

An Exact Analysis and  
Performance Evaluation of  
Framed Aloha with Capture

by

Jeffrey E. Wieselthier  
Anthony Ephremides  
Larry A. Michaels

# AN EXACT ANALYSIS AND PERFORMANCE EVALUATION OF FRAMED ALOHA WITH CAPTURE

*Jeffrey E. Wieselthier*

Naval Research Laboratory  
Washington, D.C.

*Anthony Ephremides*

University of Maryland  
College Park, Maryland

*Larry A. Michaels*

Naval Research Laboratory  
Washington, D.C.

## ABSTRACT

A combinatorial technique is introduced that can be applied to the analysis of a variety of framed, contention-based protocols of multiple access, with or without dynamic frame-length control. Furthermore, we consider a general capture model that includes the case of perfect capture, as well as capture models that depend on the number of packets involved in a collision. We demonstrate the application of our technique to several of these cases.

Rewrite

## 1. INTRODUCTION

We consider the problem of channel random access in a system characterized by a slotted frame structure, as shown in Figure 1. As in typical slotted systems, the slot duration is equal to the length of a packet and all packet transmissions start at the beginning of a time slot. There is no coordination among users; thus, there is the possibility of collision, i.e., the transmission of two or more packets in a slot. "Framed ALOHA" and similar schemes have been studied by several investigators [1-3]. In this paper we present an exact analysis of the Framed ALOHA protocol for the case of a finite number of users. A new combinatorial technique is used for this analysis. Both uncontrolled and dynamically controlled versions of this protocol are considered. It is initially assumed that all collisions result in the destruction of all packets that are involved, but in Section 4 we incorporate into our analysis a general model for capture, in which one packet may be successfully received despite the transmission by other users in the same slot.

## 2. SYSTEM MODEL

The system consists of  $M$  unbuffered users, each of which can hold at most one packet at a time. Typically, the arrival process is bursty. All newly generated packets are transmitted in the frame in which they are generated, except for packets generated in the last slot of a frame, which are transmitted in the next frame. It is assumed that acknowledgment information is available before the end of the current frame, and without overhead penalty. A packet suffering a collision will be retransmitted in a subsequent frame; we initially assume that all such packets are retransmitted in the next frame with probability one, but later we address the case in which these packets are transmitted in the next frame with probability  $X$  that can be chosen to increase throughput. A user whose packet has suffered a collision is known as a *blocked* user. The precise retransmission strategy and the arrival process description are discussed later on. Now we define:

$T_i$  = number of packets that are transmitted in frame  $i$ ,

$U_i$  = number of users transmitting in frame  $i$ ,

$L_i$  = length of frame  $i$  (in slots),

$C_i$  = number of packets that suffer collisions in frame  $i$ ,

$S_i$  = number of successful transmissions in frame  $i$ ,

$A_i$  = number of new arrivals in frame  $i$ .\*

$\phi$  = probability of transmission by non-blocked user in frame.

Note that

$$T_i = C_{i-1} + A_i = S_i + C_i. \quad (1)$$

The colliding packet process  $C_i$ , also known as the *backlog process*, is of fundamental importance in the modeling of system dynamics, as we shall discuss in Section 5.

Capture in random access systems has been investigated to some extent [5-7]. For the Framed ALOHA protocol, the case of perfect capture i.e., the case in which one packet is successful in every slot chosen by one or more users) is straightforward to analyze, as will be demonstrated in Section 4. However, for non-perfect capture the analysis is complicated. The combinatorial technique that we propose is useful for analyzing more general capture models in which the probability of one user capturing a slot is dependent on the number of other users transmitting in that slot. Thus, although our mathematical formulation is perhaps more powerful than is actually necessary for the non-capture case, it permits the analysis of general models for capture, as well.

### 3. THE BASIC ANALYSIS TECHNIQUE—NO CAPTURE

We begin by considering a constant frame length  $L$ , although later we will consider schemes in which the frame length varies in response to channel traffic. We can for the moment neglect consideration of the arrival process and examine the distribution of the number of successful and colliding packets ( $S_i$  and  $C_i$  respectively), given the total number of packets,  $T_i$ , transmitted in the frame. We do note, however, that since the users are unbuffered the arrival process at a user

---

\*  $A_i$  is actually the number of new arrivals that are transmitted in frame  $i$ , and it includes packets that arrive during the last slot of frame  $(i-1)$ , but not those that arrive in the last slot of frame  $i$ ,

must be turned off whenever a new packet is generated. We assume it is turned back on in the frame following that in which the user's packet is successfully transmitted. Thus, each user transmits at most one packet in any frame. We assume that each of the  $T_i$  packets transmitted in frame  $i$  is equally likely to be transmitted in any of the  $L$  slots of the frame.

If each user were allowed to transmit more than a single packet in one frame then the number of packets transmitted would be greater than the number of users transmitting in that frame. Clearly, the transmissions of any given user would not be independent, because he would not transmit two or more packets in the same slot. Our analysis here, based on the simpler model of each user transmitting at most a single packet in each frame, provides a lower bound for the throughput of the other case just discussed if  $T_i$  is the same in both cases. In this sense our analysis here, although exact for the case of at most a single packet transmitted per frame per user, is applicable, albeit pessimistic, for the other case.

To simplify notation, for now we suppress the subscripts that represent frame number. These subscripts will be reintroduced when we discuss the dynamic behavior of the system, i.e., the evolution of system behavior from one frame to the next.

We define the *state* of the system in a given frame in which  $T$  users transmit as

$$\mathbf{n}_T = (n_T(1), n_T(2), \dots, n_T(T)), \quad (2)$$

where  $n_T(j) =$  number of slots in each of which exactly  $j$  users transmit.

For example, consider the case of  $T = 5$ . The state  $\mathbf{n}_T = (3,1,0,0,0)$  corresponds to a realization in which one slot is occupied by two users and three slots are singly occupied.

