AD-A113 840 MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMA--ETC F/G 17/2
A CLASS OF EFFICIENT CONTENTION RESOLUTION ALGORITHMS FOR MULTI--ETC(U)
MAR 82 J MOSELY, P HUMBLET N00014-75-C-1183
UNCLASSIFIED LIDS-P-1194 NL

1 OF 1
AD A
113840

END
DATE
FILMED
5-82
DTIC

AD A113840

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A113840	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Class of Efficient Contention Resolution Algorithms for Multiple Access Channels		5. TYPE OF REPORT & PERIOD COVERED Paper
		6. PERFORMING ORG. REPORT NUMBER LIDS-P-1194
7. AUTHOR(s) Jeannine Mosely, Pierre Humblet		8. CONTRACT OR GRANT NUMBER(s) ARPA Order No. 3045/5-75; ONR/N00014-75-C-1183
9. PERFORMING ORGANIZATION NAME AND ADDRESS M.I.T. Laboratory for Information and Decision Systems, Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10; ONR Identifying No..049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency, 1400 Wilson Boulevard, Arlington, Virginia 22209		12. REPORT DATE March 1982
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research, Information Systems Program, Code 437, Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

March 1982 LIDS-1194

A CLASS OF EFFICIENT CONTENTION RESOLUTION ALGORITHMS FOR MULTIPLE ACCESS CHANNELS*

by

Jeannine Mosely**

and

Pierre Humblet**

*This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by Defense Advanced Research Projects Agency under Contract No. ARPA ONR/N00014-75-1183 and National Science Foundation under Grant No. NSF ECS 79-19880. This paper is based in part on a paper entitled "Efficient Accessing of a Multi-access Channel" presented at the Conference for Decision and Control, 1980, and in part on J. Mosely's S. M. thesis, submitted to the M.I.T. Department of Electrical Engineering, June 1979.

**The authors are with the M.I.T. Laboratory for Information and Decision Systems, Bldg. 35, M.I.T., Cambridge, MA 02139.

Abstract

A discrete time multiaccess channel is considered where the outcome of a transmission is either "idle", "success" or "collision", depending on the number of users transmitting simultaneously. Messages involved in a "collision" must be retransmitted. An efficient access allocation policy is developed for the case where infinitely many sources generate traffic in a Poisson manner and can all observe the outcomes of the previous transmissions. Its rate of successs is 0.48776. Modifications are presented for the cases where the transmission times depend on the transmission outcomes and where observations are noisy.

I. Introduction

We consider the following model of a multiple access channel. A large number of sources generate messages in a Poisson manner, at a total rate of λ messages per unit of time, starting at time 0. Once a message has been generated, its source can transmit it on a common channel. Transmissions can only start at integer multiples of the unit of time and last one unit of time, also called a "slot". If the transmissions from two or more sources overlap, a collision is said to occur, all messages are lost and must be retransmitted at a later time. If only one source transmits, the transmission is successful.

All sources can observe the channel and learn at the end of a slot whether it is idle, or if a success or a collision has occured. This common feedback is the only information the sources share. The problem is to find an effective way of using the feedback to schedule the transmission of the messages.

The previous model is an idealization of practical communication systems [1], [2], [3] that have been the object of numerous papers in the communication theory literature [4],[5]. Similar problems have also been treated in control theory journals [6],[7],[8], indeed they are nice examples of distributed control. Algorithms similar to the ones presented here have also been derived independently by Tsybakov and Mikhailov [9].

In Section II we will present the basic algorithm and some of its properties. In Section III we show how to analyze the algorithm in order to optimize the throughput, i.e., the maximum long term rate of success. In Section IV we discuss the implementation of the algorithm in real time. In Section V we introduce the general form of a first come first served algorithm. We then show how to modify and analyze the algorithm if the transmission times depend on the transmission outcomes (Section VI) or if the feedback is noisy (Section VII).

II. The Basic Algorithm

The algorithm defined below allows the transmission of the messages on the basis of their generation times. It has the advantage of being effective no matter the number of sources, even infinite, and is a generalization of the procedure presented in [10], which is itself based on an idea from Hayes [11] and Capetanakis [12]. This idea had been used previously for other applications as described in [13].

We abstract the problem as follows: messages are generated according to a Poisson point process with rate λ on $R^{+}=[0,\infty)$. At each step, the algorithm designates a subset of the half-line, and messages generated in that subset are transmitted. This transmission subset is chosen as a fuction of the history of the outcomes of all previous transmissions. The process is repeated ad infinitum. The rate at which successes are produced is called the "throughput" of the algorithm.

In our algorithm, the set of messages that are transmitted during the nth time unit interval are those generated in a time interval of the form [a,b). This interval will be referred to as the "transmission interval". At each step of the algorithm, we update three parameters, y_n, s_n, and t_n, which characterize the state of the algorithm. These parameters are used to calculate a and b, the endpoints of the transmission interval. Specifically, the transmission interval is given by $[a,b) = [y_n, y_n + F(s_n,t_n))$, where F is a given function to be optimized below. F has the properties that it maps $R^+ \cup \{\infty\} \times R^+ \cup \{\infty\}$ into R^+ and $F(s,t) < t$.

The values of y_0, s_0, and t_0 are initially equal respectively to $0, \infty$ and ∞, and the y_n's, s_n's and t_n's $(s_n \le t_n)$ are updated by the following rule, where $F_n = F(s_n,t_n)$.

If a transmission results in

idle: $y_{n+1} \leftarrow y_n + F_n$

$s_{n+1} \leftarrow s_n - F_n$

$t_{n+1} \leftarrow t_n - F_n$

success: $y_{n+1} \leftarrow y_n + F_n$

$s_{n+1} \leftarrow t_n - F_n$

$t_{n+1} \leftarrow \infty$

collision: $y_{n+1} \leftarrow y_n$

$s_{n+1} \leftarrow \min(s_n, F_n)$

$t_{n+1} \leftarrow F_n$

As an example of how this algorithm works, consider Figure 1. Here $F(\infty,\infty)=\tau$, where τ is some finite constant, $F(s,\infty)=s$ and $F(s,s)=s/2$ when s is finite. With the initial conditions for y,s and t given above it is easy to verify that for this F, the pair of parameters (s,t) will always be one of these three forms (i.e. (∞,∞), (s,∞) or (s,s)). This is Gallager's binary splitting algorithm [10].

In Fig. 1.a, the time line is divided into unit intervals and observation of the process begins at a time n, such that $s_n=t_n=\infty$. The numbers written above each slot indicate the transmission outcomes for that slot: 0 represents an idle, 1 a successful transmission, and ≥ 2 a collision. Figs. 1.b-h show the sample process of message generation times that gives rise to the transmission outcome sequence of Fig. 1.a, together with the sequence of transmission intervals selected by the algorithm. In the nth slot, there is a success, and the algorithm moves the transmission interval forward as shown. In the (n+1)st slot there is a collision, and so the transmission interval is split in half and the first half tested for messages. Because there is an idle in slot n+2, this implies that the colliding messages were both generated in the second half of the (n+1)st transmission interval. Hence, in slot n+3, it is desirable to examine only the third quarter of that interval. Since another collision occurs, this interval is split in two and the first half tested, yielding a success in slot n+4. Now it is known that the second half of the transmission interval for slot n+3 contains at least one message. Using it as the transmission interval for slot n+5 produces another success. At this time it can be observed that all messages generated between y_n and y_{n+6} have been successfully transmitted and all conflicts have been resolved. For slot n+6 the algorithm selects the transmission interval of length τ as shown.

It should be noted that the algorithm used in this example is a special case of the algorithm described above, and the choice of F is sub-optimal with respect to throughput. It is not always desirable to divide a transmission interval containing a collision exactly in half. Also, when an interval is known to contain at least one message generation time, there exist conditions such that that interval is not the best choice for the transmission interval. These issues are discussed in section III.

