

Dynamic control schemes for a packet switched multi-access broadcast channel*

by SIMON S. LAM

*IBM Thomas J. Watson Research Center
Yorktown Heights, New York*

and

LEONARD KLEINROCK

*University of California
Los Angeles, California*

INTRODUCTION

Domestic satellites are emerging as an exciting alternative to satisfying the communications requirements of data users, providing both flexibility and economy. Two attributes of satellites are especially advantageous for the transmission of data in large geographically distributed computer networks. They are (i) the availability of wide transmission bandwidths over long distances and (ii) the multi-access broadcast capability inherent in radio communications which permits transmission to, and reception from, all points in a satellite connected network. These considerations also apply (on a smaller geographical scale) to the use of ground radio channels in a terminal access computer-communication network exemplified by the ALOHA System at the University of Hawaii.¹

The random access scheme of the ALOHA System has inspired a number of packet switching techniques which permit the sharing of a high-speed multi-access broadcast channel by a large population of channel users.²⁻⁸ Such packet switched radio systems (both satellite and ground radio) have a number of advantages over conventional wire communication techniques for computer communications, such as: the elimination of complex topological design and routing problems in large networks, the possibility of mobile users, the cost reduction over long distances and the increased flexibility for system reconfiguration and upgrading. Another attractive feature is that in these systems each user is merely represented by an ID number. Thus, the number of active users is bounded only by the channel capacity and there is no limitation to the number of inactive (but potentially active) users beyond that of a finite address space. Moreover, measurement studies have shown that interactive computer data traffic tends to be bursty.⁹ A single high-speed radio channel permits the total demand of a large population of bursty users to be statistically averaged at the channel. Furthermore, each user transmits data at the full wideband

data rate of the radio channel. Such efficient sharing and wideband transmission are in general not possible in a geographically distributed computer-communication network using wire communications.

Of interest in this paper is the slotted ALOHA random access scheme.^{3,4,7,10-13} A slotted ALOHA channel multi-accessed by a large number of users has been shown to exhibit unstable behavior, i.e., the system may drift into an undesirable saturation state with a virtually zero probability of transmission success as a result of repeated user conflicts.^{4,7,10-12,14} In this paper, a model is first presented for a slotted ALOHA channel supporting input from a large population of bursty users; the data rate of each channel user is assumed to be much less than the channel transmission rate. The underlying concepts of channel stability are then introduced. A dynamic channel control model is next presented and four dynamic channel control algorithms are given. The performance of these algorithms are tested through simulation and compared to analytic results previously obtained.^{7,13} We conclude that these algorithms are capable of preventing the occurrence of channel saturation under temporary channel overload conditions and at the same time achieving a level of channel performance close to the theoretical optimum.

The slotted ALOHA model here is similar to one previously studied by Metcalfe through a steady-state analysis.^{10,14} He has also recognized the need for control of the channel and proposed a method for controlling the transmission probability of "ready" packets.

Other multi-access broadcast packet switching schemes have been proposed to take advantage of special system and traffic characteristics. A reservation scheme studied by Roberts⁵ employs a slotted ALOHA subchannel for broadcasting block transfer reservation requests. Reservation-ALOHA² and carrier sense multi-access³ are both interesting variants of the random access scheme. These systems seem to exhibit unstable behavior similar to that of slotted ALOHA and may be dynamically controlled by algorithms similar to those presented in this paper. Consider, for instance, the ALOHA System at the University of Hawaii which uses two 24 KBPS radio channels and which has been

* This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-73-C-0368.

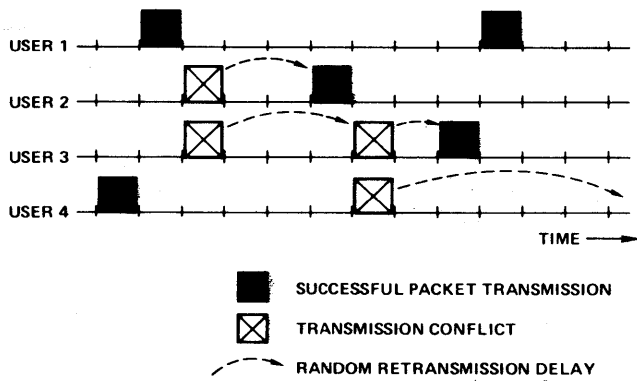


Figure 1—Slotted ALOHA random access

estimated to be able to support up to 500 interactive users.^{1,3} We feel that this figure is unrealistic for an uncontrolled system, but may be achieved given some appropriate dynamic channel control.

THE RANDOM ACCESS CHANNEL MODEL

Consider a radio communication system such as a packet switching satellite system³⁻⁷ or the ALOHA System.¹ In each case, there is a *broadcast* channel for point-to-multipoint communication and a *multi-access* channel shared by a large number of users. Since the broadcast channel is used by a single transmitter, no transmission conflict will arise. All nodes covered by the radio broadcast can receive on the same single frequency, picking out packet transmissions addressed to themselves and discarding packets addressed to others. The problem we are faced with is how to resolve conflicts which arise when "simultaneous" demands are placed upon the multi-access channel. If two or more packet transmissions overlap in time at the multi-accessed radio receiver (of the satellite transponder or the central computer), it is assumed that none is received correctly. This event will be referred to as a *channel collision*. The channel may be *slotted* by requiring all channel users to synchronize the leading edges of their packet transmissions at the multi-accessed radio receiver.³⁻⁷ The duration of a channel time slot is set equal to a packet transmission time. In the slotted ALOHA random access scheme, all users transmit newly generated packets into channel time slots independently. In the event of a channel collision, each collided packet is retransmitted independently after a *retransmission delay* of RD slots. The above scheme is illustrated in Figure 1 for the case of a channel random-accessed by four users. (In a ground radio system, RD corresponds to the positive acknowledgment time-out interval.)