In general, the states  $\mathbf{n}_T$  that are realizable must satisfy two physical constraints. First, since there are a total of  $T$  packets in the frame we must have, for any state  $\mathbf{n}_T$ ,

$$\sum_{j=1}^T j n_T(j) = T \quad (3)$$

Also, the total number of occupied slots, which we denote by  $m$ , can not be greater than either the frame size ( $L$ ) or the number of users transmitting in the frame ( $T$ ), i.e.,

$$m = \sum_{j=1}^T n_T(j) \leq \min(L, T). \quad (4)$$

We want to determine  $P(\mathbf{n}_T)$ , which is the conditional probability distribution for the states, for a given  $T$ . This distribution is necessary for the study of the system dynamics, because the throughput is directly related to some of the components of the state  $\mathbf{n}_T$ . For example, in the present case of no capture the number of packets transmitted successfully in a frame is simply given by,

$$S = n_T(1) \quad (5)$$

and the number of colliding packets by

$$C = T - n_T(1). \quad (6)$$

In addition to the evaluation of  $P(\mathbf{n}_T)$  for any given state, we must be able to enumerate the states  $\mathbf{n}_T$  that can occur for any particular values of  $T$  and  $L$ , i.e., all states that satisfy eqs. (3) and (4), in order to implement the calculations that will be presented later on. The detailed state description and enumeration are not strictly necessary when dealing with either noncapture or perfect capture. They are needed, however, to analyze general capture models. We first present a procedure for the enumeration of states, and we then present the derivation for the probability distribution for the states.

### The Enumeration of States

The enumeration of the states  $\mathbf{n}_T$  that can occur for a fixed value of  $L$  (frame length) and for any particular value of  $T$ , i.e., all states for which eqs. (3) and (4) are satisfied, is equivalent to the construction of Young's lattice [8]. One approach is to proceed iteratively as follows:

Assume we know all states for a given value of  $T$ . (For example, we may start with the trivial case of  $T = 1$ , for which the only possible state is  $\mathbf{n}_T = (1)$ ). For each state  $\mathbf{n}_T$  that is consistent with the presence of  $T$  users transmitting in the frame, we determine the states  $\mathbf{n}_T$  that can be generated as one additional user is added to the system. To do so we first consider the case in which the new  $(T+1)^{\text{st}}$  user has chosen a slot not previously chosen by any of the first  $T$

users. The number of singly occupied slots thus increases by 1 while the number of slots containing  $i$  users ( $i = 2, 3, \dots, T$ ) remains unchanged. The new states are thus generated by the following procedure:

for each state  $\mathbf{n}_T$ , set

$$\begin{aligned} n_{T+1}(1) &= n_T(1) + 1, \\ n_{T+1}(i) &= n_T(i), \quad i = 2, 3, \dots, T \end{aligned} \quad (7)$$

We now consider the case in which the new  $(T+1)^{\text{st}}$  user has chosen one of the slots already chosen by one of the first  $T$  users. If that slot already contained  $i$  users, then it would now contain  $i+1$  users, thus decrementing the number containing  $i$  users by 1 while incrementing the number containing  $i+1$  by 1. The new states are thus generated by the following procedure:

for each state  $\mathbf{n}_T$  and every  $n_T(i) \neq 0$ , set

$$\begin{aligned} n_{T+1}(i) &= n_T(i) - 1 \\ n_{T+1}(i+1) &= n_T(i+1) + 1. \end{aligned} \quad (8)$$

Note that duplicate identical states are generated by this process, because two different  $\mathbf{n}_T$  states can evolve to the same  $\mathbf{n}_{T+1}$  successor state. Such duplicate states are easily recognized and thinned out in this iterative procedure. Also with this procedure, states for which the number of occupied slots ( $m$ ) is greater than  $L$  may be generated. Such states are discarded.

As an example we show in Table 1 the resulting states for  $T < 5$  (and  $L > T$ ).

Table 1 — Enumeration of states for  $T < 5$  and  $L > T$

$$T = 1: (1)$$

$$T = 2: (2,0), (0,1)$$

$$T = 3: (3,0,0), (1,1,0), (0,0,1)$$

$$T = 4: (4,0,0,0), (2,1,0,0), (0,2,0,0), (1,0,1,0), (0,0,0,1)$$

$$T = 5: (5,0,0,0,0), (3,1,0,0,0), (1,2,0,0,0), (2,0,1,0,0), (0,1,1,0,0), (1,0,0,1,0), (0,0,0,0,1)$$

The number of states increases dramatically with increasing  $T$  as shown in the table below.

T	L	number of states
10	10	42
20	10	530
25	10	1455
30	10	3590
32	10	5013
20	20	627
25	25	1958
30	30	5604
32	32	8349

*add some larger values of T, at least for L=10.*

39 10 14663  
 40 10 16928  
~~45~~ 10 33401

**The State Probability Distribution:  $P(\mathbf{n}_T)$**

We now determine the probability distribution  $P(\mathbf{n}_T)$ . We first observe that we are not interested in *which* of the  $L$  slots in a frame are occupied by a specific number of packets, but only in the *number* of such slots. Thus, each state  $\mathbf{n}_T$  corresponds to several different specific slot occupancy realizations, all of which are equally likely.

Our approach is to consider  $T$  users, each of which places a packet into one of the  $L$  slots of the frame; each slot is chosen by each user in equally likely fashion with probability  $1/L$ . We now consider the sequence in which some subset of the  $L$  slots is filled as we examine the  $T$  users, which are numbered from 1 to  $T$ . We want to realize the state  $\mathbf{n}_T = (n_T(1), n_T(2), \dots, n_T(T))$ .

There are numerous ways to do so. Let us consider a realization in which the first  $n_T(1)$  users all choose different slots; these are followed by  $n_T(2)$  pairs of users, such that the two members of each pair choose the same slot as each other, but different from all those previously chosen; these are followed by  $n_T(3)$  groups of three users, such that the three members of each triplet choose the same slot as each other, but different from all those previously chosen, etc. We



note that  $n_T(T)$ , and in fact  $n_T(j)$  for  $j > T/2$ , cannot be greater than 1.

As we consider the users numbered from 1 to  $T$ , we evaluate the probability that the conditions corresponding to the specific realization of  $\mathbf{n}_T$ , which we denote  $\mathbf{I}_T$ , are satisfied. We now demonstrate that:

**Proposition**

$$Pr(\mathbf{I}_T) = \frac{L!}{L^T (L-m)!} \quad (9)$$

where  $m$  is the total number of occupied slots, as defined in eq. (4). We note that the probability of any other specific realization corresponding to the same state  $\mathbf{n}_T$  is identical to that which we have derived here.