Returning to our general algorithm, we make some assertions about the message generation times. Given the outcomes of the n past tranmissions, we know that all messages generated in $[0,y_n)$ have been successfully transmitted. To see this, note that the monotonic sequence of times $y_0, y_1, \ldots y_n$ divides the interval $[0,y_n)$ into the sequence of intervals $[y_0,y_1), [y_1,y_2), \ldots,$ $[y_{n-1},y_n)$, such that each of these intervals contains exactly 0 or 1 generation time. (Whenever there is a collision, $y_{n+1}=y_n$ and the "interval" $[y_n,y_{n+1})$ is empty.) Hence at time n, all messages generated prior to y_n must have been successfully transmitted.

It can also be shown that, given the outcomes of the n past transmissions, the generation processes in $[0,y_n)$, $[y_n,y_n+t_n)$ and $[y_n+t_n,\infty)$ are independent. The generation times in $[y_n,y_n+t_n)$ are distributed according to a Poisson process with rate λ, conditioned on the facts that there is at least one generation time in $[y_n,y_n+t_n)$, and at least two generation times in $[y_n,y_n+t_n)$. The generation times in $[y_n+t_n,\infty)$ are distributed according to a Poisson process with rate λ.

The proof of these facts is intuitively straightforward when each possible case is considered separately. For example, consider the case where the generation times in $A=[y_n,y_n+t_n)$ are Poisson conditioned on A containing a conflict and the generation times in $[y_n+t_n,\infty)$ are known to be Poisson. Then, if the transmission interval $[y_n,y_n+F_n)$ is found to contain a conflict, it is easy to show that the generation times in $[y_{n+1},y_{n+1}+t_{n+1})=[y_n,y_n+F_n)$ are Poisson conditioned on that interval containing a conflict. Furthermore, the generation times in $[y_{n+1}+t_{n+1},\infty)=[y_n+F_n,\infty)$ are Poisson. To see this, note that $[y_n+F_n,\infty) = [y_n+F_n,y_n+t_n)\cup[y_n+t_n,\infty)$. The generation times in $[y_n+t_n,\infty)$ were asssumed to be Poisson, and the generation times in $[y_n+F_n,y_n+t_n)$ can be shown to be Poisson by the following argument. We use "k in [x,y)" as an abbreviation for "the event that there are k generation times in the interval [x,y)" and "≥2 in [x,y)" as an abbreviation for "the event that there are at least 2 generation times in [x,y)." Then,

$$
\begin{aligned}
&\Pr(k \text{ in } [y_n+F_n,y_n+t_n) \mid \geq 2 \text{ in } [y_n,y_n+F_n), \geq 2 \text{ in } [y_n,y_n+t_n))\\
&=\Pr(k \text{ in } [y_n+F_n,y_n+t_n), \geq 2 \text{ in } [y_n,y_n+F_n), \geq 2 \text{ in } [y_n,y_n+t_n))\\
&\qquad /\Pr(\geq 2 \text{ in } [y_n,y_n+F_n), \geq 2 \text{ in } [y_n,y_n+t_n))\\
&=\Pr(k \text{ in } [y_n+F_n,y_n+t_n), \geq 2 \text{ in } [y_n,y_n+F_n))/ \Pr(\geq 2 \text{ in } [y_n,y_n+F_n))\\
&=\Pr(k \text{ in } [y_n+F_n,y_n+t_n)).
\end{aligned}
$$

Hence $[y_{n+1}+t_{n+1},\infty)$ is the union of two disjoint intervals, each of which contains message generation times distributed according to independent Poisson processes.

In order to make a rigorous statement oi these assertions, a few definitions are needed. Define: $A_n=[0,y_n)$, $B_n=[y_n,y_n+s_n)$, $C_n=[y_n,y_n+t_n)$, $D_n=[y_n+t_n,\infty)$ and $T_n=[y_n,y_n+F_n)$. Let N(S) be the number of generation times in a set S. Let

$$\theta_n=\begin{cases}0 & \text{if } N(T_n)=0\\ 1 & \text{if } N(T_n)=1\\ 2 & \text{if } N(T_n)\geq 2\end{cases}$$

Let $\theta(n)=(\ _1,\ldots,\ _n)$ and for convenience define $\theta(0)=\theta(-1)=\Omega$, the set of all sample processes. (This is done so that we may condition on the events $\theta(0)$ and $\theta(-1)$.) If we have an n-vector of sets $\underline{S}=(S_1,\ldots,S_n)$, for any set A, let $A\cap\underline{S}$ denote the n-vector $(A\cap S_1,\ldots,A\cap S_n)$ and let $\underline{N}(\underline{S})$ denote $(N(S_1),\ldots,N(S_n))$. We may now state the following :

Theorem: For any integers N_A, N_C, N_D, choose measurable finite subsets $A_{n_i}\subset A_n$ for $i=1,\ldots,N_A$, $C_{n_j}\subset C_n$ for $j=1,\ldots,N_C$, $D_{n_k}\subset D_n$ for $k=1,\ldots,N_D$. Then for any vectors $\underline{m}\in Z^{N_A}$, $\underline{p}\in Z^{N_C}$, $\underline{q}\in Z^{N_D}$,

$$\Pr(\underline{N}(\underline{A}_n)=\underline{m},\underline{N}(\underline{C}_n)=\underline{p},\underline{N}(\underline{D}_n)=\underline{q}\mid\theta(n-1))$$
$$=\Pr(\underline{N}(\underline{A}_n)=\underline{m}\mid\theta(n-1))\Pr(\underline{N}(\underline{C}_n)=\underline{p}\mid N(B_n)\geq 1,N(C_n)\geq 2)$$
$$\Pr(\underline{N}(\underline{D}_n)=\underline{q}) \qquad \{1\}$$

for all $n=0,1,\ldots$.

The proof of this theorem is given in the Appendix.

In the next section we will show how to define $F(-,-)$ so as to maximize the rate of successful transmission.

III. Analysis and Optimization

The key to the analysis of the algorithm is to realize that the process (s_n, t_n) is Markovian, as the probabilities of the different outcomes of the (n+1)st transmission and the values of (s_{n+1}, t_{n+1}) depend only on s_n and t_n. This is a direct result of the theorem stated above, since the transmission interval T_n is a subset of C_n.

We should notice the peculiar role of the (∞,∞) state. Physicallly it corresponds to all messages generated before y_n having been successfully transmitted and no information except the a priori statistics being available about generation times greater than y_n. That state is entered every time two transmissions result in a success without an intervening conflict. Thus it is reachable from all other states.

Moreover, if $F(-,-)$ is such that the probability of successful transmission in any state (s,t) has positive lower bound (this is always the case for the $F(-,-)$'s considered below), then state (∞, ∞) is positive recurrent along with only countably many other states accessible from it. Thus the computation of stationary state probabilities and expected values, with a given degree of precision, is a straighforward numerical matter.

We will now direct our attention to the problem of selecting $F(-,-)$ to maximize the long term rate of success, i.e., $\liminf_{N\to\infty}\frac{1}{N}(\sum_{i=0}^{N} r(i))$, where $r(i)$ is equal to one if the ith transmission is successful, and zero otherwise.

Throughout the analysis that follows, the parameters s_n, t_n and F_n are taken to be normalized, that is, the units in which they are measured are chosen such that the generation rate of the messages is 1.

We find the optimum $F(-,-)$ by the succesive approximation method of solving undiscounted infinite horizon Markovian decision theory problems [14]. That is, we assume that the process will end after N more transmissions and assign to each state a value equal to the expected reward on the next state transition plus the expected value of the subsequent state. That is,

$$
\begin{aligned}
v_{(s,t)}^{(n+1)} = & \max_{F_n(s,t)\geq 0} [\Pr(\theta_n=0 \mid s,t,F_n(s,t))v_{(s-F_n(s,t),t-F_n(s,t))}^{(n)} \\
& +\Pr(\theta_n=1 \mid s,t,F_n(s,t))(1+v_{(t-F_n(s,t),\infty)}^{(n)}) \\
& +\Pr(\theta_n=2 \mid s,t,F_n(s,t))v_{(\min(s,F_n(s,t)),F_n(s,t))}^{(n)}]
\end{aligned}
$$

with $v_{(s,t)}(0)=0$ for all s,t. As N goes to infinity the differences $v_{(s,t)}(N+1)-v_{(s,t)}(N)$ converge to the throughput λ^* and the sequence of functions $F_N(s,t)$ converge to the function $F(s,t)$ that achieves the throughput λ^*.

