Mixing Collision Resolution Algorithms Exploiting Information of Successful Messages

ILAN KESSLER, STUDENT MEMBER, IEEE, AND MOSHE SIDI, SENIOR MEMBER, IEEE

Abstract —**Algorithms for accessing a slotted-time collision-type broadcast channel are considered under a Poisson infinite-user model, where each user observes the channel and determines for each slot whether idle or success or collision has occurred. The highest throughput previously achieved for this model is 0.4878. We introduce a new scheme which does not exceed the limitations of the system model, yet achieves a throughput of 0.4892. Extra information is added to the message that can be used only when the message is received successfully. One form of such extra information is described, and a new algorithm that exploits it is proposed.**

I. Introduction

CONSIDER the following usual model for multiaccess communications: An infinite number of independent users are transmitting messages of equal length over a slotted-time broadcast channel; the number of messages generated by all users collectively in each slot is Poisson distributed with mean λ. If only one message is transmitted in a slot, it is received successfully; if two or more messages are transmitted in the same slot, then a collision is said to occur, the contents of all collided messages is completely destroyed, and the messages must be retransmitted at some later time. Each user observes the channel and determines at the end of each slot whether that slot was idle or if there was a successful transmission or a collision has occurred. For this model, a number of collision resolution algorithms (CRA) have been introduced in the literature [1]–[6], and the highest throughput achieved so far is 0.4878 [6]. In this paper we introduce a new scheme which does not exceed the limitations of the system model, yet it achieves throughput higher than 0.4878.

It is clear that in the foregoing system, when a message is received successfully, the information contained in the message is available to all users. However, in the algorithms introduced so far the information contained in a successfully transmitted message is not considered, and no use is made of the fact that this information is available. The new scheme we propose utilizes this fact by adding extra information to the message and exploiting this information when the message is received successfully.

Manuscript received November 1986; revised June 1987. This paper was presented in part at the 24th Annual Allerton Conference, Urbana, IL, October 1986.

The authors are with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel.

IEEE Log Number 8821817.

It should be clear that this scheme is safely within the limitations of the system model since the information contained in a successfully transmitted message is anyway available to all users. No assumption of the system model is violated—the extra information is treated exactly in the same manner as the rest of the information of the message; that is, it is exploited *only* when the message is received successfully. This is nothing more than a fuller exploitation of the available features of the system. (If one includes in the previous model the assumption that it is not allowed to consider the contents of a message, then our scheme is viewed as a minor extension of that model.)

However, it is not clear at all whether the throughput can be increased beyond 0.4878 in such a way (the users are independent, and each of them can have at most one message in a lifetime). It is also not clear what kind of information should be added to the message to increase the throughput.

In Section II we introduce a type of information that can be added to the message, and in Section III we propose a new algorithm which exploits this information obtained during successful transmissions. As opposed to all previously presented algorithms, the algorithm proposed here is a *mixing* algorithm which is neither a degenerate intersection algorithm (DIA) nor a first-come first-serve algorithm (FCFSA) [7]. In Section IV we show that this scheme does indeed increase the throughput beyond 0.4878.

II. Set Marker in Mixing Algorithms

In general, a CRA selects at the beginning of each slot a subset A of the time axis, and all messages that arrived during this subset are transmitted in that slot. This is referred to as *enabling* the subset A. We define a *mixing algorithm* to be a CRA that at some step enables a subset A that satisfies the following:

1) $A = \bigcup_{i=1}^{N} A_i'$;

2) $A_i' \subset A_i, \qquad i = 1, 2, \cdots, N$;

3) $\bigcap_{i=1}^{N} A_i = \phi$;

4) it is known that the number of messages contained in A_i is greater than zero, for all $1 \le i \le N$, except possibly some single i_0.

This is referred to as mixing the subsets A_i, $i = 1, 2, \cdots, N$.