**Proof**

We first consider the  $n_T(1)$  singly occupied slots. The first user chooses a slot at random; thus, the probability that he picks a previously unchosen slot is simply  $L/L = 1$ . The second user also chooses a slot at random; the probability that he picks a previously unchosen slot is  $(L-1)/L$ . Continuing in the same manner, user  $n_T(1)$  chooses a slot that was not previously chosen with probability  $[L-(n_T(1)-1)]/L$ . Thus,

$$\begin{aligned} &Pr(\text{first } n_T(1) \text{ users choose different slots}) \\ &= \frac{L}{L} \frac{(L-1)}{L} \frac{(L-2)}{L} \dots \frac{[L-(n_T(1)-1)]}{L}. \end{aligned} \quad (10)$$

Given that the first  $n_T(1)$  users have all chosen different slots, we now evaluate the probability that we then have  $n_T(2)$  pairs of users that choose the same slot as each other, but different from previously chosen slots. Thus, user  $n_T(1)+1$  must choose a slot that is different from that of the first  $n_T(1)$  packets, an event that occurs with probability  $[L-n_T(1)]/L$ . User  $n_T(1)+2$  must choose the *same* slot as user  $n_T(1)+1$ , an event that occurs with probability  $1/L$ . Similarly, user  $n_T(1)+3$  must choose a slot not previously chosen (resulting in a probability of  $[L-n_T(1)-1]/L$ ), and user  $n_T(1)+4$  must choose the same slot as user  $n_T(1)+3$  (resulting in

1/L). Continuing in the same manner, we finally obtain

$$\begin{aligned}
 Pr(\mathbf{I}_T) &= \left( \frac{L}{L} \right) \left( \frac{L-1}{L} \right) \left( \frac{L-2}{L} \right) \cdots \left( \frac{L-[n_T(1)-1]}{L} \right) \\
 &\quad \langle \text{-----} n_T(1) \text{ factors} \text{-----} \rangle \\
 &\times \left( \frac{L-n_T(1)}{L^2} \right) \left( \frac{L-n_T(1)-1}{L^2} \right) \cdots \left( \frac{L-n_T(1)-[n_T(2)-1]}{L^2} \right) \\
 &\quad \langle \text{-----} n_T(2) \text{ factors} \text{-----} \rangle \\
 &\times \left( \frac{L-n_T(1)-n_T(2)}{L^3} \right) \cdots \left( \frac{L-n_T(1)-n_T(2)-[n_T(3)-1]}{L^3} \right) \\
 &\quad \langle \text{-----} n_T(3) \text{ factors} \text{-----} \rangle \\
 &\quad \vdots \\
 &\quad \vdots \\
 &\times \frac{(L-n_T(1)-n_T(2)-\cdots-[n_T(T)-1])}{L^T} \\
 &\quad \langle \text{-----} n_T(T) \text{ factors}(\leq 1) \text{-----} \rangle \\
 &= \frac{L(L-1)(L-2) \cdots [L-(n_T(1)+n_T(2)+n_T(T)-1)]}{L^{n_T(1)}(L^2)^{n_T(2)} \cdots (L^T)^{n_T(T)}} \\
 &= \frac{L!}{L^T(L-m)!} \tag{11}
 \end{aligned}$$

---

Thus, we can write,

$$P(\mathbf{n}_T) = N(\mathbf{n}_T)Pr(\mathbf{I}_T) \tag{12}$$

where,

a)  $Pr(\mathbf{I}_T)$  = the conditional probability of one particular realization of  $\mathbf{n}_T$ , like the one described above

and,

b)  $N(\mathbf{n}_T) =$  the number of the equally likely realizations of  $\mathbf{n}_T$ .

We now determine  $N(\mathbf{n}_T)$ . This determination coincides with finding the number of different partitions of a set of  $T$  objects into classes of  $n_T(j)$  groups, each group having  $j$  objects, for  $j = 1, 2, \dots, T$ . This number is given in [8] by,

$$N(\mathbf{n}_T) = \frac{T!}{\prod_{j=1}^T (j!)^{n_T(j)} n_T(j)!} \quad (13)$$

Combining this result with the probability of a specific realization results in

$$P(\mathbf{n}_T) = \frac{L! T!}{L^T (L-m)! \prod_{j=1}^T (j!)^{n_T(j)} n_T(j)!} \quad (14)$$

We have demonstrated a procedure to enumerate all possible states  $\mathbf{n}_T$  corresponding to a given number of time slots ( $L$ ) and packets transmitted ( $T$ ) in the frame. We have also determined the probability of any arbitrary state  $\mathbf{n}_T$ , given  $L$  and  $T$ . For the case of no capture that we are now considering, the conditional distributions (given  $L$  and  $T$ ) of the number of successful packets and colliding packets can be evaluated numerically from eq. (14) for all states  $\mathbf{n}_T$  as follows:

$$\begin{aligned} Pr(S = s | T) &= \sum_{\mathbf{n}_T} Pr(\overset{\text{not bold}}{\mathbf{n}_T}(1) = s | \mathbf{n}_T) P(\mathbf{n}_T) \\ &= \sum_{\mathbf{n}_T \in Z_s} P(\mathbf{n}_T). \end{aligned}$$

where  $Z_s$  is the set of states for which  $n_T(1) = s$ . Similarly,

$$Pr(C = T-s | T) = Pr(S = s | T).$$

As noted in the introduction, our mathematical formulation is not strictly necessary for the case of systems without capture. In fact, Szpankowski's analysis of S/V-ALOHA, or "Slotted ALOHA with  $V$  Subchannels," [4], can be applied to Framed ALOHA for noncapture channels. In S/V-ALOHA the channel is divided into  $V$  parallel sub-channels. The time axis is divided into multi-slots, each containing  $V$  slots, i.e., one in each sub-channel. The number of successful

packets in a multislot is thus equal to the number of slots chosen by exactly one user. The choice of a slot within a multislot is therefore analogous to the choice of a slot within a frame in a Framed ALOHA model. The mathematical model of [4] can thus be applied to the case of Framed ALOHA in the case where there is no capture. Here we have introduced a technique that, in addition to being applicable to ordinary Framed ALOHA, is capable of analyzing performance when capture is allowed, in which case the other methods of analysis are not applicable.

#### 4. FRAMED ALOHA WITH CAPTURE

We consider now the case in which one of the users captures the channel, despite the presence of other users' packets in the same time slot. Depending upon the implementation, the capture property may be based on either relative signal magnitudes (i.e., the stronger signal captures the channel) or on time of arrival. In frequency hopped systems, one signal can capture the transmitter provided that later signals are delayed by more than one dwell time. This may occur in practical slotted systems, either as a result of different propagation delays by users at different communication ranges, or by an intentional randomization of transmission times (within a guard time established for this purpose at the beginning of each slot). It is possible that two geographically separated receivers may capture the packets of different users. In our analysis, we assume that the same packet (if any) is captured by all users, or equivalently, in the mathematical sense, that there is a single destination.