The value functions $v_{(s,t)}(N)$ and the control functions $F_N(s,t)$ were evaluated numerically for a finite number of points over an appropriately bounded, discretized state space. Details of this work are found in [15]. Several interesting conclusions were reached.

First, the optimal $F(s,t)$ is never greater than s, so that all states (s,t) with $s \neq t$ or $t \neq \infty$ are transient. In addition, although a threshold s_T exists such that for $s>s_T$, $F(s,t)<s$, if the optimal F is used for all transmissions, we can never enter a state where s exceeds this threshold. Hence, for all non-transient states we have $F(s,t)=s$.

The optimal $F(\infty,\infty)$ is 1.275, so that all states with $\infty>t>s>1.275$ are transient.

All that is required now to complete the specifications of $F(-,-)$ are the values of $F(s,s)$ for $0<s<1.275$. These are given in Table 1. Observe that $F(s,s)$ is approximately s/2. Hence, this algorithm is very close to the binary splitting algorithm described in Section II. Indeed, the improvement in the throughput of this algorithm over the other is negligible:

0.48776 versus 0.48711. The binary splitting algorithm would be optimal if whenever a collision occurred the collision were known to involve exactly two messages. But because there is some probability of more than two messages colliding, the optimal $F(s,s)$ is slightly less than $s/2$.

We note, however, that the first remark above (i.e., that the optimal F satisfies $F(s,t) \leq s$) does not hold for finite horizon ($N<\infty$) problems. For these, the optimal $F(s,\infty)$ may be larger than s for small N. The optimal $F_N(s,\infty)$ is shown in Figure 2 for N=3, 4 and 5. We note that for each N, there is a large discontinuity in F, i.e., a threshold $s_T(N)$ such that for $s<s_T(N)$, $F_N(s,\infty)>s$ and for $s>s_T(N)$, $F_N(s,\infty)=s$. The threshold descreases with increasing N, becoming smaller than the grid size (.01) of the discretized state space for $N>5$. No similar behavior was observed for $F(s,t)$, $t<\infty$, probably because the numerical optimization did not consider (transient) states in the region where the phenomenon would occur.

The existence of this discontinuity in F is suprising and the reason for it is worth discussing. In Figure 3 we see the value functions at N=3 plotted as a function of F for three states in the neighborhood of the threshold. Each of these functions have two local maxima, one at F=s and one for F>s. For the state $(.06,\infty)$ the maximum occurs at F=.3. For $(.07,\infty)$ the two maxima are equal. For $(.08,\infty)$, the maximum is at F=.08. Hence we have a threshhold at s=.07.

As mentioned in the introduction, a similar algorithm has been presented independently by Tsybakov and Mikhailov [9]. Their version is somewhat more restrictive than ours, as they impose the condition that $F(s,\infty)=s$. That is, when an interval is known to contain at least one generation time, that interval is chosen as the next transmission interval. Hence, the only recurrent states have the form (s,s), (s,∞) and (∞,∞), exactly as in Gallager's binary splitting algorithm. Thus, only $F(\infty,\infty)=\tau$ and $F(s,s)$ for $s\leq\tau$ need to be determined. They do not actually find the optimal F, but claim that if $F(\infty,\infty)=1.275$, $F(1.275,1.275)=.584$ and $F(s,s)=.485s$ for all $s\leq.584$, then the throughput is .48778. Using the same values for $F(-,-)$, we calculate a throughput of .48773. This discrepancy is unexplained, but since the results differ only in the fifth decimal place, is not very important.

Note that the optimal algorithm in the class we consider appears to lie in the subclass considered in [9].

IV. Real Time Implementation

In the idealized version of Section III it was assumed that all messages were generated before the algorithm started. In practice, generations and transmissions would take place concurrently. Hence, the original algorithm is not causal, in the sense that it sometimes specifies that messages should be transmitted before having been generated. This can be remedied by defining

$$F_n(s_n,t_n,y_n)=\min[F(s_n,t_n),n-y_n] \qquad \{2\}$$

The quantity $n-y_n$ that appears above we call the lag of the algorithm. That is, the lag is length of time during which messages have already been generated but not yet successfully transmitted.

To analyze real time performances, like message delay, one must study the Markov process (s_n,t_n,y_n), as the process (s_n,t_n) is no longer Markovian. This appears to be extremely complicated when the boundary condition {2} is imposed. However, some simple statements can be made regarding the behavior of the lag $n-y_n$ as a function of λ.

Let k(m) denote the time when $(s,t)=(\infty,\infty)$ for the nth time (that is, $k(m)=\min\{k|k>k(m-1),\ (s_k,t_k)=(\infty,\infty)\}$). Note that if the probability of success at a step has a positive lower bound, the random variable $k(m)-k(m-1)$ has a geometric tail distribution, and $E[k(m)-k(m-1)]$ is finite. Moreover, the "drifts" $y_{k(m)}-y_{k(m-1)}$ are independent. By a renewal argument, we can show

$$E(y_{k(m+1)}-y_{k(m)})=\frac{\lambda^*}{\lambda} E(k(m+1)-k(m))$$

where λ^* is the throughput of the algorithm. This follows since the throughput is the expected number of message generations in $y_{k(m+1)}-y_{k(m)}$ divided by the expected number of trials $E(k(m+1)-k(m))$. This holds for both the unconstrained case, where $\lambda^*=.4877$, and the constrained case.

Now, the expected difference in the lag at times $y_{k(m+1)}$ and $y_{k(m)}$ is

$$E[k(m+1)-k(m)] - E[y_{k(m+1)}-y_{k(m)}]=(1-\lambda^*/\lambda)E[k(m+1)-k(m)].$$

Hence, in the idealized version, as long as $\lambda<.4877$, the expected changes in lag are negative, and the algorithm will repeatedly select transmission intervals $[y_n,y_n+F_n)$ where $y_n+F_n>n$.

When {2} holds, it is easy to see that $\lambda^*=\min(\lambda,.4877)$. Clearly $\lambda^*\le\lambda$. If $\lambda^*<\lambda$, then the expected change in lag from $y_{k(m)}$ to $y_{k(m+1)}$ is positive and the lag increases without bound as n goes to infinity. But whenever the lag is greater than or equal to $\tau=F(\infty,\infty)$, the choice of F_n is the same as for the unconstrained algorithm, and the throughput is .4877. Hence, when $\lambda>.4877$, $\lambda^*=.4877$ and the lag goes to infinity.

When $\lambda\le.4877$, $\lambda^*=\lambda$, and the expected change in lag is 0. Furthermore, from the above observations (i.e., the facts that k(m)-k(m-1) has a geometric tail distribution and the expected change in lag is negative given that the lag exceeds τ) and the results in Hajek [16], we can show that the probability that the lag is greater than x has an upper bound exponentially decreasing with x.

It is reassuring to note that, even when the generation rate of the messages exceeds the throughput of the algorithm, it will continue to transmit successfully at its maximum throughput.

V. The General FCFS Algorithm

We note that the algorithm is first-come first-served (FCFS), although it is not the most general FCFS algorithm. The results of the Section III suggest some conjectures with regard to the most general FCFS algorithm, which we describe in this section.

For an algorithm to be FCFS, it must satisfy one of two conditions. Either it does not allow a message to be transmitted when other messages with earlier generation times must wait, or if it does, the probability of successful transmission must be zero.

Suppose that we are using a FCFS algorithm to resolve conflicts and that all messages with generation times prior to y_n have been successfully transmitted. We will call the set of messages generated after y_n the "queue". The algorithm selects some subset of the queue to transmit in the next slot, which we call the transmission set. Then the most general form of a transmission set which satisfies the first condition will clearly be the set of all messages generated in an interval of the form $[y_n, y_n+F_n]$.

