice. A possible implementation is depicted schematically in Fig. 10. The bit string stored in the shift register represents the channel history in a window of W slots. An empty channel slot is represented by "1" while a nonempty channel slot is represented by "0." In the figure, the circle represents a summer, the triangle an attenuator, and the square a unit delay of one slot. Simulation results on the channel performance given by the CONTEST algorithms will be examined below.

B. Another Retransmission Control Procedure

We describe in this section a simple heuristic control procedure which has the property that as the channel traffic increases the retransmission delays of backlogged packets will also increase. Hence, it will be referred to as the heuristic retransmission control procedure (heuristic RCP). The advantage of such a control procedure is that it is simple and can be implemented easily without any need for monitoring the channel history and estimating the channel state.

Algorithm 4 (heuristic RCP): For a backlogged packet with m previous channel collisions, the uniform retrans-

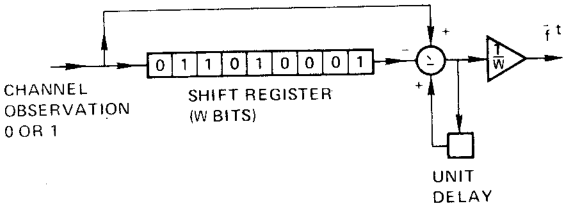

Fig. 10. Determination of $\bar{f}^t$.

mission randomization[4] interval is taken to be $K = K_m$ where K_m is a monotone nondecreasing function in m.

When the channel traffic increases, the probability of channel collision increases. As a result, the "effective" value of K increases. If K_m is a steep enough function in m, we see that channel saturation will be prevented.

C. Simulation Results

We summarize, in Tables II and III, throughput-delay results for the channel load lines specified by $(n_o, S_o) = (4, 0.32)$ and $M = 200, 400$. In both cases, we assume that $K_o = 10$ and $K_c = 60$. Included in these tables are 1) optimum POLITE results for ICP, RCP, and IRCP, 2) simulation results for ICP and RCP using optimal control policies and under the assumption of perfect channel state information, 3) simulation results for the CONTEST algorithms using ICP and RCP optimal control policies, and 4) simulation results for heuristic RCP. The duration of each simulation run was taken to be 30 000 time slots. IRCP was not tested by simulation since the optimal value of $\hat{n}_2$ is in all cases so large that within the simulation duration, the channel state N^t (almost surely) will not exceed it; the control procedure becomes effectively RCP specified by $\hat{n}_1$.

The ICP-CONTEST algorithm was tested with channel history window sizes of 20, 40, 60, and 80 time slots. We see from Tables II and III that $W = 40$ appears to give the best throughput-delay results. Note that for $R = 12$ and $K = 10$, $W = 40$ is approximately twice $R + K$.

The RCP-CONTEST algorithm was also tested with various values of W. In this case, K takes on two values, $K_o = 10$ and $K_c = 60$. There is no clear-cut optimal W. It appears that $W = 60$ is a good choice. There is no significant degradation in channel performance (from the optimum) given by the CONTEST algorithms and heuristic RCP. The CONTEST algorithms, however, seem to have an edge over heuristic RCP. The excellent performance of the CONTEST algorithms can be attributed to the flatness of S_{out} and D near the optimum as a function of the control limit (see Figs. 5 and 6). We found that this flatness property is less pronounced for channel load lines with a large value of S_o or M, such as $S_o = 0.36$ or $M = 400$. This explains the more significant degradation in channel performance given by the CONTEST algorithms shown in Table III for $M = 400$ than in Table II for $M = 200$.

For an uncontrolled slotted ALOHA channel, it is shown in [6] that a channel input rate of 0.8 packet/

[4] Or, equivalently, $p = p_m$ where p_m is a monotone nonincreasing function in m.

TABLE II
THROUGHPUT-DELAY RESULTS OF A CONTROLLED CHANNEL ($M = 200, S_o = 0.32$)

CONTROL SCHEME	S_{out}	D
ICP (POLITE)	0.3178	29.9
RCP (POLITE)	0.3182	29.1
IRCP (POLITE)	0.3182	29.1
ICP (Simulation)	0.315	33.4
RCP (Simulation)	0.318	28.8
ICP-CONTEST W = 20	0.314	40.9
ICP-CONTEST W = 40	0.315	30.5
ICP-CONTEST W = 60	0.317	32.4
ICP-CONTEST W = 80	0.318	35.8
RCP-CONTEST W = 20	0.315	33.1
RCP-CONTEST W = 40	0.322	33.3
RCP-CONTEST W = 60	0.319	32.1
RCP-CONTEST W = 80	0.317	32.5
Heuristic RCP { $K_1 = 10$, $K_m = 60$ $m \geq 2$ }	0.316	33.7
	0.315	34.6
Heuristic RCP { $K_1 = 10$, $K_2 = 60$, $K_m = 120$ $m \geq 3$ }	0.310	35.4
	0.316	34.6