Consider a satellite multi-access broadcast system. Let R be the number of time slots in a round-trip satellite channel propagation time which is assumed to be the same for all earth stations. Thus, R time slots after transmitting a packet, a user will either hear that he was successful or know that he had a channel collision. (We have ignored the possibility of random noise errors assuming that the channel has a low

error rate.) The retransmission delay RD for a collided packet must be greater than R . Randomization of RD is necessary to minimize the probability of repeated channel collisions for the same packets. Randomization schemes which have been considered include: (1) the uniform retransmission randomization scheme⁴ in which the probability distribution of RD is given by

$$\text{Prob [RD=i]} = \begin{cases} 0 & i \leq R \\ 1/K & R+1 \leq i \leq R+K \\ 0 & i > R+K \end{cases} \quad (1)$$

and (2) the geometric retransmission randomization scheme^{6,7,10-14} in which the probability distribution of RD is given by

$$\text{Prob [RD=i]} = \begin{cases} 0 & i \leq R \\ p(1-p)^{i-R-1} & i > R \end{cases} \quad (2)$$

The uniform retransmission randomization scheme is adopted in Reference 4. In that reference, R is taken to be 12 and each time slot is 22.5 milliseconds long, giving 44.4 slots/second. These figures are computed from the assumptions of a 50 KBPS satellite voice channel, 1125 bits/packet and a roundtrip channel propagation time of 0.27 second for all channel users. These same numerical constants are adopted in this paper. However, to study the problems of stability and dynamic channel control, it is necessary to consider a simplified Markovian model in which $R=0$ and the geometric retransmission randomization scheme is assumed, such that RD has a memoryless geometric distribution.^{7,10-14} Simulation results have shown that the slotted ALOHA channel performance (in terms of average throughput and delay) is dependent primarily upon the average retransmission delay $\overline{\text{RD}}$ and quite insensitive to the exact probability distributions considered.⁷ In order to use the analytic results of the Markovian model to predict the throughput-delay performance of a real slotted ALOHA channel with nonzero R , it is necessary to use a value of p in the Markovian model which matches the value of $\overline{\text{RD}}$. For example, to approximate the slotted ALOHA channel with uniform retransmission randomization and for which $\overline{\text{RD}} = R + (K+1)/2$, we must let

$$p = \frac{1}{R + (K+1)/2} \quad (3)$$

such that $\overline{\text{RD}}$ is the same in both cases. Numerical results in this paper will always be expressed in terms of K (rather than p) through use of Equation (3).

Let us now introduce the Markovian model,^{7,11-13} in which 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*.^{10,14} 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 the *channel backlog*) representing the total number of backlogged packets at time t . The "channel input" rate at time t is $S^t = (M - N^t)\sigma$. We shall assume M and σ to be time-invariant unless stated otherwise. In this case, N^t is a Markov process (chain) with stationary transition probabilities and serves as the state description for the system. The discrete *state space* consists of the set of integers $\{0, 1, 2, \dots, M\}$.

CHANNEL STABILITY

In this section, we give a brief description of the stability behavior of an uncontrolled slotted ALOHA system studied earlier.^{7,10-12,14} Consider the trajectory of (N^t, S^t) in the two-dimensional (n, S) plane. Assuming that M and σ are constant, (N^t, S^t) is constrained to lie on the straight line $S = (M - n)\sigma$ called the *channel load line*. Corresponding to a fixed value* of K , 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

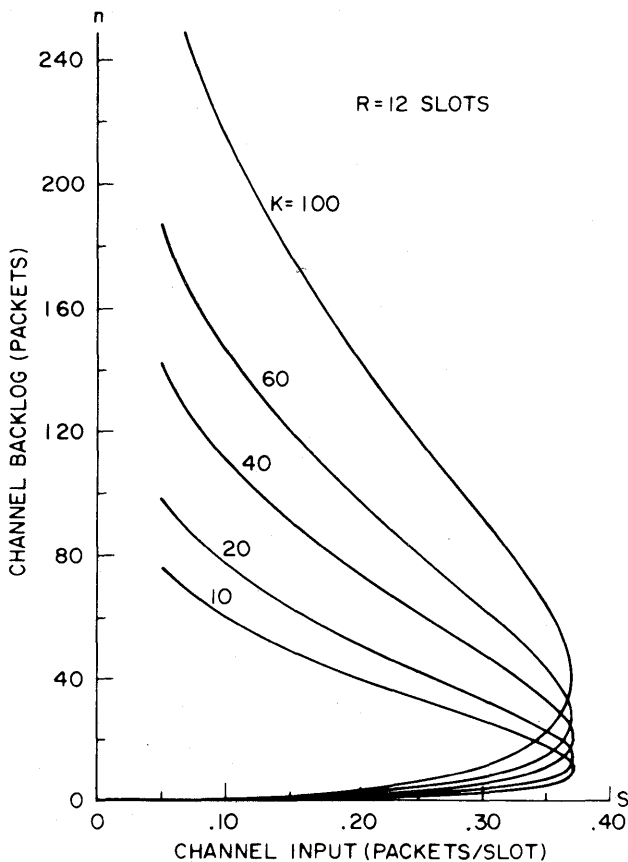


Figure 2—Equilibrium contours

* Or equivalently a fixed value of p under Equation (3).

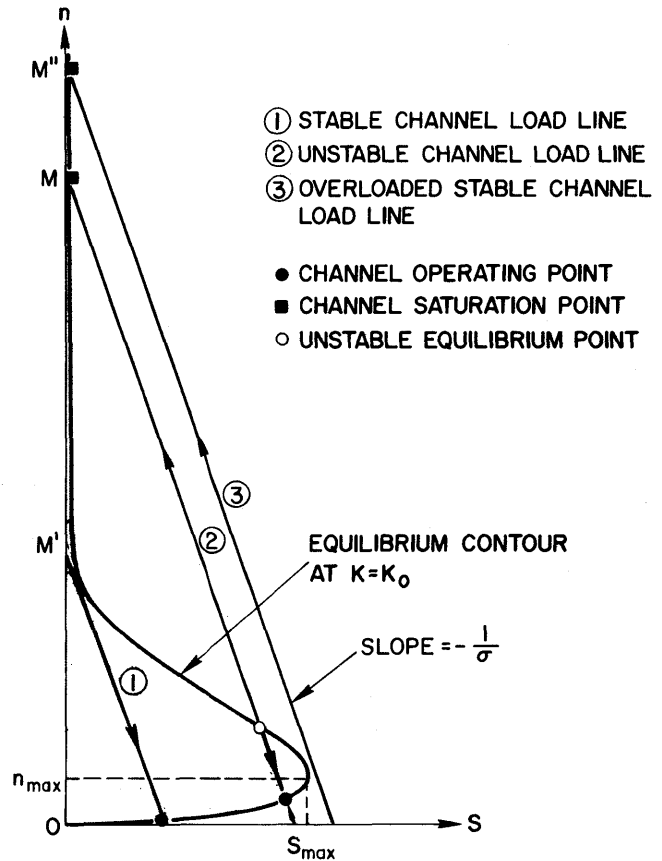


Figure 3—Stable and unstable channels

throughput (defined to be the probability of a successful packet transmission) $S_{out}(n, S)$ in a time slot. A family of such contours is illustrated in Figure 2. Let us focus upon an equilibrium contour corresponding to $K = K_0$ in Figure 3. In the shaded region enclosed by the equilibrium contour, $S_{out}(n, S)$ is greater than S ; elsewhere, S exceeds $S_{out}(n, S)$. Arrows on the channel load lines point in the direction of "drift" of the channel backlog size N^t . Three channel load lines are also shown in Figure 3 corresponding to channel user population sizes M, M' and M'' , and an average user think time of $1/\sigma$ slots.