Now let us consider transmission sets satisfying the second condition. Suppose we have a subset of the queue, S_1, which is known to contain at least one message, but it is not known whether it contains more. Consider the transmission set which is the union of S_1 and some subset of the queue, S_2. If the subset of S_2 which is disjoint from S_1 is not empty, then a conflict occurs and the second condition holds. If the subset of S_2 which is disjoint from S_1 is empty, then S_1 must be the set of messages generated in an interval of the form $[y_n, y_n+F_n]$ for the first condition to hold. Assume that the algorithm never chooses a transmission set that is known to contain a conflict, we cannot know in advance if the subset of S_2 disjoint from S_1 is non-empty. Hence, S_1 must be the set of messages generated in an interval of the form $[y_n, y_n+F_n]$ to insure that one of the two conditions hold.

The general FCFS algorithm must use transmission sets satisfying one of the two conditions described above. It differs from the basic algorithm of section II only by permitting the use of transmission sets satisfying the second condition. But when we consider that, for our algorithm the optimal $F_n(s,\infty)$ is s; that is, when an interval $[y_n, y_n+F_n)$ is known to contain at least one generation time, the optimal transmission interval is just $[y_n, y_n+F_n)$, it seems unlikely that using a more general transmission set of the form $[y_n, y_n+F_n) \cup S_2$ would offer any improvement. We cannot prove this conjecture, however since allowing this type of transmission set makes it no longer possible to characterize the algorithm as a Markov process with just two (or even a finite number of) state variables.

While the algorithm described in section II is not the most general FCFS, it is the most general algorithm having the property that sources attempt to transmit messages in the order that they were generated.

VI. Unequal Observation Times

In fact many multiaccess communication systems differ from the model introduced in section I in that the times necessary to learn the transmission outcomes depend on the outcomes. We denote by t_0, t_1 and t_2 respectively the times necessary to learn that the channel was idle, or that a success or a collision occured.

For example carrier sense radio systems [2] can detect idles quickly (no carrier present), while they rely on error detecting codes and the transmissions of acknowledgements to distinguish between successes and collisions, thus $t_0 << t_1 = t_2$. In addition, some cable broadcast systems [3] have a listen-while-transmit feature that allows the quick abortion of transmissions resulting in collisions, thus $t_0 = t_2 << t_1$.

The general algorithm outlined in section II and the remarks about its Markovian nature remain valid, but the reward function r(-) and the maximization in section III are not appropriate. A better measure of quality is to minimize the expected time to send a message, i.e.,

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} \sum_{j=0}^{3} t_j \, I(\theta_n = j)}{\sum_{i=1}^{N} r(i)}$$

$$= t_1 + t_2 \lim_{N} \frac{\sum_{i=1}^{N} \frac{t_0}{t_2} I(\theta_i = 0) + I(\theta_i = 2)}{\sum_{i=1}^{N} r(i)}$$

where I(-) denotes the indicator function.

The limit of the expected value in the right hand side can be interpreted as the expected time overhead per message, and depends only on t_0/t_2 for a given F(-,-). It will be denoted by c and should be minimized over F(-,-) for a given t_0/t_2.

The optimization of the general algorithm under this formulation for a large number of values of t_0/t_2 is time consuming. It is greatly simplified if we consider only those $F(-,-)$ such that $F(s,t) \leq s$. The only recurrent states are then of the form (s,s), or (s,∞), see above. Note that the optimal F found in Section III belonged to the restricted class. We will now show how to proceed with the optimization.

By a renewal argument

$$c = \frac{E\left(\sum_{i=1}^{b} \frac{t_0}{t_2} I(\theta_i=0)+I(\theta_i=2)\right)}{E\left(\sum_{i=1}^{b} r(i)\right)}$$

where in the right hand side one assumes that $(s_1,t_1)=(\infty,\infty)$ and b is the time of first return to (∞,∞).

Let us now assume that we guess a value $\hat{c}$ for the minimum of c over all restricted $F(-,-)$, and consider the function

$$v(s,t) = E\left[\sum_{i=1}^{b} \frac{t_0}{t_2} I(\theta_i=0)+I(\theta_i=2)-\hat{c}r(i) \mid (s_1,t_1)=(s,t)\right]$$

Because s_{n+1} is either equal to ∞ or is less than s_n, $V(s,s)$ and $V(s,\infty)$ can be written as convex combinations of $V(s',s')$ and $V(s',\infty)$, $s'<\min(s,F(\infty,\infty))$. It is straightforward [13] to minimize $V(s,s)$ and $V(s,\infty)$ recursively for increasing s, and to obtain the minimum value of $V(\infty,\infty)$.

If the minimum value is 0, $\hat{c}$ was guessed correctly and is the minimum value of c. If the minimum value of $V(\infty,\infty)$ is positive (negative), $\hat{c}$ was guessed too small (large), and the minimization of V(-) must be repeated with a new $\hat{c}$.

The resulting minimum value of c is shown in Figure 4, as a function of t_0/t_2. It is almost equal to the expected time overhead per message for the binary splitting algorithm, with only $F(\infty,\infty)$ is optimized.

VII. Noisy Feedback

The previous algorithm assumed that the transmission outcomes were perfectly observed by all sources. This assumption is critical. One verifies easily that if an idle at time n is falsely observed as a collision then the algorithm will deadlock. The algorithm will behave as if there is a conflict in the interval $[y_n,y_n+t_n)$ and when the next transmission interval $[y_n,y_n+F_n)$ produces an idle, the algorithm will proceed as if there was a conflict in the interval $[y_n+F_n,y_n+t_n)$. Hence, y_n+t_n will remain constant while t_n goes to zero.

D. Ryter [17] has recently examined the problem of noisy feedback, where the noise can cause idles or successes to be observed as collisions. He showed that the binary splitting algorithm [10] outlined in section II can be modified to work properly. The essential modification is the introduction of a threshold value. If t_n is smaller than the threshold, then the algorithm becomes non-stationary, in the sense that it alternates between using F(s,s)=s and F(s,s)=s/2, thus first seeking confirmation that a collision really occured, then trying to resolve it. The analysis and optimization are too long to be reported here. The main result is that with the proper choice of parameters, the throughput behaves roughly like .487-p, where p is the probability of false collision indication.

VIII. Final Comments

The main results of this paper are the description and analsyis of an access algorithm for the channel model described in Section I, with infinitely many sources. Its throughput is .4877, the largest known to this day. Much research has been done to determine upper bounds on the possible throughput [18], [19], [20], [21]. Tsybakov and Mikhailov [22] have recently shown that no algorithm can have a throughput higher than 0.5874, and it is widely believed that the best achievable throughput is in the neighborhood of .5. However, throughputs arbitrarily close to 1 are possible, at the expense of high average message delay, when the number of sources is finite.

We have also shown how the algorithm can be modified in the cases of variable transmission times and noisy feedback. Upper bounds on the throughput for the case of variable transmission times are given by Humblet in [23].

Finally, it should be pointed out that although the algorithm presented here uses the message generation times to specify when they should be transmitted, this is not necessary. Another algorithm can be described, with the same throughput and expected time overhead per message, where sources generate random numbers to determine if they should transmit. Of course, real time properties, like first-generated first-transmitted will not be conserved.