TABLE III
THROUGHPUT-DELAY RESULTS OF A CONTROLLED CHANNEL ($M = 400, S_o = 0.32$)

CONTROL SCHEME	S_{out}	D
ICP (POLITE)	0.3181	33.1
RCP (POLITE)	0.3184	31.6
IRCP (POLITE)	0.3184	31.6
ICP (Simulation)	0.315	31.4
RCP (Simulation)	0.317	31.0
ICP-CONTEST W = 20	0.315	43.3
ICP-CONTEST W = 40	0.314	34.7
ICP-CONTEST W = 60	0.312	53.2
ICP-CONTEST W = 80	0.316	39.1
RCP-CONTEST W = 20	0.313	41.1
RCP-CONTEST W = 40	0.319	43.4
RCP-CONTEST W = 60	0.318	38.8
RCP-CONTEST W = 80	0.317	40.1
RCP-CONTEST W = 100	0.314	35.7
RCP-CONTEST W = 120	0.319	47.1
Heuristic RCP { $K_1 = 10$, $K_m = 150$ $m \geq 2$ }	0.316	45.2
	0.316	44.8
Heuristic RCP { $K_1 = 10$, $K_2 = 100$, $K_m = 200$ $m \geq 3$ }	0.312	42.0
	0.311	43.1

TABLE IV
SIMULATION RUN FOR IRCP-CONTEST SUBJECT TO A CHANNEL INPUT PULSE

INPUT PARAMETERS:
NUMBER OF TERMINALS $M = 400$, PROPAGATION DELAY $R = 12$
FOR THE TIME PERIOD 1-1000, INPUT RATE $M\sigma = 0.3232$
FOR THE TIME PERIOD 1001-1200, INPUT RATE $M\sigma = 1.0$
FOR THE TIME PERIOD 1201-6000, INPUT RATE $M\sigma = 0.3232$
RETRANSMISSION CONTROL LIMIT = 23, INPUT CONTROL LIMIT = 116
$K_o = 10$, $K_c = 150$, WINDOW SIZE $W = 60$

AVERAGE VALUES IN 200 TIME SLOT PERIODS:

Time Period	Throughput rate	Traffic rate	Average delay	Average Backlog	Packets Rejected
1 - 200	0.290	0.625	30.2	5.5	0
201 - 400	0.325	0.700	34.0	6.9	0
401 - 600	0.285	0.450	23.6	2.8	0
601 - 800	0.295	0.625	31.7	5.9	0
801 - 1000	0.325	0.850	42.5	9.3	0
1001 - 1200	0.205	2.345	524.3	50.0	49
1201 - 1400	0.345	1.330	389.6	75.2	13
1401 - 1600	0.355	0.880	188.1	51.5	0
1601 - 1800	0.375	0.735	179.3	34.9	0
1801 - 2000	0.225	1.295	297.8	33.7	5
2001 - 2200	0.325	1.005	530.7	35.3	21
2201 - 2400	0.380	0.905	127.3	16.7	0
2401 - 2600	0.305	0.485	27.6	3.1	0
2601 - 2800	0.290	0.430	20.8	2.3	0
2801 - 3000	0.345	0.745	35.3	7.2	0
3001 - 3200	0.300	0.455	17.6	2.4	0
3201 - 3400	0.280	0.615	28.4	5.6	0
3401 - 3600	0.390	0.810	37.9	7.9	0
3601 - 3800	0.330	0.655	30.6	5.6	0
3801 - 4000	0.300	0.390	19.3	1.7	0
4001 - 4200	0.315	0.615	29.2	5.1	0
4201 - 4400	0.335	0.600	24.5	4.5	0
4401 - 4600	0.300	0.450	24.3	2.6	0
4601 - 4800	0.280	0.480	25.2	3.7	0
4801 - 5000	0.285	0.585	32.0	5.3	0
5001 - 5200	0.330	0.570	26.4	4.3	0
5201 - 5400	0.335	0.550	23.6	3.7	0
5401 - 5600	0.335	0.640	28.8	5.2	0
5601 - 5800	0.275	0.410	21.5	2.4	0
5801 - 6000	0.285	0.445	22.3	2.7	0