A channel load line may intersect the equilibrium contour one or more times, and we refer to these as equilibrium points which we denote 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 ; an equilibrium point is said to be an *unstable equilibrium point* if it acts as a "source." A stable equilibrium point is said to be the *channel operating point* if $n_e \leq n_{max}$ as shown in Figure 3; it is said to be the *channel saturation point* if $n_e > n_{max}$. (We shall use (n_o, S_o) instead of (n_e, S_e) to distinguish the channel operating point from other equilibrium points.) A channel load line is defined to be *stable* if it has exactly one stable equilibrium point; otherwise it is defined to be *unstable*. Thus, the load lines 1 and 3 in Figure 3 are stable by definition; the load line 2 is unstable.

INPUT PARAMETERS => => =>
 INPUT RATE = 0.350
 PROPAGATION DELAY = 12
 K = 15

AVERAGE VALUES IN 200 TIME SLOT PERIODS

| TIME PERIOD | THROUGHPUT RATE- S | TRAFFIC RATE- G | PACKET DELAY- D | FRACTION EMPTY | AVERAGE BACKLOG |
|-------------|--------------------|-----------------|-----------------|----------------|-----------------|
| 1 - 200 | 0.330 | 0.510 | 17.924 | 0.595 | 3.1 |
| 201 - 400 | 0.370 | 0.605 | 30.892 | 0.530 | 5.3 |
| 401 - 600 | 0.360 | 0.860 | 32.764 | 0.425 | 9.5 |
| 601 - 800 | 0.340 | 0.840 | 43.147 | 0.435 | 10.5 |
| 801 - 1000 | 0.315 | 1.415 | 58.889 | 0.250 | 21.5 |
| 1001 - 1200 | 0.380 | 1.260 | 72.066 | 0.260 | 18.0 |
| 1201 - 1400 | 0.325 | 0.455 | 37.215 | 0.610 | 3.0 |
| 1401 - 1600 | 0.355 | 0.480 | 20.803 | 0.590 | 2.5 |
| 1601 - 1800 | 0.275 | 0.405 | 20.600 | 0.665 | 2.6 |
| 1801 - 2000 | 0.360 | 0.560 | 27.528 | 0.550 | 4.5 |
| 2001 - 2200 | 0.330 | 0.430 | 18.561 | 0.620 | 1.8 |
| 2201 - 2400 | 0.310 | 0.545 | 22.065 | 0.580 | 4.3 |
| 2401 - 2600 | 0.335 | 0.840 | 44.866 | 0.455 | 9.9 |
| 2601 - 2800 | 0.320 | 0.705 | 34.703 | 0.500 | 7.5 |
| 2801 - 3000 | 0.325 | 1.085 | 43.815 | 0.350 | 14.1 |
| 3001 - 3200 | 0.310 | 1.715 | 67.161 | 0.180 | 26.9 |
| 3201 - 3400 | 0.105 | 3.255 | 147.143 | 0.050 | 61.6 |
| 3401 - 3600 | 0.015 | 5.910 | 220.333 | 0.000 | 113.4 |
| 3601 - 3800 | 0.000 | 9.155 | 0.000 | 0.000 | 179.3 |
| 3801 - 4000 | 0.000 | 12.400 | 0.000 | 0.000 | 243.8 |

Figure 4—An unstable channel drifting into saturation

If M is finite, a stationary probability distribution always exists for N^t . In a stable channel, the equilibrium point (n_o, S_o) gives (approximately) 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 toward the channel saturation point. When this happens, the channel performance degrades rapidly as the channel throughput rate decreases and the average packet delay increases. In this state, the communication channel can be regarded as having failed. (In a practical system, external control should be applied at this point to restore proper channel operation.) In Figure 4, we have shown a simulation of the above behavior. In this example, M is assumed to be so large that the channel input is Poisson distributed at a constant rate $S=0.35$.

The channel load line labelled 3 in Figure 3 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.

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. In Refer-

ences 7, 11 and 12, the average "up" time of an unstable channel has been quantified as a stability measure of the channel. To render the channel stable, two obvious solutions are available: (1) use a larger value for K (see Figure 2), and (2) reduce the user population size M . The first solution gives rise to a smaller S_o and a larger n_o ; the corresponding average packet delay may then be too large to be acceptable. In the second solution, a small M implies that $S_o \ll S_{max}$ (see Figure 3) since $\sigma \ll 1$ under the assumption of bursty users. This results in a waste of channel capacity.

The third solution is the use of dynamic channel control which constitutes the subject matter of the balance of this paper.

THE DYNAMIC CHANNEL CONTROL MODEL

To prevent the disastrous consequences of channel saturation, various dynamic control measures may be taken. In this section, we describe the dynamic channel control model studied in References 7 and 13, and outline some of the results obtained there under the assumption of *perfect channel state information*, i.e., each channel user knows the exact value of the channel backlog N^t at time t . In the next section, we shall consider practical control schemes which estimate the channel state and apply the theoretical optimal control policies using this estimate.

Consider the finite-state Markovian decision model obtained by injecting the following two classes of control actions into our earlier model for N^t :

- (i) each packet arrival is accepted for transmission with probability β and rejected with probability $1-\beta$ where $0 \leq \beta \leq 1$ and $\beta \in \{\beta_1, \beta_2, \dots, \beta_m\} \triangleq \mathcal{A}_1$;
- (ii) each backlogged packet is retransmitted with probability γ where $0 < \gamma < 1$ and $\gamma \in \{\gamma_1, \gamma_2, \dots, \gamma_k\} \triangleq \mathcal{A}_2$.

$\mathcal{A} \triangleq \mathcal{A}_1 \times \mathcal{A}_2$ is said to be the control action space. Three special cases have been studied extensively in References 7 and 13, namely,

- (1) The Input Control Procedure (ICP) with $\mathcal{A} = \{0, 1\} \times \{p_o\}$,
- (2) The Retransmission Control Procedure (RCP) with $\mathcal{A} = \{1\} \times \{p_o, p_c\}$, and
- (3) The Input-Retransmission Control Procedure (IRCP) with $\mathcal{A} = \{0, 1\} \times \{p_o, p_c\}$.