Appendix

Define: $A_n=[0,y_n)$, $B_n=[y_n,y_n+s_n)$, $C_n=[y_n,y_n+t_n)$, $D_n=[y_n+t_n,\infty)$ and $T_n=[y_n,y_n+F_n)$. Let N(S) be the number of generation times in a set S. Let

$$\theta_n=\begin{cases}0 & \text{if } N(T_n)=0\\ 1 & \text{if } N(T_n)=1\\ 2 & \text{if } N(T_n)\geq 2\end{cases}$$

Let $\theta(n)=(\theta_1,\ldots,\theta_n)$ and for convenience define $\theta(0)=\theta(-1)=\Omega$, the set of all sample processes. (This is done so that we may condition on the events $\theta(0)$ and $\theta(-1)$.) If we have an n-vector of sets $\underline{S}=(S_1,\ldots,S_n)$, for any set A, let $A\cap\underline{S}$ denote the n-vector $(A\cap S_1,\ldots,A\cap S_n)$ and let $\underline{N}(\underline{S})$ denote $(N(S_1),\ldots,N(S_n))$. Then:

Theorem: For any integers N_A, N_C, N_D, choose measurable finite subsets $A_{n_i}\subset A_n$ for $i=1,\ldots,N_A$, $C_{n_j}\subset C_n$ for $j=1,\ldots,N_C$, $D_{n_k}\subset D_n$ for $k=1,\ldots,N_D$. Then for any vectors $\underline{m}\in Z^{N_A}$, $\underline{p}\in Z^{N_C}$, $\underline{q}\in Z^{N_D}$,

$$\begin{aligned}&\Pr(\underline{N}(\underline{A}_n)=\underline{m},\underline{N}(\underline{C}_n)=\underline{p},\underline{N}(\underline{D}_n)=\underline{q}\mid\theta(n-1))\\ &\quad=\Pr(\underline{N}(\underline{A}_n)=\underline{m}\mid\theta(n-1))\Pr(\underline{N}(\underline{C}_n)=\underline{p}\mid N(B_n)\geq 1,N(C_n)\geq 2)\\ &\quad\ \Pr(\underline{N}(\underline{D}_n)=\underline{q})\qquad \{1\}\end{aligned}$$

for all $n=0,1,\ldots$.

Proof : For n=0, $A_0=\emptyset$, $B_n=C_n=[0,\infty)$ and $D_n=\emptyset$, and so {1} is trivially true.

Now we proceed by induction. Suppose that {1} holds for n. We will show that {1} holds for $n+1$. For any N_A, N_C, N_D, we choose finite measurable subsets $A_{(n+1)_i} \subset A_{n+1}$ for $i=1,\ldots,N_A$, $C_{(n+1)_j} \subset C_{n+1}$ for $j=1,\ldots,N_C$, $D_{(n+1)_k} \subset D_{n+1}$ for $k=1,\ldots,N_D$. Note that since $\mathcal{E}_n$ is a function of $N(T_n)$ and $T_n \subset C_n$, {1} implies that $\Pr(\mathcal{E}_n | \mathcal{E}(n-1)) = \Pr(\mathcal{E}_n | N(B_n) \geq 1, N(C_n) \geq 2)$. Hence,

$$\begin{aligned}
&\Pr(\underline{N}(\underline{A}_{n+1})=\underline{m}, \underline{N}(\underline{C}_{n+1})=\underline{p}, \underline{N}(\underline{D}_{n+1})=\underline{q} | \mathcal{E}(n)) \\
&\quad = \Pr(\underline{N}(\underline{A}_{n+1})=\underline{m}, \underline{N}(\underline{C}_{n+1})=\underline{p}, \underline{N}(\underline{D}_{n+1})=\underline{q}, \mathcal{E}_n | \mathcal{E}(n-1)) \\
&\qquad\qquad / \Pr(\mathcal{E}_n | \mathcal{E}(n-1)) \\
&\quad = \Pr(\underline{N}(\underline{A}_{n+1})=\underline{m}, \underline{N}(\underline{C}_{n+1})=\underline{p}, \underline{N}(\underline{D}_{n+1})=\underline{r}, \mathcal{E}_n | \mathcal{E}(n-1)) \\
&\qquad\qquad / \Pr(\mathcal{E}_n | N(B_n) \geq 1, N(C_n) \geq 2) \quad \{A.1\}
\end{aligned}$$

Now since $A_{n+1} \subset A_n \cup C_n$, $C_{n+1} \subset C_n \cup D_n$ and $D_{n+1} \subset C_n \cup D_n$, and because A_{n+1}, C_{n+1} and D_{n+1} are disjoint, {A.1} is equal to

$$\begin{aligned}
&\sum_{\hat{m}} \sum_{\hat{p}} \sum_{\hat{q}} \Pr(\underline{N}(\underline{A}_{n+1} \cap A_n)=\hat{m}, \underline{N}(\underline{A}_{n+1} \cap C_n)=\underline{m}-\hat{m}, \\
&\qquad \underline{N}(\underline{C}_{n+1} \cap C_n)=\hat{p}, \underline{N}(\underline{C}_{n+1} \cap D_n)=\underline{p}-\hat{p}, \\
&\qquad \underline{N}(\underline{D}_{n+1} \cap C_n)=\hat{q}, \underline{N}(\underline{D}_{n+1} \cap D_n)=\underline{q}-\hat{q}, \mathcal{E}_n | \mathcal{E}(n-1)) \\
&\qquad\qquad / \Pr(\mathcal{E}_n | N(B_n) \geq 1, N(C_n) \geq 2) \\
&= \sum_{\hat{m}} \sum_{\hat{p}} \sum_{\hat{q}} \Pr(\underline{N}(\underline{A}_{n+1} \cap A_n)=\hat{m} | \mathcal{E}(n-1)) \Pr(\underline{N}(\underline{A}_{n+1} \cap C_n)=\underline{m}-\hat{m}, \\
&\qquad \underline{N}(\underline{C}_{n+1} \cap C_n)=\hat{p}, \underline{N}(\underline{D}_{n+1} \cap C_n)=\hat{q}, \mathcal{E}_n | N(B_n) \geq 1, N(C_n) \geq 2) \\
&\qquad \cdot \Pr(\underline{N}(\underline{C}_{n+1} \cap D_n)=\underline{p}-\hat{p}, \underline{N}(\underline{D}_{n+1} \cap D_n)=\underline{q}-\hat{q}) \\
&\qquad / \Pr(\mathcal{E}_n | N(B_n) \geq 1, N(C_n) \geq 2) \\
&= \sum_{\hat{m}} \sum_{\hat{p}} \sum_{\hat{q}} \Pr((\underline{N}(\underline{A}_{n+1} \cap A_n)=\hat{m} | \mathcal{E}(n-1)) \Pr(\underline{N}(\underline{C}_{n+1} \cap D_n)=\underline{p}-\hat{p} \\
&\qquad \cdot \Pr(\underline{N}(\underline{D}_{n+1} \cap D_n)=\underline{q}-\hat{q}) \Pr(\underline{N}(\underline{A}_{n+1} \cap C_n)=\underline{m}-\hat{m}, \\
&\qquad \underline{N}(\underline{C}_{n+1} \cap C_n)=\hat{p}, \underline{N}(\underline{D}_{n+1} \cap C_n)=\hat{q}, \mathcal{E}_n, N(B_n) \geq 1, N(C_n) \geq 2) \\
&\qquad / \Pr(\mathcal{E}_n, N(B_n) \geq 1, N(C_n) \geq 2) \qquad \{A.2\}
\end{aligned}$$

where the second step follows by the induction hypothesis, and the last step follows by elementary probability theory and the fact that for a Poisson process, arrivals in disjoint sets are independent. Now we evaluate equation {A.2} for two cases: one where $\mathcal{C}_n=0$ or 1 and one where $\mathcal{C}_n=2$.