##### Perfect Capture

The simplest, and most optimistic, model for capture is that of perfect capture, i.e., in every slot in which two or more users transmit one user captures the channel, and is therefore successful; the other users are unsuccessful and must be retransmitted in the next frame. In this case the number of successful packets in the frame,  $S$ , is simply equal to the number of occupied slots. From eq. (4) we thus obtain,

$$S = m = \sum_{j=1}^T n_T(j). \quad (15)$$

The probability distribution function for  $\mathbf{n}_T$  is evaluated as presented earlier in this paper. We must in this case, however, reinterpret  $C$  as the **number of unsuccessful packets in the frame**, rather than the number transmitted in slots with one or more others. For the present case of perfect capture, we thus have,

$$C = T - m. \quad (16)$$

The distribution for the number of unsuccessful packets is therefore obtained as,

$$\begin{aligned} P(C=T-m \mid T,L) &= Pr(\text{exactly } m \text{ slots are occupied} \mid T,L), \\ &= \sum_{\mathbf{n}_T \in Z} P(\mathbf{n}_T), \end{aligned} \quad (17)$$

where  $Z$  is the set of states such that  $m$  slots are occupied.

The case of perfect capture can actually be analyzed without making use of the powerful combinatorial technique of this paper, since it only requires the statistics of the number of slots chosen by one or more users. This is equivalent to the classical occupancy problem in which  $T$  balls (packets) are distributed among  $L$  cells (slots). The probability that  $m$  slots are occupied is simply the probability that  $(L-m)$  slots are empty, which is given in [9] by,

$$Pr(m \text{ slots are occupied} \mid T,L) = \binom{L}{L-m} \sum_{\nu=0}^m (-1)^\nu \binom{m}{\nu} \frac{(m-\nu)^T}{L}. \quad (18)$$

### More General Capture Models

The full force of the combinatorial procedure presented in this paper is needed to accommodate other models for capture, such as those considered in [5-7]. For any such model the conditional probability distribution of the number of successful packets, given  $\mathbf{n}_T$  must be determined.

For example, consider a deterministic model in which one packet captures the receiver whenever the number of users in the slot is less than or equal to some threshold  $\tau$ . We then obtain,

$$S = \sum_{j=1}^{\tau} n_T(j), \quad (19)$$

which is simply a truncated version of eq. (15). We thus have,

$$\begin{aligned} P(C = T-S | T, L) &= Pr(S \text{ successes} | T, L). \\ &= \sum_{\mathbf{n}_T \in \hat{Z}} P(\mathbf{n}_T) \end{aligned} \quad (20)$$

where  $\hat{Z}$  is the set of states such that eq. (19) is satisfied.

We may also consider the more general and more realistic case in which, with probability  $Q(j)$ , the receiver will be captured by one of the users whenever a total of  $j$  users transmit in the slot.\* The expected value of the number of successes in the frame is then

$$E(S | \mathbf{n}_T) = \sum_{j=1}^T Q(j) n_T(j). \quad (21)$$

To determine equilibrium performance, we need the conditional probability distribution of  $S$ , for a given state  $\mathbf{n}_T$ . To evaluate this distribution we note that if we are given the number of users transmitting in any arbitrary slot, then the success or failure of a packet to capture the receiver in that slot is independent of that in any other slot. Since there are  $m$  occupied slots, we have  $m$  independent Bernoulli trials,  $n_T(j)$  of which have success probability  $Q(j)$ , for  $j = 1, 2, \dots, T$ . In each slot of the  $n_T(j)$  slots (in each of which exactly  $j$  users are transmitting) there can be at most one successful transmission, an event that occurs with probability  $Q(j)$ . Therefore the total number of successful transmissions,  $S(j)$ , in that entire class of slots is binomially distributed with parameters  $n_T(j)$  and  $Q(j)$ . Its generating function is thus given by

$$G_j(z) = [1 - Q(j) + zQ(j)]^{n_T(j)}. \quad (22)$$

The total successful packet process is the superposition of the successful transmissions  $S(j)$  over all values of  $j = 1, \dots, T$ . Thus the total number of successes is given by

$$S = \sum_{j=1}^T S(j) \quad (23)$$

and, owing to the independence amongst the  $S(j)$ 's, the generating function of  $S$  is given by

\* The quantity  $Q(j)$  can also incorporate the effects of channel noise on the probability of correct reception.   
 can also be incorporated into the quantity  $Q(j)$ .

$$G(z) = \prod_{j=1}^T [1 - Q(j) + zQ(j)]^{n_T(j)}. \quad (24)$$

Expanding this expression yields a polynomial form for the generating function:

$$G(z) = \sum_{s=1}^m B_s z^s \quad (25)$$

in which

$$B_s = Pr(s \text{ successes in frame} | \mathbf{n}_T). \quad (26)$$

The probability distribution of unsuccessful transmissions given  $\mathbf{n}_T$  is easily found from that of successful transmissions since, given a particular state, the probability of  $k$  failures is simply equal to the probability of  $T-k$  successes. The distributions for the number of successes and failures, given  $T$  and  $L$ , are found by summing the distributions for each state while weighting them by the probability of that state. They are given by  $P(S|T)$  and  $P(C|T)$ :

$$P(S = s | T) = \sum_{\mathbf{n}_T} B_s P(\mathbf{n}_T) \quad (27)$$

$$P(C = c | T) = \sum_{\mathbf{n}_T} B_{T-c} P(\mathbf{n}_T) \quad (28)$$

where the summations are performed over all states consistent with the values of  $L$  and  $T$ , as discussed in Section 3, using the expression for  $P(\mathbf{n}_T)$  presented in eq. (13), and the expressions for  $B_s$  that follow from the expansion in eqs. (23) and (24).

## 5. DYNAMIC SYSTEM MODEL: UNCONTROLLED FRAMED ALOHA

We now consider the dynamics as the system evolves from frame to frame. The subscripts representing frame number must therefore be reinserted into the notation. The frame length is kept constant at  $L$  slots in the present discussion. We first assume that all unsuccessful packets are retransmitted in the next frame. The number of packets transmitted in frame  $i$  is therefore

$$T_i = C_{i-1} + A_i. \quad (26)$$

The next step in the analysis is to find the distribution of  $T_i$  given  $C_{i-1}$ . This is denoted as

$P(T_i | C_{i-1})$  and depends on the new arrival process  $A_i$ , where  $A_i$  is the number of new packets transmitted in frame  $i$ . Since it is assumed that blocked users do not produce new packets, and the number of users is finite,  $A_i$  depends on  $C_{i-1}$ . Since each non-blocked user transmits with probability  $\phi$  in the current frame,  $A_i$  is binomially distributed with parameters  $M - C_{i-1}$  and  $\phi$ . Since  $T_i = A_i + C_{i-1}$ , it is also binomially distributed (given  $C_i$ ). Therefore, we have the following:

$$P(A_i | C_{i-1}) = \binom{M - C_{i-1}}{A_i} \phi^{A_i} (1 - \phi)^{M - C_{i-1} - A_i}, \quad 0 \leq A_i \leq M - C_{i-1}. \quad (30)$$

$$P(T_i | C_{i-1}) = \binom{M - C_{i-1}}{T_i - C_{i-1}} \phi^{T_i - C_{i-1}} (1 - \phi)^{M - T_i}, \quad C_{i-1} \leq T_i \leq M. \quad (31)$$

$$P(C_i | C_{i-1}) = \sum_{T_i=1}^M P(C_i | T_i) P(T_i | C_{i-1}). \quad (32)$$

where  $P(C_i | T_i)$  is obtained from eq. (-).