In these control procedures, p_o corresponds to some K_o which optimizes the channel operating point of the given channel load line; p_c corresponds to some K_c which is sufficiently large to render the given channel load line stable.

A control policy f is defined to be any rule for choosing control actions in \mathcal{A} . The action α^t , at time t given by the policy f , specifies both the state transition probabilities and some predefined expected state transition cost for the t th time slot. Thus f determines both the evolution in time of N^t and the sequence of costs it incurs. Given a cost structure (denoted by δ), the cost rate $g_s(f)$ of N^t under a control policy f is defined to be the steady-state average cost per unit time incurred by N^t .

An important subclass of all policies is the class of stationary policies. A stationary policy is defined to be one which chooses an action at time t depending only upon the state of the process at that time. From well-known results in Markov decision theory, we know that (1) if f is a stationary policy, $g_s(f)$ is independent of the initial state of the process N^t , and (2) a stationary policy f^* exists, which minimizes $g_s(f)$ over the class of all policies. Thus in our search for an optimal control strategy, we can limit our attention to the class of stationary policies only.

As the process N^t evolves from one time slot to the next, various expected state transition costs may be incurred, such as

- (1) the expected channel throughput in the t th time slot,
- (2) the (delay) cost of holding backlogged packets, and
- (3) the expected (delay) cost of rejecting packet arrivals.

Type 1 costs take on negative values since we want to maximize the channel throughput rate. Type 2 costs are chosen such that each backlogged packet incurs 1 unit of delay per time slot. In the references, the expected cost in units of delay per packet arrival rejected (type 3 costs) is assumed

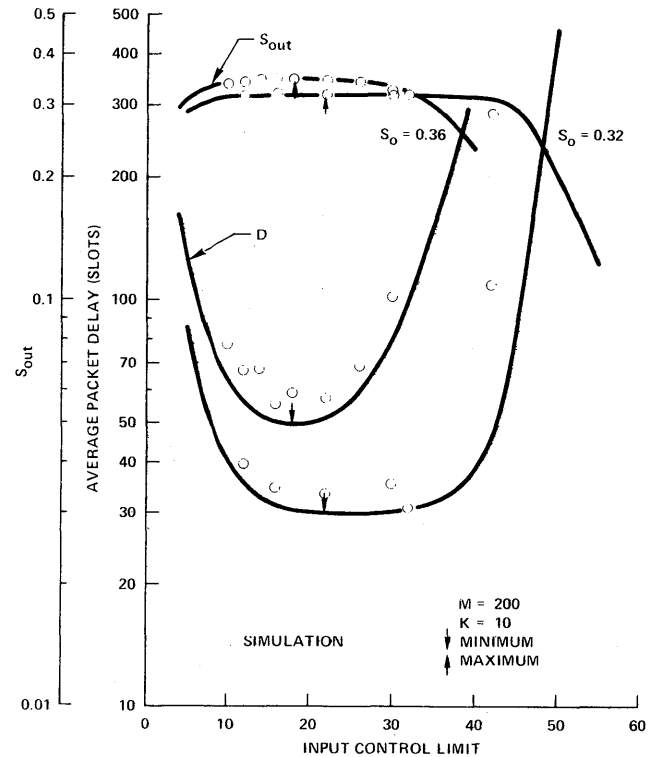


Figure 5—Channel performance versus ICP control limit for $M=200$

to be equal to an average user think time. This assumption is needed for our Markovian model formulation and may be justified in a terminal access communications environment as follows. A person sitting at a terminal generates a new packet with an average think time of $1/\sigma$ whenever his previous packet has been successfully transmitted. If, at the time of a packet arrival, the channel is in the reject state, this packet is lost in the sense that it is not retransmitted over the channel immediately. In a practical situation, the user may be informed of the event and must enter some command character to "resend" the packet. Hence, the cost in terms of delay is probably in the order of an average think time ($=1/\sigma$).

Let $g_s(f)$ denote the cost rate of N^t given by policy f and type 1 costs, and $g_d(f)$ denote the cost rate of N^t given by policy f and types 2 and 3 costs. The channel performance measures, namely, the steady-state channel throughput rate S_{out} and the expected packet delay D can then be calculated in terms of $g_s(f)$ and $g_d(f)$.

In the references, it is shown that for the given model an optimal stationary control policy maximizes S_{out} and minimizes D simultaneously. An efficient computational algorithm (POLITE) based upon Howard's policy-iteration method¹⁵ is given for calculating the optimal policy. Given a channel load line and a dynamic control procedure (\mathcal{A}), this algorithm usually arrives at the optimal control policy and the optimum values of S_{out} and D in very few iterations. Furthermore, numerical results indicate that each optimal

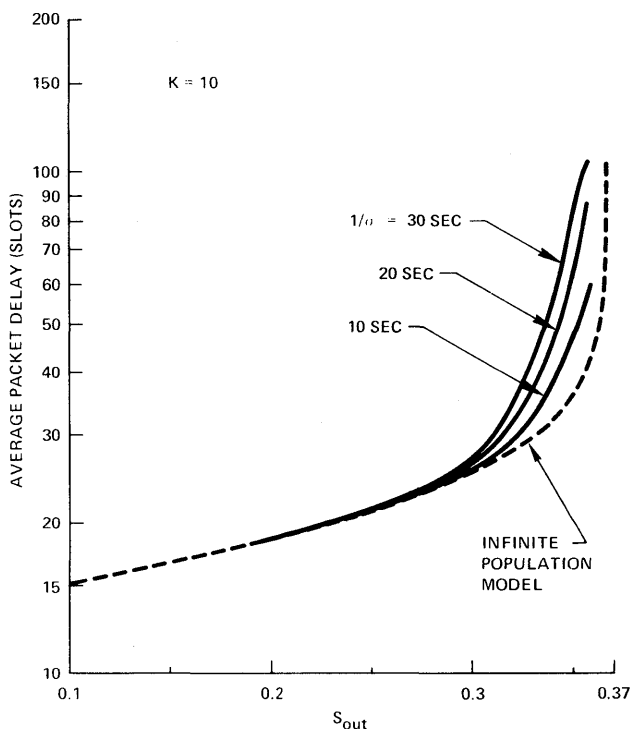


Figure 6—ICP optimum throughput-delay tradeoffs at fixed σ

control policy f for the control procedures ICP and RCP has the following structure:

$$f(i) = \begin{cases} a_o & 0 \leq i \leq \hat{n} \\ a_c & \hat{n} < i \leq M \end{cases} \quad (4)$$

where a_o corresponds to “accept” in ICP and “ p_o ” in RCP; a_c corresponds to “reject” in ICP and “ p_c ” in RCP. On the other hand, an optimal control policy f for IRCP has the following structure:

$$f(i) = \begin{cases} (\text{accept}, p_o) & 0 \leq i \leq \hat{n}_1 \\ (\text{accept}, p_c) & \hat{n}_1 < i \leq \hat{n}_2 \\ (\text{reject}, p_c) & \hat{n}_2 < i \leq M \end{cases} \quad (5)$$

We shall refer to \hat{n} , \hat{n}_1 and \hat{n}_2 as control limits and the control policies in Equations (4) and (5) as *control limit policies*.