TABLE V
SIMULATION RUN FOR HEURISTIC RCP SUBJECT TO A CHANNEL INPUT PULSE

INPUT PARAMETERS:
NUMBER OF TERMINALS $M = 400$, PROPAGATION DELAY $R = 12$
FOR THE TIME PERIOD 1-1000, INPUT RATE $M\sigma = 0.3232$
FOR THE TIME PERIOD 1001-1200, INPUT RATE $M\sigma = 1.0$
FOR THE TIME PERIOD 1201-6000, INPUT RATE $M\sigma = 0.3232$
$K_1 = 10$ $K_m = 150$ $(m \geq 2)$

AVERAGE VALUES IN 200 TIME SLOT PERIODS:

Time Period	Throughput rate	Traffic rate	Average delay	Average Backlog
1 - 200	0.285	0.395	19.8	2.1
201 - 400	0.320	0.390	16.3	1.2
401 - 600	0.255	0.425	22.8	2.8
601 - 800	0.290	0.475	26.1	4.0
801 - 1000	0.325	0.570	28.5	5.7
1001 - 1200	0.230	2.395	34.1	68.8
1201 - 1400	0.285	1.695	141.3	112.6
1401 - 1600	0.310	1.500	273.1	91.8
1601 - 1800	0.375	1.415	288.6	68.5
1801 - 2000	0.280	1.110	224.6	53.1
2001 - 2200	0.360	1.240	257.3	48.8
2201 - 2400	0.355	0.925	193.9	31.3
2401 - 2600	0.385	0.655	122.8	15.2
2601 - 2800	0.320	0.565	68.0	8.8
2801 - 3000	0.280	0.420	39.3	5.6
3001 - 3200	0.295	0.495	31.6	6.3
3201 - 3400	0.265	0.680	45.0	11.7
3401 - 3600	0.350	0.750	37.0	13.3
3601 - 3800	0.310	0.465	65.2	8.2
3801 - 4000	0.275	0.520	33.6	7.7
4001 - 4200	0.330	0.480	34.6	5.2
4201 - 4400	0.325	0.615	29.5	7.5
4401 - 4600	0.370	0.525	38.6	7.6
4601 - 4800	0.260	0.705	44.2	15.9
4801 - 5000	0.375	0.720	63.5	11.1
5001 - 5200	0.350	0.635	41.7	9.0
5201 - 5400	0.285	0.475	29.3	6.6
5401 - 5600	0.315	0.510	36.4	4.9
5601 - 5800	0.290	0.425	24.1	4.1
5801 - 6000	0.305	0.490	28.7	4.7

slot sustained for 100 time slots is enough to cripple the channel indefinitely. In Tables IV and V, we show by simulations that under similar but more severe pulse overload circumstances both the IRCP-CONTEST algorithm and heuristic RCP prevented the channel from going into saturation. In these simulations, the normal channel load line was given by $M = 400$ and $(n_o,S_o) = (4,0.32)$ both before and after the pulse. During a period of 200 slots (namely the time period 1000–1200 shown in the tables) the packet generation probability σ was increased such that $M\sigma = 1$ packet/slot. Observe that both algorithms handled the sudden influx of new packets with ease. In both cases, the channel throughput, instead of vanishing to zero as in an uncontrolled channel, maintained at a high rate and within less than 3000 slots, the channel returned to almost normal operation.

D. Discussions of Results

In an actual system, the channel user population and their transmission requirements will typically fluctuate with time. We must emphasize the fact that the control algorithms considered have been designed to control statistical channel fluctuations under the assumption of a stationary channel input. Although we showed that they can temporarily handle very high channel input rates, additional control mechanisms should be designed into the system to make sure that channel overload conditions do not prevail for long periods of time (e.g., by limiting the maximum number of users who can "sign-on" and become active channel users).

We showed earlier that IRCP gives a channel performance at least as good as ICP and RCP. Comparing IRCP-CONTEST and heuristic RCP, we see that the latter is easier to implement. However, under a normal load (say $S_o \leq 0.32$), IRCP-CONTEST is superior to heuristic RCP. This is because heuristic RCP introduces longer delays to collided packets even when these packets are merely unlucky in light channel traffic. On the other hand, with IRCP, control actions are not exerted until the channel traffic exceeds certain "dangerous" levels.