The information we propose to add to the message refers to mixing algorithms and is as follows: whenever the enabled subset A satisfies 1)–4), each transmitted message contains the index j of the subset A_j $(1 \le j \le N)$ to which the message belongs (this requires $\lceil \log_2 N \rceil$ bits). In particular, in this work we propose a mixing algorithm that mixes only two subsets A_1 and A_2. The associated information to be added to the message is then a single bit that indicates to which of the two subsets, A_1 or A_2, the message belongs. We refer to this bit as a *set marker*.

The following notations are used in the sequel. The number of messages that arrived during some subset A of the time axis is denoted by $[A]$. If $[A] = 1$, and A is a mix of the subsets A_1 and A_2, and the single message that arrived during A belongs to A_1, then it is denoted by $[A_1 \cup A_2] = 1A_1$ (and by $1A_2$ if the message belongs to A_2). A subset A for which $[A] \ge 2$ is called a *collision interval.*

III. An Algorithm Utilizing the Set Marker

A. Operation of the Algorithm — General Description

Consider two time axes. The first, called the arrival axis, shows the Poisson arrival instants. The second, called the transmission axis, is segmented into consecutive intervals called *collision resolution intervals* (CRI), each consisting of an integral number of slots. At the first slot of the ith CRI the algorithm enables a subset E_i of the arrival axis. If no collision occurs in this slot, the ith CRI terminates and a new CRI starts immediately thereafter. If a collision occurs, the algorithm initiates a *collision resolution process* (CRP) whose termination defines the termination of the ith CRI. Next, the algorithm enables a new subset E_{i+1}, and so on. In this way the arrival axis is entirely searched.

At the end of each CRI the arrival axis can be segmented into disjoint intervals such that each interval belongs to one of the following categories. 1) an interval for which it is known that all messages that arrived during it have already been successfully transmitted; 2) an interval for which nothing is known about the number of messages that arrived during it, beyond the knowledge of the Poisson distribution.

Definition: The *new intervals pool* (NIP) at the beginning of some slot of the ith CRI is the continuous and ordered (according to increasing time index) union of all intervals of category 2) at the end of the $(i-1)$-st CRI minus certain intervals as determined by the algorithm from the beginning of the ith CRI until that moment.

A *new interval* is defined to be a continuous interval of the NIP, starting at the initial instant of the NIP. E_i is a new interval whose length is x (where x is a parameter to be optimized for achieving maximum throughput). In Fig. 1 a typical picture of the arrival axis at the end of a CRI is shown, and the selection of E_i is demonstrated.

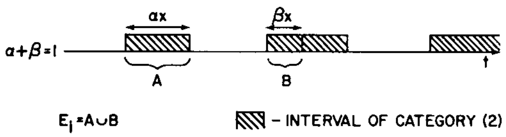

Fig. 1. Arrival axis at end of CRI (typically), and selection of E_i.

B. The Collision Resolution Process

Before describing the CRP formally, it will be helpful to give an informal description. Referring to Fig. 2, E is the collision interval to be resolved. In the first step, the process enables the subinterval E' together with F which is a new interval (mixing).

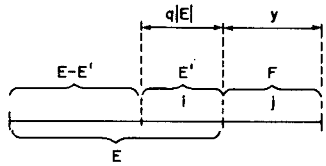

Fig. 2. Mixing collision interval with new interval.

• If the outcome is $[E' \cup F] = 0$, then there is no need to treat E' and F any longer (all messages contained in them have been successfully transmitted). It is also clear now that $[E - E'] \ge 2$, and the process is repeated for $(E - E')$.

• If the outcome is $[E' \cup F] = 1F$, that is, the transmission was successful and the set marker indicated that the message belongs to F (i.e., $[E'] = 0$ and $[F] = 1$), then the situation is similar to that of the previous case and the same action is taken.

• If the outcome is $[E' \cup F] = 1E'$, then the situation is again similar, but now $[E - E'] \ge 1$ and the next step is to enable this interval. If $[E - E'] = 1$, then the process terminates, and if $[E - E'] \ge 2$, then the process is repeated for this interval.