In Figure 5, we have shown the performance measures, S_{out} and D , for two channel load lines specified by $M=200$ and the channel operating point $(n_o, S_o) = (4, 0.32)$ and $(7, 0.36)$, over a range of ICP control limit policies. Observe that the same control limit minimizes D and maximizes S_{out} at the same time as predicted by the theory. Note the amazing flatness of S_{out} and D near the optimum point for the channel load line with $S_o=0.32$. The consequence is that even if a nonoptimal control policy is used (due, for example, to not knowing the exact current backlog size such as in most practical systems), it is still possible to achieve a throughput-delay performance close to the optimum.

In Figure 5, we have also shown simulation results for throughput and delay. In these simulations, channel control

policies are applied assuming that the exact 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 R and its retransmission is randomized uniformly over the next K slots. The excellent agreement between the simulation and analytic results presented here demonstrates the accuracy of the approximation.

In Figure 6, we show optimum throughput-delay tradeoffs at fixed values of σ for ICP. ($1/\sigma$ is the average think time of a channel user.) In this case, increasing S_{out} corresponds to increasing M , that is, admitting more channel users. We see 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 M is smaller. In the latter case, the channel is “less unstable.”^{7,11,12}

PRACTICAL CONTROL SCHEMES

In a practical system, the channel users often have no means of communication among themselves other than the multi-access broadcast channel itself. Each channel user must individually estimate the channel state by observing the outcome in each channel slot. Moreover, whatever channel state information available to the channel users is at least one round-trip propagation delay (R) old and 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 then will lead to some degradation in channel performance.

Below we first give a heuristic scheme for estimating the channel state assuming that the channel history (i.e., empty slots, successful transmissions or collisions) is available to all channel users. The optimal ICP, RCP and IRCP control policies will be applied based upon the above 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 optimum throughput-delay results in the previous section. The ability of these control procedures to handle time-varying inputs (with pulses) is also examined.

Channel control-estimation (CONTEST) 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 procedure for estimating the channel state is based upon the observation that the channel traffic in a time slot is approximately Poisson distributed. (See Chapter 4 and Appendix A of Reference 7.) Below we present algorithms which implement channel control procedures studied in the previous sections using estimates of the channel state. These channel CONTROL-ESTIMATION algorithms will be referred to as CONTEST algorithms.

Here we give a procedure for implementing RCP. Suppose \hat{n} is the RCP control limit such that the channel users switch

their retransmission K value from K_o to K_c when the channel backlog size exceeds \hat{n} and from K_c to K_o as soon as the channel backlog size drops below \hat{n} . We define

$$\hat{G}_o = \hat{n}p_o + (M - \hat{n})\sigma \quad (6)$$

and

$$\hat{G}_c = \hat{n}p_c + (M - \hat{n})\sigma \quad (7)$$

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

$$\hat{f}_o = e^{-\hat{G}_o} \quad (8)$$

and

$$\hat{f}_c = e^{-\hat{G}_c} \quad (9)$$

Since $K_c > K_o$, we must have

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

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

Algorithm 1 (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).

- (1) $t \leftarrow t + 1$
 $d^t = K_o$
- (2) If $\bar{f}^t < \hat{f}_o$, go to (4)
- (3) Go to (1)
- (4) $t \leftarrow t + 1$
 $d^t = K_c$
- (5) If $\bar{f}^t > \hat{f}_c$, go to (1)
- (6) Go to (4)

Next we consider a similar implementation for ICP. We define

$$\hat{G}_a = \hat{n}p + (M - \hat{n})\sigma \quad (10)$$

$$\hat{G}_r = \hat{n}p \quad (11)$$

$$\hat{f}_a = e^{-\hat{G}_a} \quad (12)$$

and

$$\hat{f}_r = e^{-\hat{G}_r} \quad (13)$$

Since $\hat{G}_a > \hat{G}_r$, we must have

$$\hat{f}_a < \hat{f}_r$$

TABLE I—Throughput-delay Results of a Controlled Channel
($M=200, S_o=0.32$)

| CONTROL SCHEME | S_{out} | D |
|--|-----------|--------|
| ICP (POLITE) | 0.31778 | 29.857 |
| RCP (POLITE) | 0.31817 | 29.085 |
| IRCP(POLITE) | 0.31817 | 29.085 |
| ICP (Simulation) | 0.315 | 33.427 |
| RCP (Simulation) | 0.318 | 28.824 |
| ICP-CONTEST W = 20 | 0.314 | 40.893 |
| ICP-CONTEST W = 40 | 0.315 | 30.514 |
| ICP-CONTEST W = 60 | 0.317 | 32.355 |
| ICP-CONTEST W = 80 | 0.318 | 35.809 |
| RCP-CONTEST W = 20 | 0.315 | 33.052 |
| RCP-CONTEST W = 40 | 0.322 | 33.335 |
| RCP-CONTEST W = 60 | 0.319 | 32.138 |
| RCP-CONTEST W = 80 | 0.317 | 32.501 |
| Heuristic RCP $\begin{cases} k_1 = 10 \\ k_m = 60 \quad m \geq 2 \end{cases}$ | 0.316 | 33.720 |
| | 0.315 | 34.554 |
| Heuristic RCP $\begin{cases} k_1 = 10 \\ k_2 = 60 \\ k_m = 120 \quad m \geq 3 \end{cases}$ | 0.310 | 35.425 |
| | 0.316 | 34.635 |

Algorithm 2 (ICP-CONTEST)—This algorithm generates the decision $d^t = \text{accept, reject}$ at time t , based upon the channel state estimate \bar{f}^t and ICP control limit \hat{n} . Start at step (1) or step (4).