The quantities  $P(C_i | C_{i-1})$ , for  $0 \leq C_i, C_{i-1} \leq M$ , are the elements of the transition probability matrix that represents the system dynamics in terms of the backlog of packets measured at the beginning (or equivalently at the end) of each frame. The system we have represented thus far is uncontrolled in the following senses: (a) the frame length is kept fixed at  $L$  slots; (b) the arrival process at each non-blocked user is independent of system backlog or channel traffic; (c) all blocked users transmit their packets in some slot in the frame with probability 1. The steady state distribution of backlog is evaluated by starting with an arbitrary probability vector for  $P(C_{i-1})$ , and iterating until convergence is achieved, or by solving the Markov transition equations for a given initial condition. In all of our examples, the iteration has converged, regardless of whether the initial state is the empty state or if all  $M$  users are blocked.

Once the equilibrium distribution for the channel backlog has been determined, it is possible to determine the expected values of traffic, throughput, and backlog as follows:

$$E(S_i) = \sum_{C_{i-1}=0}^M E(S_i | C_{i-1}) P(C_{i-1})$$



$$= \sum_{C_{i-1}=0}^M \left( \sum_{S_i=0}^M S_i P(S_i | C_{i-1}) \right) P(C_{i-1}) \quad (33)$$

$$\begin{aligned} E(T_i) &= \sum_{C_{i-1}=0}^M E(T_i | C_{i-1}) P(C_{i-1}) \\ &= \sum_{C_{i-1}=0}^M \left( \sum_{T_i=0}^M T_i P(T_i | C_{i-1}) \right) P(C_{i-1}) \end{aligned} \quad (34)$$

$$E(C_i) = \sum_{C_{i-1}=0}^M C_{i-1} P(C_{i-1}) \quad (35)$$

where  $P(C_{i-1})$  is the steady state distribution of  $C_i$ .

For the case of no capture ( $Q(j) = 0$  for  $j \neq 1$ ),  $X = 1$ , and  $L_i$  held fixed at  $L$ , the results show that the maximum obtainable throughput is  $\approx 1/e$  for values of  $M > L$ . The highest throughput, which is obtained when  $M = L$ , is slightly greater than  $1/e$ . Figures 1 and 3 show throughput as a function of  $\phi$  as  $M$  is varied and  $L = 8$  and 10. It can be seen that as  $M/L$  increases, the throughput versus  $\phi$  curve becomes more sharply peaked and the optimal value of  $\phi$  decreases. Figures 2 and 4 are plots of traffic versus throughput for the same cases. They show that for large values of  $M/L$ , the throughput peaks when  $E(T_i)/L = 1$  and decreases as channel traffic increases beyond this point. Figure 5 shows the steady state backlog as a function of  $\phi$  for  $L = 10$  and various values of  $M$ . For large  $M/L$  the backlog increases rapidly for  $\phi$  greater than  $\approx 0.1$ .

We have considered the following capture model which is based on [—]

$$\begin{aligned} Q(1) &= 1 \\ Q(j) &= \alpha^j \quad j \geq 2 \end{aligned} \quad (36)$$

Figure -- shows traffic versus throughput for several values of  $\alpha$ . Note that maximum throughput increases as  $\alpha$  increases, and that the effects of overload are decreased, as evidenced by the fact that the curves do not bend back as sharply. Figure -- shows throughput performance as a function of  $\phi$ .

## 6. A SYSTEM IN WHICH NOT ALL BLOCKED PACKETS ARE TRANSMITTED

Thus far, we have assumed that all unsuccessful packets are retransmitted in the following frame. Such systems, when heavily loaded, can suffer from unacceptably low values of throughput, as we have just shown. To alleviate this problem, we consider the case in which each blocked user retransmits its packet sometime in the next frame with some probability  $X_i$ . Such a scheme helps to increase throughput at large values of  $\phi$ .

The following quantities describe the retransmission process:

$X_i$  = probability of retransmission of blocked packet in frame  $i$

$U_i$  = number of blocked packets not retransmitted in frame  $i$ .

In the present discussion,  $X_i$  is held constant at  $X$  over all frames. In Section 7, the case of a dynamically varying  $X_i$  is discussed. The number of packets transmitted in frame  $i$  is now (see eq. (1))

$$T_i = C_{i-1} + A_i - U_i. \quad (37)$$

$C_i$  now includes both unsuccessfully transmitted packets from frame  $i$ , as well as previously blocked packets which are not retransmitted in frame  $i$ . Also, we have

$$C_i = T_i - S_i + U_i. \quad (38)$$

To find the distributions for  $P(C_i | T_i, U_i)$  and  $P(T_i | C_{i-1}, U_i)$  the previously found distributions for the case of  $U_i = 0$  ( $X=1$ ) must be shifted by the amount  $U_i$ . Therefore:

$$Pr(C_i = c - U_i | T_i, U_i) = Pr(C_i = c | T_i, U_i = 0) \quad (39)$$

$$Pr(T_i = t + U_i | C_{i-1}, U_i) = Pr(T_i = t | C_{i-1}, U_i = 0). \quad (40)$$

The new distributions are:

$$P(C_i | T_i, U_i) = \sum_{\mathbf{n}_T} B_{T_i + U_i - C_i} P(\mathbf{n}_T) \quad (41)$$

$$P(T_i | C_{i-1}, U_i) = \left\{ \begin{matrix} M - C_{i-1} \\ T_i + U_i - C_{i-1} \end{matrix} \right\} \phi^{T_i + U_i - C_{i-1}} (1 - \phi)^{M - T_i - U_i} \quad (42)$$