If $\mathcal{C}_n=0$ or 1, {A.2} is equal to:

$$
\begin{aligned}
&\sum_{\hat{m}}\sum_{\hat{p}}\sum_{\hat{q}} \Pr(\underline{N}(\underline{A}_{n+1}\cap A_n)=\hat{m}\,|\,\mathcal{C}(n-1))\Pr(\underline{N}(\underline{C}_{n+1}\cap D_n)=\underline{p}-\hat{p}) \\
&\quad\cdot\Pr(\underline{N}(\underline{D}_{n+1}\cap D_n)=\underline{q}-\hat{q})\Pr(\underline{N}(\underline{A}_{n+1}\cap C_n)=\underline{m}-\hat{m},\underline{N}(\underline{C}_{n+1}\cap C_n)=\hat{p}, \\
&\quad\underline{N}(\underline{D}_{n+1}\cap C_n)=\hat{q},\mathcal{C}_n,N(B_n\backslash T_n)\geq 1-\mathcal{C}_n,N(C_n\backslash T_n)\geq 2-\mathcal{C}_n) \\
&\quad/\Pr(\mathcal{C}_n,N(B_n\backslash T_n)\geq 1-\mathcal{C}_n,N(C_n\backslash T_n)\geq 2-\mathcal{C}_n) \\
=&\sum_{\hat{m}}\sum_{\hat{p}}\sum_{\hat{q}} \Pr(\underline{N}(\underline{A}_{n+1}\cap A_n)=\hat{m},\mathcal{C}(n-1))\Pr(\underline{N}(\underline{C}_{n+1}\cap D_n)=\underline{p}-\hat{p}) \\
&\quad\cdot\Pr(\underline{N}(\underline{D}_{n+1}\cap D_n)=\underline{q}-\hat{q})\Pr(\underline{N}(\underline{A}_{n+1}\cap C_n)=\underline{m}-\hat{m},\mathcal{C}_n) \\
&\quad\cdot\Pr(\underline{N}(\underline{C}_{n+1}\cap C_n)=\hat{p},N(B_n\backslash T_n)\geq 1-\mathcal{C}_n,N(C_n\backslash T_n)\geq 2-\mathcal{C}_n) \\
&\quad\cdot\Pr(\underline{N}(\underline{D}_{n+1}\cap C_n)=\hat{q})/[\Pr(\mathcal{C}(n-1))\Pr(\mathcal{C}_n)\Pr(N(B_n\backslash T_n)\geq 1-\mathcal{C}_n, \\
&\qquad N(C_n\backslash T_n)\geq 2-\mathcal{C}_n)] \qquad \{A.3\}
\end{aligned}
$$

Here we have made use of the facts that for $\mathcal{C}_n=0$ or 1, $T_n\subset A_{n+1}$, $B_n\backslash T_n\subset C_{n+1}$, $C_n\backslash T_n\subset C_{n+1}$, and, since A_{n+1}, B_{n+1} and C_{n+1} are disjoint, we may use the Poisson assumption to decouple the probabilities as above.

Now if $\mathcal{C}_n=0$, $B_n\backslash T_n=B_{n+1}$ and $C_n\backslash T_n=C_{n+1}$. It is always true that $N(B_n\backslash T_n)\geq 0$. Also, if $\mathcal{C}_n=1$, $C_n\backslash T_n=B_{n+1}$, and since $C_{n+1}=[y_{n+1},\infty)$, $N(C_{n+1})\geq 2$ holds with probability one. So the event $\{N(B_n\backslash T_n)\geq 1-\mathcal{C}_n,\ N(C_n\backslash T_n)\geq 2-\mathcal{C}_n\}$ is equal to $\{N(B_{n+1})\geq 1,$

$N(C_{n+1})\geq 2\}$. Hence {A.3} equals

$$\sum_{\hat{m}}\sum_{\hat{p}}\sum_{\hat{q}} Pr(\underline{N}(\underline{A}_{n+1}\cap A_n)=\hat{m},\underline{N}(\underline{A}_{n+1}\cap C_n)=\underline{m}-\hat{m}\,|\,\hat{\mathcal{E}}(n))$$
$$\cdot Pr(\underline{N}(\underline{C}_{n+1}\cap D_n)=\underline{p}-\hat{p},\underline{N}(\underline{C}_{n+1}\cap C_n)=\hat{p}\,|\,N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$\cdot Pr(\underline{N}(\underline{D}_{n+1}\cap C_n)=\hat{q},\underline{N}(\underline{D}_{n+1}\cap D_n)=\underline{q}-\hat{q})$$
$$=Pr(\underline{N}(\underline{A}_{n+1})=\hat{m}\,|\,\mathcal{E}(n))Pr(\underline{N}(\underline{C}_{n+1})=\underline{p}\,|\,N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$\cdot Pr(N(\underline{D}_{n+1})=\underline{q})$$

which is the desired result.

If $\mathcal{E}_n=2$, the event $\{\mathcal{E}_n,\ N(B_n)\geq 1,\ N(C_n)\geq 2\}$ is equal to $\{N(B_n)\geq 1,\ N(T_n)\geq 2\}$, since $T_n\subset C_n$. If $F_n\leq s_n$, then $T_n\subset B_n$ and $B_{n+1}=C_{n+1}=T_n$, so this event is equal to $\{N(T_n)\geq 2\}$ or $\{N(B_{n+1})\geq 1,$ $N(C_{n+1})\geq 2\}$. If $F_n>s_n$, $B_{n+1}=B_n$, $C_{n+1}=T_n$, and this event is still equal to $\{N(B_{n+1})\geq 1,\ N(C_{n+1})\geq 2\}$. Hence, {A.2} is equal to:

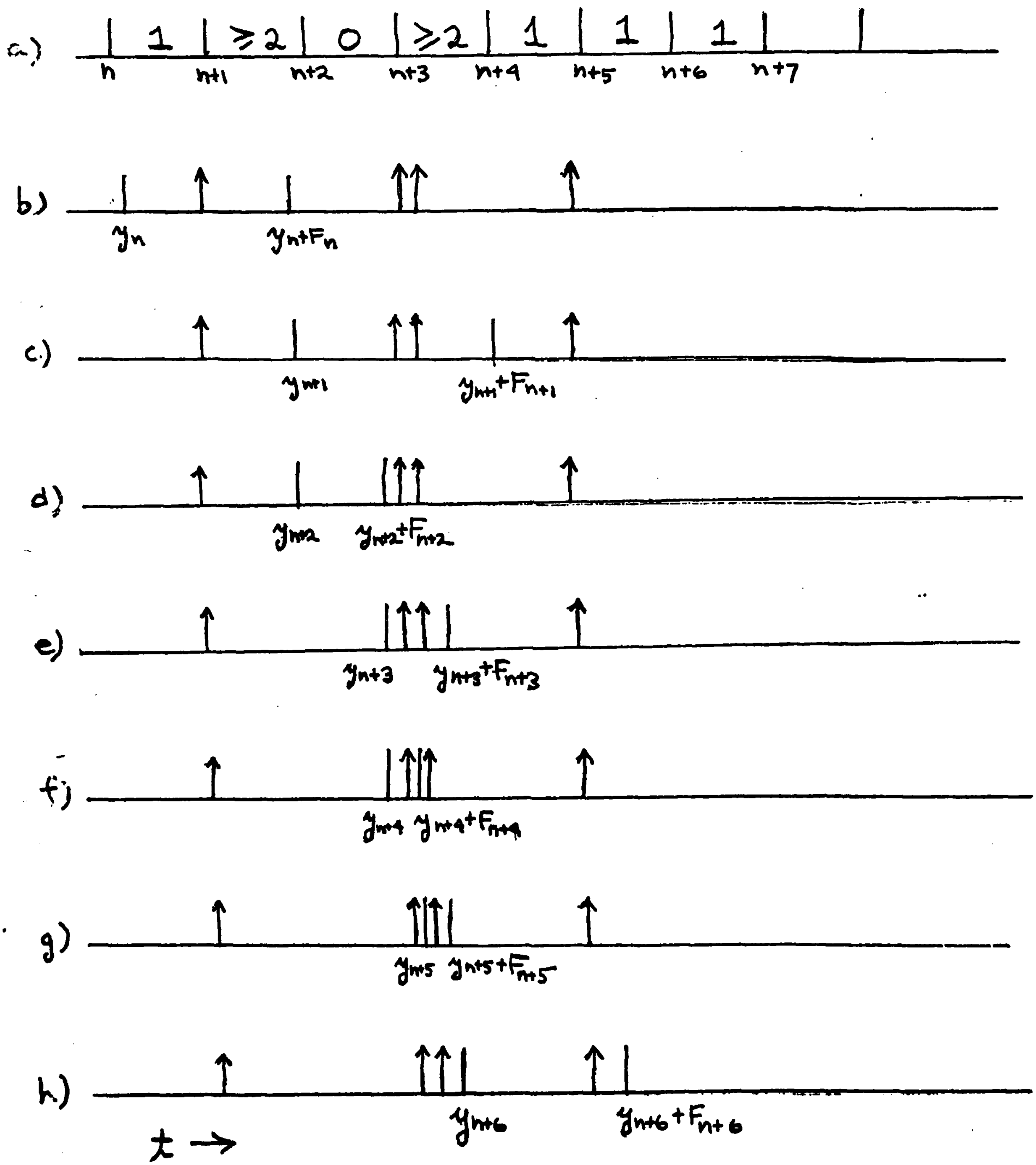
$$\sum_{\hat{m}}\sum_{\hat{p}}\sum_{\hat{q}} Pr(\underline{N}(\underline{A}_{n+1}\cap A_n)=\hat{m}\,|\,\mathcal{E}(n-1))Pr(\underline{N}(\underline{C}_{n+1}\cap D_n)=\underline{p}-\hat{p})$$
$$\cdot Pr(\underline{N}(\underline{D}_{n+1}\cap D_n)=\underline{q}-\hat{q})Pr(\underline{N}(\underline{A}_{n+1}\cap C_n)=\underline{m}-\hat{m})$$
$$\cdot Pr(\underline{N}(\underline{C}_{n+1}\cap C_n)=\hat{p},N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$\cdot Pr(\underline{N}(\underline{D}_{n+1}\cap C_n)=\hat{q})/Pr(N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$=\sum_{\hat{m}}\sum_{\hat{p}}\sum_{\hat{q}} Pr(\underline{N}(\underline{A}_{n+1}\cap A_n)=\hat{m},\mathcal{E}(n-1))Pr(\underline{N}(\underline{A}_{n+1}\cap C_n)=\underline{m}-\hat{m})Pr(\mathcal{E}_n)$$
$$/Pr(\mathcal{E}(n-1))Pr(\mathcal{E}_n)$$
$$\cdot Pr(\underline{N}(\underline{C}_{n+1}\cap D_n)=\underline{p}-\hat{p},\underline{N}(\underline{C}_{n+1}\cap C_n)=\hat{p},N(B_{n+1}\geq 1,N(C_{n+1})\geq 2)$$
$$/Pr(N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$\cdot Pr(\underline{N}(\underline{D}_{n+1}\cap C_n)=\hat{q},\underline{N}(\underline{D}_{n+1}\cap D_n)=\underline{q}-\hat{q})$$
$$=Pr(\underline{N}(\underline{A}_{n+1})=\hat{m}\,|\,\mathcal{E}(n))Pr(\underline{N}(\underline{C}_{n+1})=\underline{p}\,|\,N(B_{n+1})\geq 1,N(C_{n+1})\geq 2)$$
$$Pr(\underline{N}(\underline{D}_{n+1})=\underline{q}).$$