- (1) $t \leftarrow t + 1$
 $d^t = \text{accept}$
- (2) If $\bar{f}^t < \hat{f}_a$, go to (4)
- (3) Go to (1)
- (4) $t \leftarrow t + 1$
 $d^t = \text{reject}$
- (5) If $\bar{f}^t > \hat{f}_r$, go to (1)
- (6) Go to (4)

Finally, to implement IRCP, we assume that the control policy is of the form given in Equation (5) such that it is uniquely specified by the control limits \hat{n}_1 and \hat{n}_2 . We define \hat{f}_o and \hat{f}_c by using \hat{n}_1 in Equations (6)-(9), \hat{f}_{ac} and \hat{f}_{rc} by using \hat{n}_2 and p_c in Equations (10)-(13) and \hat{f}_{ao} by using \hat{n}_2 and p_o in Equations (10) and (12). Since $p_o > p_c > \sigma$ and $\hat{n}_2 > \hat{n}_1$, we have $\hat{f}_{ao} < \hat{f}_o$ and $\hat{f}_{rc} < \hat{f}_c$.

Algorithm 3 (IRCP-CONTEST)—This algorithm generates the decision $d^t = (\text{accept}, K_o), (\text{accept}, K_c), (\text{reject}, K_c)$

TABLE II—Throughput-delay Results of a Controlled Channel
($M=400, S_o=0.32$)

| CONTROL SCHEME | S_{out} | D |
|---|-----------|--------|
| ICP (POLITE) | 0.31807 | 33.096 |
| RCP (POLITE) | 0.31844 | 31.608 |
| IRCP (POLITE) | 0.31844 | 31.608 |
| ICP (Simulation) | 0.315 | 31.427 |
| RCP (Simulation) | 0.317 | 31.023 |
| ICP-CONTEST $W = 20$ | 0.315 | 43.262 |
| ICP-CONTEST $W = 40$ | 0.314 | 34.723 |
| ICP-CONTEST $W = 60$ | 0.312 | 53.240 |
| ICP-CONTEST $W = 80$ | 0.316 | 39.112 |
| RCP-CONTEST $W = 20$ | 0.313 | 41.087 |
| RCP-CONTEST $W = 40$ | 0.319 | 43.379 |
| RCP-CONTEST $W = 60$ | 0.318 | 38.821 |
| RCP-CONTEST $W = 80$ | 0.317 | 40.068 |
| RCP-CONTEST $W = 100$ | 0.314 | 35.689 |
| RCP-CONTEST $W = 120$ | 0.319 | 47.149 |
| Heuristic RCP $\begin{cases} K_1 = 10 \\ K_m = 150 \quad m \geq 2 \end{cases}$ | 0.316 | 45.150 |
| Heuristic RCP $\begin{cases} K_1 = 10 \\ K_2 = 100 \\ K_m = 200 \quad m \geq 3 \end{cases}$ | 0.312 | 42.040 |
| | 0.311 | 43.136 |

at time t based upon the channel state estimate \bar{f}^t and IRCP control policy (\hat{n}_1, \hat{n}_2). Start at step (1), (4), or (7).

- (1) $t \leftarrow t+1$
 $d^t = (\text{accept}, K_o)$
- (2) If $\bar{f}^t < \hat{f}_{ao}$ go to (7)
otherwise, if $\bar{f}^t < \hat{f}_o$ go to (4)
- (3) go to (1)
- (4) $t \leftarrow t+1$
 $d^t = (\text{accept}, K_o)$
- (5) If $\bar{f}^t > \hat{f}_o$ go to (1)
otherwise, if $\bar{f}^t < \hat{f}_{ac}$ go to (7)
- (6) go to (4)
- (7) $t \leftarrow t+1$
 $d^t = (\text{reject}, K_o)$
- (8) If $\bar{f}^t > \hat{f}_{re}$ go to (4)
- (9) go to (7)

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 .

Another retransmission control procedure

In this section we describe a simple heuristic control procedure which has the property that when 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 retransmission randomization* 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 of m , we see that channel saturation will be prevented. An effective value of K can be defined only with respect to a specific performance measure (e.g., average packet delay). To illustrate the effect of the function K_m , we derive below the *average* value of K as a function of q (the probability of successful transmission) for two cases. Let

$$r_i = \text{Prob} [\text{a packet retransmits } i \text{ times before success}]$$

$$= (1-q)^i q \quad i \geq 1$$

Case 1 $K_m = K_2$ for $m \geq 2$ and $K_2 > K_1$

\bar{K} = average value of K

$$\begin{aligned} &= \frac{1}{1-q} \sum_{i=1}^{\infty} r_i \sum_{m=1}^i \frac{K_m}{i} \\ &= \frac{1}{1-q} \sum_{i=1}^{\infty} (1-q)^i q \left(\frac{K_1}{i} + \frac{i-1}{i} K_2 \right) \\ &= K_2 + \frac{q \ln q}{1-q} (K_2 - K_1) \end{aligned} \quad (14)$$

which is equal to K_1 at $q=1$ and increases to K_2 as q decreases to zero; \ln is the natural logarithm function.

Case 2 $K_m = mK$ $m \geq 1$

$$\begin{aligned} \bar{K} &= \frac{1}{1-q} \sum_{i=1}^{\infty} r_i \sum_{m=1}^i \frac{K_m}{i} \\ &= \frac{K}{1-q} \sum_{i=1}^{\infty} \frac{(1-q)^i q}{i} \sum_{m=1}^i m \\ &= \frac{K}{2} \left(1 + \frac{1}{q} \right) \end{aligned} \quad (15)$$

* Note that the same control scheme can be extended to geometric retransmission randomization by letting $p = p_m$ where p_m is a monotone nonincreasing function in m .