$P(S_i | T_i, U_i)$  is unchanged from the case of  $U_i = 0$ . The distribution of  $U_i$  depends on  $C_{i-1}$ , and is binomial with parameters  $C_{i-1}$  and  $X$ . Its distribution is given by:

$$P(U_i | C_{i-1}) = \binom{C_{i-1}}{U_i} (1-X)^U X^{C_{i-1}-U}, \quad (43)$$

The probability distribution of  $T_i$  given  $C_{i-1}$  is therefore:

$$P(T_i | C_{i-1}) = \sum_{U_i=0}^{C_{i-1}} P(T_i | C_{i-1}, U_i) P(U_i | C_{i-1}). \quad (44)$$

We now have everything that is needed to evaluate the transition probability matrix for the backlog process. Thus, we have

$$\begin{aligned} P(C_i | C_{i-1}) &= \sum_{U_i=0}^{C_{i-1}} \sum_{T_i=0}^{M-U_i} P(C_i | T_i, U_i, C_{i-1}) P(T_i, U_i | C_{i-1}) \\ &= \sum_{U_i=0}^{C_{i-1}} \left[ \sum_{T_i=0}^{M-U_i} P(C_i | T_i, U_i, C_{i-1}) P(T_i | U_i, C_{i-1}) \right] P(U_i | C_{i-1}). \end{aligned} \quad (45)$$

Note that  $P(C_i | T_i, U_i, C_{i-1})$  is actually independent of  $C_{i-1}$ , given that  $T_i$  and  $U_i$  are known.  $P(S_i | C_{i-1})$  is found by replacing  $C_i$  by  $S_i$  in eq. (-).

Figures 6 through 13 show throughput versus  $\phi$  and traffic versus throughput as  $X$  is held constant at a value which is varied from 1.0 to 0.1. When  $X$  is decreased from 1, the traffic versus throughput curve begins to flatten out until  $X \approx 0.3$  for the  $M = 30$  case (figure 6;  $L=10, M=30$ ), and  $X \approx 0.4$  for the  $M = 20$  case (figure 10;  $L=10, M=20$ ). At these points the throughput experiences little decrease as traffic is increased. Below  $X \approx 0.3$  (figure 6), or  $X \approx 0.4$  (figure 10) the throughput becomes a gradually increasing function of traffic.

## 7. DYNAMICALLY CONTROLLED SYSTEMS

In the previous section, we considered a system in which  $X_i$ , the probability that a blocked user retransmits its packet in frame  $i$ , is held constant at a value of  $X$ . We now consider a system in which  $X_i$  can be chosen as a function of channel backlog, and the frame length is again fixed at  $L$  slots. Our goal is to choose  $X_i$  such that  $S_i$  is maximized. We initially assume that in

frame  $i$ , all users know  $C_{i-1}$ , the number of blocked users from the previous frame, as well as  $\phi$ , the probability that a non-blocked user transmits in the frame. Although channel backlog is not a directly observable quantity, it can be estimated, based on channel feedback information. Some techniques for traffic estimation are discussed in [2,3,12].

Although the events in the  $L$  slots of a frame are not independent (since a user can transmit at most one packet in a frame), the statistics of the events in each of the slots are identical. Thus, to maximize  $S_i$  it is sufficient to maximize the probability that there is a successful transmission in any particular slot. We first consider the case of no capture. We have,

$$\begin{aligned} Pr(\text{success in any particular slot}) &= Pr(\text{exactly 1 user in any particular slot}) \\ &= Pr(1 \text{ new user, no blocked users}) + Pr(\text{no new user, 1 blocked user}) \end{aligned} \quad (46)$$

To simplify the notation, we let

$$\begin{aligned} p &= \frac{\phi}{L} = \text{probability a particular non-blocked user transmits in an arbitrary given slot} \\ q &= \frac{X}{L} = \text{probability a particular blocked user transmits in an arbitrary given slot} \end{aligned}$$

Thus, in an arbitrary slot of frame  $i$ ,  $(M-C_{i-1})$  users each transmit with probability  $p$ , and  $C_{i-1}$  users each transmit with probability  $q$ . Therefore, we have,

$$\begin{aligned} Pr(\text{exactly 1 user in any particular slot}) &= (M-C_{i-1})p(1-p)^{(M-C_{i-1}-1)}(1-q)^{C_{i-1}} + (1-p)^{M-C_{i-1}}C_{i-1}q(1-q)^{C_{i-1}-1} \end{aligned} \quad (47)$$

To maximize with respect to  $q$ , we simply differentiate with respect to  $q$ , and set the result equal to 0, which yields,

$$X_i^* = L \left[ \frac{\phi(M-C_{i-1}+1) - L}{M\phi - CL} \right]. \quad (48)$$

Note that when  $X_i = X_i^*$ , the point at which throughput is maximized, the expected number of packets transmitted per slot is not equal to 1. This is a consequence of the fact that the superposition of two Bernoulli sequences with different parameters does not result in a binomial distribution.

Equation (--) will yield values for  $X_i^*$  which are greater than one as well as values which are negative. However, since  $X_i$  is a probability, it must be between 0 and 1. The upper and lower bounds on  $X_i$  are therefore set at 1 and 0 respectively. For each value of  $C_{i-1}$  between 0 and  $M$ , the optimum value of  $X_i^*$  is found, and the new distribution for  $U_i$ , given  $C_{i-1}$  is calculated. The calculation is as in Section --, except that now  $X_i$  is a function of  $C_{i-1}$ .

Performance results for  $X = X_i^*$  were included in the plots of Figures -- - -. We see that the performance of a system that uses a *good* fixed value of  $X$  is almost as good as that of a system that uses  $X_i^*$ . This result is consistent with those of the early slotted ALOHA studies, in which it was shown that performance is relatively insensitive to the packet retransmission probability [-]. This result also indicates that inconsistencies in channel backlog estimates are not expected to affect performance adversely. We also note that in [11] the retransmission probability was chosen as a threshold-based function of the channel backlog. Although not necessarily optimal, such threshold protocols stabilize slotted ALOHA.

For a system with capture, the probability of successful packet transmission in an arbitrarily chosen slot is

$$Pr(\text{success} \mid C_{i-1}) = \sum_{j=1}^M Pr(j \text{ users transmit in slot} \mid C_{i-1}) Q(j) \quad (49)$$

where

$$\begin{aligned} & Pr(j \text{ transmit} \mid C_{i-1}) \\ &= \sum_{b=0}^j Pr(b \text{ blocked users transmit, } j-b \text{ non-blocked users transmit} \mid C_{i-1}) \\ &= \sum_{b=0}^j \left[ \binom{C_{i-1}}{b} q^b (1-q)^{C_{i-1}-b} \right] \left[ \binom{M-C_{i-1}}{j-b} p^{j-b} (1-p)^{M-C_{i-1}-j+b} \right]. \end{aligned} \quad (50)$$

$X_i^*$ , the value of  $X = qL$  that maximizes the probability of successful packet transmission (eq. (49)) for each value of  $C_{i-1}$  and  $\phi$ , can be determined computationally using search techniques.