The last steps here follow by noting that the appropriate groups of subsets are disjoint and applying the Poisson assumption.

Hence we have shown that (*) holds for n+1. Q.E.D.

s	F(s,s)	s	F(s,s)	s	F(s,s)
0.01	0.005	0.44	0.214	0.87	0.410
0.02	0.010	0.45	0.218	0.88	0.415
0.03	0.015	0.46	0.223	0.89	0.419
0.04	0.020	0.47	0.228	0.90	0.423
0.05	0.025	0.48	0.232	0.91	0.428
0.06	0.030	0.49	0.237	0.92	0.432
0.07	0.035	0.50	0.242	0.93	0.437
0.08	0.040	0.51	0.246	0.94	0.441
0.09	0.045	0.52	0.251	0.95	0.445
0.10	0.050	0.53	0.256	0.96	0.450
0.11	0.055	0.54	0.260	0.97	0.454
0.12	0.060	0.55	0.265	0.98	0.458
0.13	0.064	0.56	0.270	0.99	0.463
0.14	0.069	0.57	0.274	1.00	0.467
0.15	0.074	0.58	0.279	1.01	0.471
0.16	0.079	0.59	0.284	1.02	0.476
0.17	0.084	0.60	0.288	1.03	0.480
0.18	0.089	0.61	0.293	1.04	0.484
0.19	0.094	0.62	0.297	1.05	0.489
0.20	0.099	0.63	0.302	1.06	0.493
0.21	0.104	0.64	0.307	1.07	0.497
0.22	0.108	0.65	0.311	1.08	0.501
0.23	0.113	0.66	0.316	1.09	0.506
0.24	0.118	0.67	0.320	1.10	0.510
0.25	0.123	0.68	0.325	1.11	0.514
0.26	0.128	0.69	0.329	1.12	0.519
0.27	0.133	0.70	0.334	1.13	0.523
0.28	0.137	0.71	0.338	1.14	0.527
0.29	0.142	0.72	0.343	1.15	0.531
0.30	0.147	0.73	0.347	1.16	0.536
0.31	0.152	0.74	0.352	1.17	0.540
0.32	0.157	0.75	0.357	1.18	0.544
0.33	0.161	0.76	0.361	1.19	0.548
0.34	0.166	0.77	0.365	1.20	0.552
0.35	0.171	0.78	0.370	1.21	0.557
0.36	0.176	0.79	0.374	1.22	0.561
0.37	0.181	0.80	0.379	1.23	0.565
0.38	0.185	0.81	0.383	1.24	0.569
0.39	0.190	0.82	0.388	1.25	0.573
0.40	0.195	0.83	0.392	1.26	0.578
0.41	0.199	0.84	0.397	1.27	0.582
0.42	0.204	0.85	0.401	1.28	0.586
0.43	0.209	0.86	0.406	1.29	0.590

Table 1


a)
1
≥2
0
≥2
1
1
1
n
n+1
n+2
n+3
n+4
n+5
n+6
n+7
b)
y_n
y_n+F_n
c)
y_{n+1}
$y_{n+1}+F_{n+1}$
d)
y_{n+2}
$y_{n+2}+F_{n+2}$
e)
y_{n+3}
$y_{n+3}+F_{n+3}$
f)
y_{n+4}
$y_{n+4}+F_{n+4}$
g)
y_{n+5}
$y_{n+5}+F_{n+5}$
h)
y_{n+6}
$y_{n+6}+F_{n+6}$
$t \rightarrow$