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- S | TRAFFIC RATE- G | PACKET DELAY- D | FRACTION EMPTY | AVERAGE BACKLOG | PACKETS REJECTED |
|-------------|-----------------------|--------------------|--------------------|-------------------|--------------------|---------------------|
| 1 - 200 | 0.290 | 0.625 | 30.29 | 0.555 | 5.5 | 0 |
| 201 - 400 | 0.325 | 0.700 | 34.06 | 0.505 | 6.9 | 0 |
| 401 - 600 | 0.285 | 0.450 | 23.67 | 0.635 | 2.9 | 0 |
| 601 - 800 | 0.295 | 0.625 | 31.76 | 0.565 | 5.9 | 0 |
| 801 - 1000 | 0.325 | 0.850 | 42.57 | 0.455 | 9.3 | 0 |
| 1001 - 1200 | 0.205 | 2.345 | 524.32 | 0.190 | 50.0 | 49 |
| 1201 - 1400 | 0.345 | 1.330 | 389.68 | 0.325 | 75.2 | 13 |
| 1401 - 1600 | 0.355 | 0.880 | 189.11 | 0.435 | 51.5 | 0 |
| 1601 - 1800 | 0.375 | 0.735 | 179.37 | 0.460 | 34.9 | 0 |
| 1801 - 2000 | 0.225 | 1.295 | 297.85 | 0.385 | 33.7 | 5 |
| 2001 - 2200 | 0.325 | 1.005 | 530.77 | 0.415 | 35.3 | 21 |
| 2201 - 2400 | 0.380 | 0.905 | 127.32 | 0.365 | 16.7 | 0 |
| 2401 - 2600 | 0.305 | 0.485 | 27.64 | 0.605 | 3.1 | 0 |
| 2601 - 2800 | 0.290 | 0.430 | 20.86 | 0.640 | 2.3 | 0 |
| 2801 - 3000 | 0.345 | 0.745 | 35.38 | 0.485 | 7.2 | 0 |
| 3001 - 3200 | 0.300 | 0.455 | 17.62 | 0.635 | 2.4 | 0 |
| 3201 - 3400 | 0.280 | 0.615 | 28.48 | 0.575 | 5.6 | 0 |
| 3401 - 3600 | 0.390 | 0.810 | 37.90 | 0.425 | 7.9 | 0 |
| 3601 - 3800 | 0.330 | 0.655 | 30.65 | 0.520 | 5.6 | 0 |
| 3801 - 4000 | 0.300 | 0.390 | 19.30 | 0.655 | 1.7 | 0 |
| 4001 - 4200 | 0.315 | 0.615 | 29.24 | 0.560 | 5.1 | 0 |
| 4201 - 4400 | 0.335 | 0.600 | 24.51 | 0.545 | 4.5 | 0 |
| 4401 - 4600 | 0.300 | 0.450 | 24.32 | 0.630 | 2.6 | 0 |
| 4601 - 4800 | 0.280 | 0.480 | 25.29 | 0.625 | 3.7 | 0 |
| 4801 - 5000 | 0.285 | 0.585 | 32.07 | 0.580 | 5.3 | 0 |
| 5001 - 5200 | 0.330 | 0.570 | 26.41 | 0.555 | 4.3 | 0 |
| 5201 - 5400 | 0.335 | 0.550 | 23.67 | 0.560 | 3.7 | 0 |
| 5401 - 5600 | 0.335 | 0.640 | 28.81 | 0.530 | 5.2 | 0 |
| 5601 - 5800 | 0.275 | 0.410 | 21.56 | 0.660 | 2.4 | 0 |
| 5801 - 6000 | 0.285 | 0.445 | 22.35 | 0.645 | 2.7 | 0 |

Figure 7—Simulation run for IRCP-CONTEST subject to a channel input pulse

which is equal to K at $q=1$ and increases to infinity as q decreases to zero.

The above results indicate that the average value of K behaves in the desired manner, namely, K increases as q decreases due to an increasing channel traffic. Below we examine the CONTEST algorithms and Heuristic RCP through simulations.

Simulation results

We summarize in Tables I-II, throughput-delay results for channel load lines specified by

- (1) $M=200$, $(n_o, S_o) = (4, 0.32)$
- (2) $M=400$, $(n_o, S_o) = (4, 0.32)$

In both cases, we assume $K_o=10$ and $K_c=60$. Included in these tables are (a) optimum POLITE results for ICP, RCP and IRCP, (b) simulation results for ICP and RCP using optimal control policies and under the assumption of perfect channel state information, (c) simulation results for the CONTEST algorithm using ICP and RCP optimal control policies, and (d) 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 I and II 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 and K_c . 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 theoretical 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 Figure 5). We found that this flatness property is less pronounced for channel load lines with large values of S_o or M , such as $S_o=0.36$ or

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- S | TRAFFIC RATE- G | PACKET DELAY- D | FRACTION EMPTY | AVERAGE BACKLOG |
|-------------|-----------------------|--------------------|--------------------|-------------------|--------------------|
| 1 - 200 | 0.285 | 0.395 | 19.877 | 0.665 | 2.1 |
| 201 - 400 | 0.320 | 0.390 | 16.328 | 0.650 | 1.2 |
| 401 - 600 | 0.255 | 0.425 | 22.824 | 0.660 | 2.8 |
| 601 - 800 | 0.290 | 0.475 | 26.172 | 0.630 | 4.0 |
| 801 - 1000 | 0.325 | 0.570 | 28.554 | 0.570 | 5.7 |
| 1001 - 1200 | 0.230 | 2.395 | 34.109 | 0.120 | 68.8 |
| 1201 - 1400 | 0.285 | 1.695 | 141.333 | 0.215 | 112.6 |
| 1401 - 1600 | 0.310 | 1.500 | 273.177 | 0.230 | 91.8 |
| 1601 - 1800 | 0.375 | 1.415 | 288.693 | 0.190 | 68.5 |
| 1801 - 2000 | 0.280 | 1.110 | 224.661 | 0.375 | 53.1 |
| 2001 - 2200 | 0.360 | 1.240 | 257.333 | 0.300 | 48.8 |
| 2201 - 2400 | 0.355 | 0.925 | 193.986 | 0.395 | 31.3 |
| 2401 - 2600 | 0.385 | 0.655 | 122.818 | 0.490 | 15.2 |
| 2601 - 2800 | 0.320 | 0.565 | 68.094 | 0.565 | 8.8 |
| 2801 - 3000 | 0.280 | 0.420 | 39.357 | 0.660 | 5.6 |
| 3001 - 3200 | 0.295 | 0.495 | 31.678 | 0.615 | 6.3 |
| 3201 - 3400 | 0.265 | 0.680 | 45.000 | 0.545 | 11.7 |
| 3401 - 3600 | 0.350 | 0.750 | 37.057 | 0.485 | 13.3 |
| 3601 - 3800 | 0.310 | 0.465 | 65.274 | 0.625 | 8.2 |
| 3801 - 4000 | 0.275 | 0.520 | 33.618 | 0.610 | 7.7 |
| 4001 - 4200 | 0.330 | 0.480 | 34.652 | 0.595 | 5.2 |
| 4201 - 4400 | 0.325 | 0.615 | 29.585 | 0.540 | 7.5 |
| 4401 - 4600 | 0.370 | 0.525 | 38.608 | 0.560 | 7.6 |
| 4601 - 4800 | 0.260 | 0.705 | 44.250 | 0.550 | 15.9 |
| 4801 - 5000 | 0.375 | 0.720 | 63.520 | 0.460 | 11.1 |
| 5001 - 5200 | 0.350 | 0.635 | 41.729 | 0.520 | 9.0 |
| 5201 - 5400 | 0.285 | 0.475 | 29.368 | 0.625 | 6.6 |
| 5401 - 5600 | 0.315 | 0.510 | 36.460 | 0.595 | 4.9 |
| 5601 - 5800 | 0.290 | 0.425 | 24.190 | 0.650 | 4.1 |
| 5801 - 6000 | 0.305 | 0.490 | 28.738 | 0.610 | 4.7 |