Although this would be a tedious calculation, it could be performed once, offline, and the results stored in a lookup table.

We can also consider a system in which the frame length,  $L_i$ , is varied as a function of

channel backlog, while  $X_i$  is held fixed at some value  $X$ . Since  $L_i$  can only assume integer values, the best way to determine  $L_i$  may be simply to search for the value of  $L$  that maximizes throughput per time slot (eq. (–) for the no-capture case and eq.(–) for a system with capture). For the no-capture case, a good value to start the search is that for which the expected channel traffic per slot is approximately equal to 1 packet. Choosing  $L_i$  such that  $E(T_i | C_{i-1}) = L_i$  yields:

$$L_i^* = (M - C_{i-1})\phi + C_{i-1}X_i. \quad (51)$$

Since  $L_i$  must be an integer, the result from the above equation is rounded to the nearest integer value.  $L_i = 0$  is not allowed since it would mean skipping frame  $i$ . The probability distribution for  $C_i$  given  $C_{i-1}$  is now calculated using the distributions of  $P(S_i | T_i, L_i)$  and  $P(C_i | T_i, L_i)$ , where each value of  $L_i^*$  corresponds to a value of  $C_{i-1}$ .  $E(S_i | C_{i-1})$  and  $E(T_i | C_{i-1})$  are each divided by the value of  $L_i^*$  corresponding to that value of  $C_{i-1}$  to give throughput and traffic per slot.

It would be more difficult to implement a system with variable frame length than one with variable retransmission probability. We have noted that an accurate estimate of channel backlog may not be available. Moreover, since this is a distributed system, not all users would have the same estimate of backlog, and so they may be operating with different frame lengths. Our mathematical model for Framed ALOHA does not accommodate such asynchronous operation, and it is not clear at this point how it would affect performance. However, as noted earlier, inconsistencies in channel backlog estimates are not expected to affect performance adversely in a system with fixed frame length and variable retransmission probability.

## 8. THE MODELING OF LARGER SYSTEMS

In this paper we have presented an exact analysis of Framed ALOHA that can accommodate a general model for capture. Of course, as with many similar analytical methods, there are some serious computational limitations that must be addressed. We have noted in Section 3 that the number of states increases dramatically as  $T$  increases. For example, for  $T_i = 32$  (the largest

value we have used in our calculations) and  $L = 10$ , there are 5013 states whose probability must be evaluated. When the number of states is too great, computational error is introduced. The large number of states affects mainly the evaluation of  $P(C_i | T_i)$ , which is needed for the calculation of  $P(C_i | C_{i-1})$ , i.e., the elements of the transition probability matrix for the system backlog (see eq. (--)). There is no effect on the evaluation of  $P(T_i | C_{i-1})$  for values of  $M$  that are larger than the largest value of  $T_i$  that can be accommodated. **EXPLAIN**

*Discuss  
memory  
requirement.  
array sizes*

It is possible to consider an alternative technique for the evaluation of the state probabilities, that is expected to permit the performance evaluation of larger systems. This technique is based on the procedure used to enumerate the states, which was presented in Section 3. The set of states can be viewed as a  $T$ -dimensional Markov chain that evolves conceptually as we consider the users being added to the system one-by-one. The transition probabilities for this system, which are defined as  $P(\mathbf{n}_{T+1} | \mathbf{n}_T)$ , are easily determined as follows:

We consider the states generated by the addition of the  $(T+1)^{\text{st}}$  user. If he has chosen a slot not previously chosen by any of the first  $T$  users, the generation of such states is defined by eq. (7). The probability that the  $(T+1)^{\text{st}}$  user chooses a new slot is given by  $1 - \frac{m}{L}$ , where  $m$  is the number of slots occupied so far by the first  $T$  users. Therefore, we have,

$$\begin{aligned} P[(n_{T+1}(1)+1, n_{T+1}(2), n_{T+1}(3), \dots, n_{T+1}(T+1)) | (n_T(1), n_T(2), n_T(3), \dots, n_T(T))] \\ = 1 - \frac{m}{L}. \end{aligned} \quad (52)$$

If the  $(T+1)^{\text{st}}$  user has chosen one of the slots already chosen by one of the first  $T$  users, the generation of new states is defined by eq. (8). Since there are  $n_T(t)$  slots already containing  $t$  packets, the probability that the  $(T+1)^{\text{st}}$  user picks a slot already containing  $t$  packets is  $\frac{n_T(t)}{L}$ . Therefore, we have,

$$\begin{aligned} P[(n_{T+1}(1)-1, n_{T+1}(2)+1, n_{T+1}(3), \dots, n_{T+1}(T+1)) | (n_T(1), n_T(2), n_T(3), \dots, n_T(T))] \\ = \frac{n_T(1)}{L}, \end{aligned} \quad (53)$$

$$P[(n_{T+1}(1), n_{T+1}(2)-1, n_{T+1}(3)+1, \dots, n_{T+1}(T+1)) | (n_T(1), n_T(2), n_T(3), \dots, n_T(T))]$$

$$= \frac{n_T(2)}{L}, \quad (54)$$

and similarly for the remaining state transitions. The state probabilities corresponding to a specific value of  $T$  are determined by iteration, beginning with the initial condition  $P[(n_1(1)] = 1$  for  $T = 1$ . Although this alternative technique does not produce closed form expressions for the state probabilities, it has the distinct advantage of involving simple and few computations, and of permitting control of the roundoff error by appropriate compensation at each transition.

This Markov chain approach may permit the numerically accurate analysis of larger systems. However, it, too, will be eventually subject to computational limitations. Therefore, it is useful to consider a truncated state space, which may be especially well suited for capture models in which the capture probability is negligible when the number of packets in a slot is greater than or equal to some threshold; we designate this threshold as  $\tau^+$ .

For a given state  $\mathbf{n}_T = (n_T(1), n_T(2), \dots, n_T(T))$ , we consider the truncated state

$$\hat{\mathbf{n}}_T = (n_T(1), n_T(2), \dots, n_T(\tau^+)), \quad (55)$$

where

$$n_T(\tau^+) = \sum_{t=\tau}^T n_T(t). \quad (56)$$

The transition probabilities are the same as those discussed above, except that

$$\begin{aligned} P[(n_{T+1}(1), n_{T+1}(2), n_{T+1}(3), \dots, n_{T+1}(\tau^+)+1) | (n_T(1), n_T(2), n_T(3), \dots, n_T(\tau^+))] \\ = \frac{n_T(\tau^+)}{L}. \end{aligned} \quad (57)$$

For sufficiently small values of  $\tau^+$ , the number of states can be reduced greatly, thus facilitating the evaluation of the relevant state probabilities.