VII. CONCLUSIONS

In this paper, a Markovian decision model has been formulated for the dynamic control of slotted ALOHA random access channels. It is shown that optimal stationary policies exist. Furthermore, an optimal stationary control policy maximizes the stationary channel throughput rate and minimizes the average packet delay simultaneously. An efficient computational algorithm (POLITE) is developed which utilizes Howard's policy-iteration method and is capable of solving for an optimal stationary policy in a small number of computational steps. Numerical results for the control procedures ICP, RCP, and IRCP indicate that optimal control policies are of the control limit type, but a rigorous mathematical proof of this result in general remains an open problem. Throughput-delay tradeoffs given by optimal control policies are also presented. These throughput-delay results are very close to the optimum performance envelope in [4] and are achievable over an infinite time horizon for originally unstable channels. Since in a practical system the exact instantaneous channel state is not known but must be estimated, channel control-estimation (CONTEST) algorithms based upon a Poisson channel traffic estimate are proposed. A heuristic retransmission control algorithm is also suggested. Simulations indicate that these control algorithms are capable of achieving a channel throughput-delay performance close to the theoretical optimum, as well as capable of preventing channel saturation under temporary overload conditions.

The problem of unstable behavior is very real in random access systems (e.g., ALOHA [2], slotted ALOHA, carrier sense multiaccess [16], etc.). To guarantee an acceptable level of channel performance for such systems, some form of dynamic channel control is a must. The probabilistic model and dynamic channel control schemes introduced herein for a slotted ALOHA channel can probably be extended to study stability and dynamic control problems of other random access systems.

REFERENCES

[1] L. Kleinrock and S. S. Lam, "Packet switching in a multiaccess broadcast channel: Performance evaluation," *IEEE Trans. Commun.*, vol. COM-23, pp. 410–423, Apr. 1975.

[2] N. Abramson, "The ALOHA System—Another alternative for computer communications," in *1970 Fall Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 37. Montvale, N. J.: AFIPS Press, 1970, pp. 281–285.

[3] ——, "Packet switching with satellites," in *1973 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, N. J.: AFIPS Press, 1973, pp. 695–702.

[4] L. Kleinrock and S. S. Lam, "Packet-switching in a slotted satellite channel," in *1973 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, N. J.: AFIPS Press, 1973, pp. 703–710.

[5] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *1973 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, N. J.: AFIPS Press, 1973, pp. 711–716.

[6] S. S. Lam, "Packet switching in a multi-access broadcast channel with application to satellite communication in a computer network," Ph.D. dissertation, Comput. Sci. Dep., Univ. of California, Los Angeles, Mar. 1974.

[7] R. M. Metcalfe, "Packet communication," Massachusetts Inst. Technol., Cambridge, Project MAC Tech. Rep. TR-114, Dec. 1973.

[8] L. Kleinrock and S. S. Lam, "On stability of packet switching in a random multi-access broadcast channel," in *Proc. 7th Hawaii Int. Conf. Syst. Sci. (Special Subconf. Comput. Nets)*, Univ. of Hawaii, Honolulu, Jan. 1974.

[9] S. S. Lam and L. Kleinrock, "Dynamic control schemes for a packet switched multi-access broadcast channel," in *1975 Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, N. J.: AFIPS Press, 1975, pp. 143–153.

[10] ——, "Packet switching in a multi-access broadcast channel: Dynamic control procedures," T. J. Watson Res. Center, IBM Corp., Yorktown Heights, N. Y., Res. Rep. RC-5062, Oct. 1974.

[11] R. Howard, *Dynamic Programming and Markov Processes.* Cambridge, Mass.: M.I.T. Press, 1960.

[12] ——, *Dynamic Probabilistic Systems, Vol. 1: Markov Models, Vol. 2: Semi-Markov and Decision Processes.* New York: Wiley, 1971.

[13] S. M. Ross, *Applied Probability Models with Optimization Applications.* San Francisco: Holden-Day, 1970.

[14] L. Kleinrock, *Queueing Systems, Vol. I, Theory.* New York: Wiley Interscience, 1975, p. 17.

[15] E. J. Craig, *Laplace and Fourier Transforms for Electrical Engineers.* New York: Holt, Rinehart and Winston, 1964.

[16] L. Kleinrock and F. A. Tobagi, "Carrier sense multiple access for packet switched radio channels," in *Proc. Int. Conf. Commun.*, Minneapolis, Minn., 1974.

★

Simon S. Lam (S'69–M'74), for a photograph and biography see page 423 of the April 1975 issue of this TRANSACTIONS.

★

Leonard Kleinrock (S'55–M'64–SM'71–F'73), for a photograph and biography see page 423 of the April 1975 issue of this TRANSACTIONS.