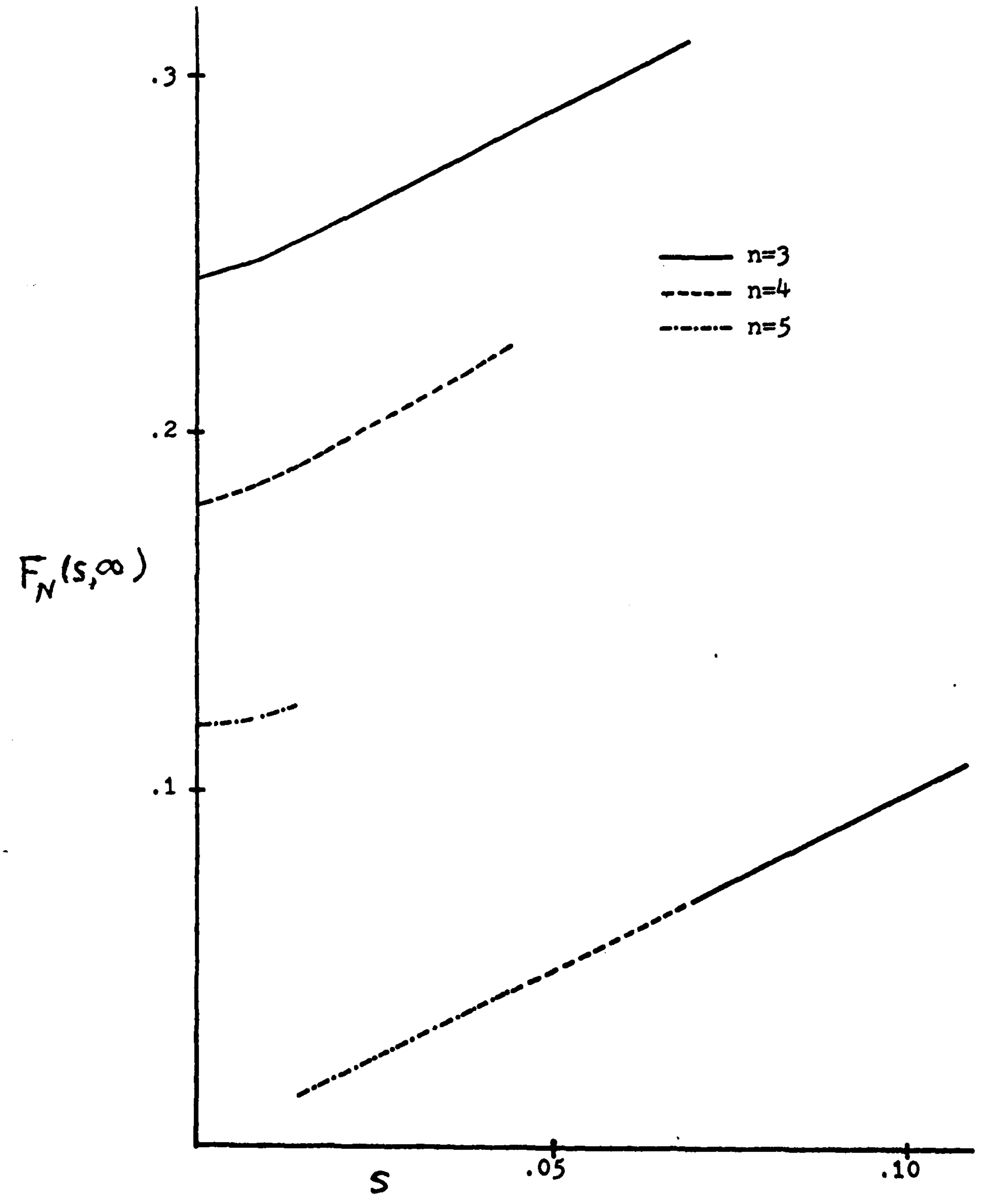
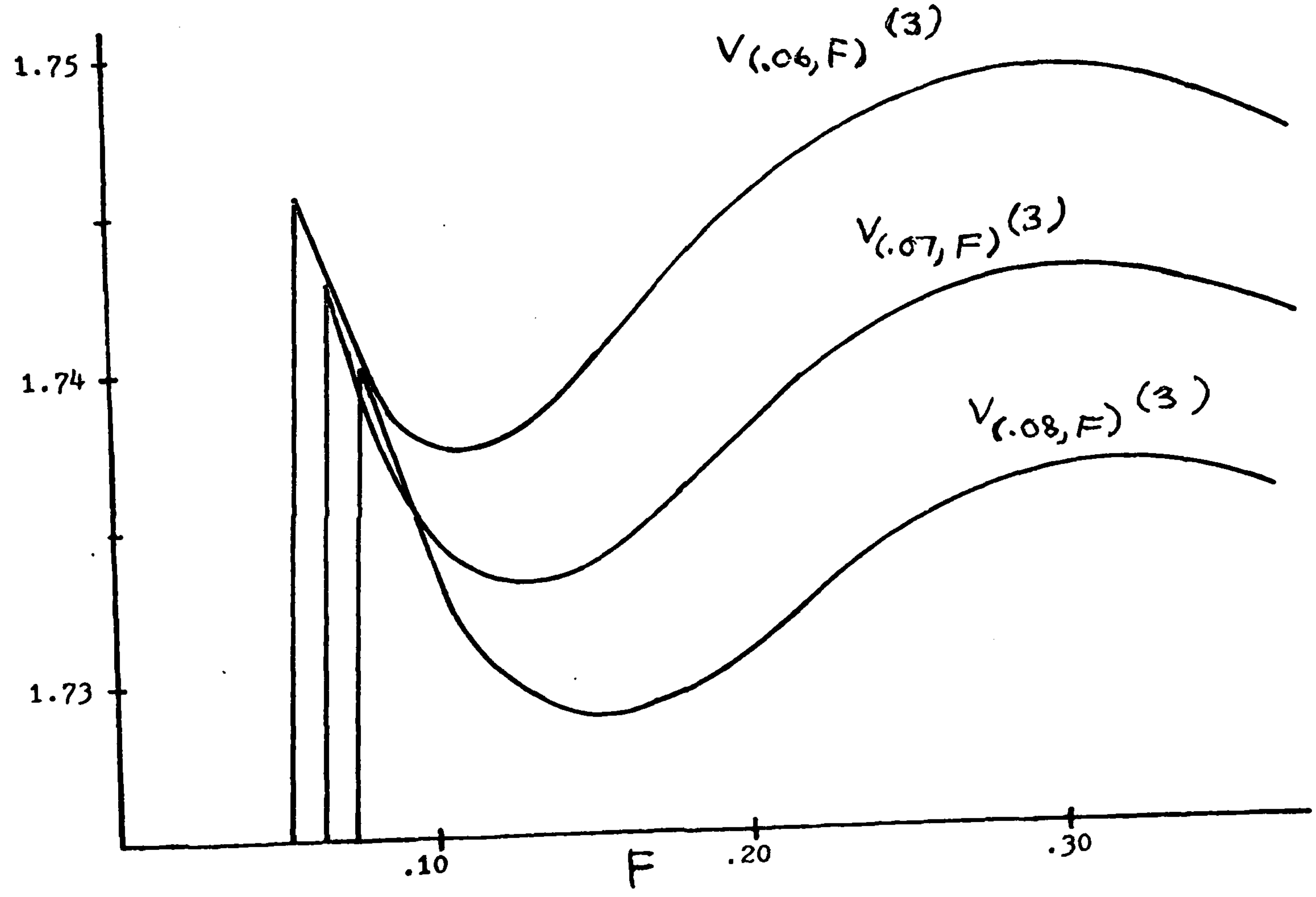


Fig. 2 $F_N(s,\infty)$ vs. s



Figure

$v_{(S,F)}^{(3)}$ as a function of F

References

[1] Roberts, L. G., "Aloha Packet System With and Without Slots and Capture", ASS, Note 8, June 26, 1972

[2] Kleinrock, L. and Tobagi, F., "Carrier Sense Multiple Access for Packet Switched Radio Channels", Proc. Int. Conf. Communications, Mineapolis, MN, June 1974.

[3] Metcalfe, R. M. and Boggs, D. R., "Ethernet: Distributed Packet Switching for Local Computer Networks", CACM, July 1976.

[4] Kleinrock, L. and Yemini, Y., "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication", Proc. Int. Conf. Commun., pp. 7.2.1-7.2.5, 1978.

[5] Kleinrock, L. and Lam, S. S. , "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation", IEEE Trans. Commun., Vol. COM-23, April 1975.

[6] Schoute, F., "Decentralized Control in Packet Switched Satellite Communication", IEEE Transactions on Automatic Control, Vol. AC-23, No. 2, April 1978.

[7] Varaiya, P. and Walrand, J., "Decentralized Control in Packet Switched Satellite Communication", IEEE Transactions on Automatic Control, Vol. AC-24, No. 5, October 1979.

[8] Varaiya, P. and Walrand, J., " A Minimum Principle for Dynamic Teams", presented at the Third Kingston Conference on Differential Games and Control Theory, Univ. of Rhode Island, June 1978.

[9] Tsybakov, B. S. and Mikhailov, B. A., "Random Multiple Packet Access: Part-and-try Algorithm", Problemy Peredachi Informatsii, Vol. 16, No. 4., Oct.-Dec. 1980.

[10] Gallager, R. G., "Conflict Resolution in Random Access Broadcast Networks", Proceedings AFOSR Workshop in Communication Theroy and Applications, Sept. 17-20, 1978, Provincetown, Mass., pp. 74-76.

[11] Hayes, J. H., "An Adaptive Technique for Local Distribution", I.E.E.E. Transactions on Communications, Vol. COM-26, August, 1978.

[12] Capetanakis, J., "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations", Ph.D. Thesis, M.I.T., 1977.

DATE
ILMED
-8