## 9. A SYSTEM WITH MULTIPLE RECEIVERS

Thus far, we have assumed that there is a single spread-spectrum code used by all network members, and that the destination can successfully receive at most one packet in a time slot. It is straightforward to extend our model to the case in which each network member uses a distinct



spread-spectrum code, and in which the destination monitors each of these codes simultaneously. With some signaling schemes, it is in fact possible to monitor many codes simultaneously without a massive proliferation of hardware [--]. If the codes were truly orthogonal, it would be possible for all network members to transmit successfully at the same time. However, since in practice only a quasi-orthogonality can generally be maintained, the probability that a packet is successful depends on the number of other packets transmitted in the same time slot. We define

$$Q(m, j) = Pr(m \text{ packets are received successfully} \mid j \text{ packets transmitted in slot}).$$

This probability depends on the characteristics of the spread spectrum signaling scheme that is being used. The generating function of  $S$  (see eq. (-)) is now given by

$$G(z) = \prod_{j=1}^T \left[ \sum_{m=0}^j z^m Q(m, j) \right]^{n_T(j)}. \quad (58)$$

The rest of the derivation is identical to that for the single-code case presented earlier. Note that the performance results for the case in which all codes can be simultaneously monitored would provide upper bounds for the more realistic case in which only a few codes can be so monitored. Such bounds would represent the maximum throughput that can be achieved under the Framed ALOHA protocol for a given signaling format (i.e., modulation scheme, spread spectrum channel bandwidth, and forward error control scheme) in the absence of equipment limitation constraints.

With some signaling structures, a distinct receiver is needed for each spread-spectrum code that is to be monitored. In such cases, the destination would be able to monitor only a limited number of codes at a time. We can consider a system in which the transmitting network members first choose a slot at random and then choose one of the codes at random to use in that slot. To analyze such a system, we would have to determine the distribution of the number of users that choose the same code, given the number that have chosen the same slot. To do so would require the application of the same technique used in this paper to determine the distribution of the number of users that choose the same slot. Thus, this technique would be used twice, first to determine the occupancy distribution of packets transmitted in slots, and then to determine the distribution of the number of transmitters that use the same code. The probability

throughput remains fairly constant over  $\phi$ . In the case of dynamically varying  $L_i$ , the throughput reaches its peak at a lower value of  $\phi$ , and the traffic peaks at about one packet per slot. This is due to the fact that  $L_i$  is allowed to vary widely (between 1 and  $M$ ), while  $X_i$  is much more restricted (between 0.1 and 1.0). Slight variations upward and downward can be seen in the curves of figures 16 and 17. This is due to the fact that  $L_i$  can take on only integer values.

Figures 18 through 21 show throughput versus  $\phi$  and traffic versus throughput as  $X$  is held constant at values between 0.1 and 1.0 for the case in which capture is considered. The capture model used here is  $Q(1) = 1$ ,  $Q(j) = (.9)^j$   $j \geq 2$ . These curves are similar to those in figures 6 through 13 except that the peak throughput and the optimum value of  $X$  are higher than the previous case of no capture. This is to be expected since colliding packets are not necessarily lost, and therefore can contribute to the throughput. Figures 22 and 23 show throughput versus  $\phi$  and traffic versus throughput for the case of  $Q(1) = 1$ ,  $Q(j) = (.9)^j$   $j \geq 2$  when  $X_i$  is allowed to vary as a function of channel backlog. As is expected, the throughput reaches a maximum and remains constant, and does not decrease with increasing traffic. Finally, figure 24 shows traffic versus throughput for several capture models with  $X_i$  held fixed at 1.

## 11. Conclusions

We have presented a new combinatorial technique for the analysis of framed slotted contention-based multiple access protocols. We have demonstrated its applicability to both uncontrolled and dynamically controlled versions of a Framed ALOHA scheme. Furthermore, a variety of models for capture, both probabilistic and deterministic, can be analyzed using this technique.

Revise

## References

1. H. Okada, Y. Nakanishi, and Y. Igarashi, "Analysis of Framed ALOHA Channel in Satellite Packet Switching Networks," Proceedings of the Fourth International Conference on Computer Communication, pp. 617-622, September 1978.
2. I. Rubin, "Group Random-access Disciplines for Multi-access Broadcast Channels," IEEE Trans. Information Theory, **IT-24**, No. 5, pp. 578-592, September 1978.
3. F. C. Schoute, "Dynamic Frame Length ALOHA," IEEE Trans. on Communications, **COM-31**, No. 4, pp. 565-568, April 1983.
4. W. Szpankowski, "Packet Switching in Multiple Radio Channels: Analysis and Stability of a Random Access System," Computer Networks, **7**, pp. 17-26, 1983.
5. D. H. Davis and S. A. Gronemeyer, "Performance of Slotted ALOHA Random Access with Delay Capture and Randomized Time of Arrival," IEEE Trans. Commun., **COM-28**, pp. 703-710, May 1980.
6. L. Fratta and D. Sant, "Some Models of Packet Radio Networks with Capture," Proceedings of the Fifth International Conference on Computer Communication, pp. 155-161, October, 1980.
7. R. D. Rosner, *Packet Switching: Tomorrow's Communications Today*, Lifetime Learning Publications, 1982.
8. C. Berge, *Principles of Combinatorics*, Academic Press, 1971.
9. W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 1, Third Edition, Johns Wiley Sons, Inc. 1970.
10. J. E. Wieselthier and A. Ephremides, "A New Class of Protocols for Multiple Access in Satellite Networks," IEEE Trans. on Automatic Control, **AC-25**, pp. 865-879, October 1980.
11. G. Fayolle, E. Gelenbe, and J. Labetoulle, "Stability and Optimal Control of the Packet Switching Broadcast Channel," J. Assoc. Computing Machinery, **24**, pp. 375-386, July 1977.
12. S. S. Lam and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," IEEE Trans. on Communications, **COM-23**, pp. 891-904, September 1975.

## References

- [1] J. E. Wieselthier, and A. Ephremides, "A Combinatorial Technique for the Analysis of Framed Contention-Based Multiple Access Protocols", 18th Annual Conference on Information Sciences and Systems, Princeton University, March 1984.