Figure 8—Simulation run for heuristic RCP subject to a channel input pulse

$M=400$. This explains the more significant degradation in channel performance given by the CONTEST algorithms shown in Table II for $M=400$ than in Table I for $M=200$.

In Figure 4, it was shown that in an uncontrolled slotted ALOHA channel, a channel input rate of 0.35 packet/slot was enough to cripple the channel indefinitely. In Figures 7 and 8, we show by simulation that under 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 $(n_0, S_0) = (4, 0.32)$ both before and after the pulse. During a period of 200 slots (namely, the time period 1000-1200 in the figures), the packet generation probability σ was increased such that $M\sigma=1.0$ 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.

Further discussions of results

In a real system, the channel input source will typically vary slowly with time; for example, the number of users fluctuates during the day. We must emphasize the fact that the control policies considered have been optimized to control statistical channel fluctuations under the assumption of a stationary channel input. Although we have shown 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 any long period of time (e.g., by limiting the maximum number of users who can "sign-on" and become active channel users).

The control action space of IRCP includes both control action spaces of ICP and RCP as subsets. Thus IRCP must give a channel performance at least as good as ICP and RCP. Next, comparing IRCP-CONTEST and Heuristic RCP, we see that the latter is easier to implement. However, under a

normal load (say $S_0 \leq 0.32$), 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.

CONCLUSIONS

Packet switched satellite and ground radio systems have been proposed as new alternatives for computer communications. A multi-access broadcast packet switching technique that has attracted considerable interest is the slotted ALOHA random access scheme. A slotted ALOHA channel multi-accessed by a large population of users has been shown to exhibit unstable behavior. Dynamic control schemes are necessary to prevent the occurrence of channel saturation in unstable channels. The dynamic channel control problem has been studied using a finite-state Markovian decision model in References 7 and 13 under the assumption of perfect channel state information.

In this paper we have studied dynamic channel control algorithms (CONTEST algorithms) which implement the theoretical control policies by using a heuristic scheme to estimate the instantaneous channel state. A heuristic retransmission control algorithm has also been studied which circumvents the state estimation problem. Simulation results 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, slotted ALOHA, reservation-ALOHA, carrier sense multi-access, 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 solve stability and dynamic control problems of other random access systems.

REFERENCES

1. Abramson, N., "THE ALOHA SYSTEM—Another Alternative for Computer Communications," Fall Joint Computer Conference, *AFIPS Conference Proceedings*, 1970, Vol. 37, pp. 281-285.
2. Crowther, W., R. Rettberg, D. Walden, S. Ornstein and F. Heart, "A System for Broadcast Communication: Reservation-ALOHA," *Proceedings of the Sixth Hawaii International Conference on System Sciences*, University of Hawaii, Honolulu, January, 1973.
3. Abramson, N., "Packet Switching with Satellites," National Computer Conference, New York, June 4-8, 1973, *AFIPS Conference Proceedings*, 1973, Vol. 42, pp. 695-702.
4. Kleinrock, L. and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel," National Computer Conference, New York, June 4-8, 1973, *AFIPS Conference Proceedings*, 1973, Vol. 42, pp. 703-710.
5. Roberts, L. G., "Dynamic Allocation of Satellite Capacity Through Packet Reservation," National Computer Conference, New York, June 4-8, 1973, *AFIPS Conference Proceedings*, 1973, Vol. 42, pp. 711-716.
6. Butterfield, S., R. Rettberg and D. Walden, "The Satellite IMP for the ARPA Network," Seventh Hawaii International Conference on System Sciences, University of Hawaii, Honolulu, January 8-10, 1974, *Proceedings of the Special Subconference on Computer Nets*, 1974.
7. Lam, S. S., *Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network*, Ph.D. Dissertation, Computer Science Department, University of California, Los Angeles, March 1974 (available as Technical Report UCLA-ENG-7429, April 1974).
8. Kleinrock, L. and F. A. Tobagi, "Carrier Sense Multiple Access for Packet Switched Radio Channels," *Proc. of the International Conference on Communications*, Minneapolis, Minn., June 1974.
9. Jackson, P. E. and C. D. Stubbs, "A Study of Multi-access Computer Communications," Spring Joint Computer Conference, *AFIPS Conference Proceedings*, 1969, Vol. 34, pp. 491-504.
10. Metcalfe, R. M., "Steady-State Analysis of a Slotted and Controlled ALOHA System with Blocking," *Proceedings of the Sixth Hawaii International Conference on System Sciences*, University of Hawaii, Honolulu, January 1973.
11. Kleinrock, L. and S. S. Lam, "On Stability of Packet Switching in a Random Multi-Access Broadcast Channel," Seventh Hawaii International Conference on System Sciences, University of Hawaii, Honolulu, January 8-10, 1974, *Proceedings of the Special Subconference on Computer Nets*, 1974.
12. Kleinrock, L. and S. S. Lam, "Packet Switching in a Multi-Access Broadcast Channel: Performance Evaluation," *IEEE Transactions on Communications*, Vol. COM-23, April 1975.
13. Lam, S. S. and L. Kleinrock, "Packet Switching in a Multi-Access Broadcast Channel: Dynamic Control Procedures," to appear in *IEEE Transactions on Communications*.
14. Metcalfe, R. M., *Packet Communications*, Ph.D. Dissertation, Harvard University, 1973 (available as MIT Project MAC Technical Report TR-114, December 1973).
15. Howard, R., *Dynamic Probabilistic Systems Vol. 1: Markov Models and Vol. 2: Semi-Markov and Decision Processes*, Wiley, New York, 1971.