• If the outcome is $[E' \cup F] \ge 2$, then in the next step the process enables E'. If $[E'] = 0$, then there is no need to treat this interval any longer; it is also clear now that $[E - E'] \ge 2$ and $F \ge 2$, and the process is repeated for each of these intervals. If $[E'] = 1$, then there is no need to treat this interval any longer; it is also clear now that $[E - E'] \ge 1$ and $F \ge 1$, and each of these intervals is enabled. If it contains only one message, then it is not treated any longer; otherwise, the process is repeated for it. If $[E'] \ge 2$, then it is easy to see that now nothing is known about the number of messages contained in each of the intervals $(E - E')$ and F, beyond the knowledge of the Poisson distribution, and both are merged into the NIP. The process is repeated for E'.

We now give a formal description of the CRP. The process accepts in its input a subset E of the arrival axis which is known to be a collision interval. During the process, additional intervals, each of which is known to be a collision interval, are generated. The process appends, and later on deletes, some of these intervals to a list named *collision intervals list* (CIL) (in a first-in first-out manner), as specified in the CRP description. At the beginning and at the end of the process the CIL is empty. The process also deletes and/or merges certain intervals to the NIP. While *merging* intervals is specified explicitly in the CRP description, *deleting* is done according to the following rule.

Rule: A new interval which is contained in an enabled subset in some slot is deleted from the NIP at the beginning of that slot.

The process is defined by the following sequence of steps.

The Collision Resolution Process:

1) Enable a right subinterval E' of E whose length is $|E'| = q|E|$ $(0 < q < 1)$ together with a new interval F whose length is y,
   If $[E' \cup F] = \{0 \text{ or } 1F\}$: Denote $(E - E')$ by E and GOTO (1)
   If $[E' \cup F] = 1E'$: Denote $(E - E')$ by E'' and GOTO (5)
   If $[E' \cup F] \geq 2$: GOTO (2)
2) Enable E',
   If $[E'] = 0$: Append F to the CIL; denote $(E - E')$ by E and GOTO (1)
   If $[E'] = 1$: GOTO (3)
   If $[E'] \geq 2$: Merge $(E - E')$ and F into the NIP; denote E' by E and GOTO (1)
3) Enable $(E - E')$,
   If $[E - E'] = 1$: Denote F by E'' and GOTO (5)
   If $[E - E'] \geq 2$: Denote $(E - E')$ by E and GOTO (4)
4) Enable F,
   If $[F] = 1$: GOTO (1)
   If $[F] \geq 2$; Append F to the CIL and GOTO (1)
5) Enable E'',
   If $[E''] = 1$: GOTO (6)
   If $[E''] \geq 2$: Denote E'' by E and GOTO (1)
6) Check the CIL:
   If it is empty: Terminate
   Otherwise: Take the next interval from the CIL, denote it by E and GOTO (1).

IV. Performance Analysis

Let L_N denote the mean length of a CRI that exactly N messages were transmitted in its first slot, and let M_N denote the mean number of successfully transmitted messages during that CRI. From the operation of the algorithm it is clear that

$$L_0 = L_1 = 1 \qquad M_0 = 0 \qquad M_1 = 1. \tag{4.1}$$

For $N \geq 2$, E contains exactly N messages at the beginning of the CRP. Let i and j be the number of messages contained in E' and F, respectively (see Fig. 2). Then from the operation of the algorithm it follows that for $N \geq 2$

$$\begin{aligned} L_N = 1 &+ P(i=0)P(j=0)L_N + P(i=0)P(j=1)L_N \\ &+ P(i=1)P(j=0)(1+L_{N-1}) \\ &+ P(i=0)\sum_{k=2}^{\infty} P(j=k)(L_N + L_k) \\ &+ P(i=1)\sum_{k=1}^{\infty} P(j=k)(2 + L_{N-1} + L_k) \\ &+ \sum_{r=2}^{N} P(i=r)(1+L_r) \end{aligned} \tag{4.2}$$

$$\begin{aligned} M_N = &P(i=0)P(j=0)M_N + P(i=0)P(j=1)(1+M_N) \\ &+ P(i=1)P(j=0)(1+M_{N-1}) \\ &+ P(i=0)\sum_{k=2}^{\infty} P(j=k)(M_N + M_k) \\ &+ P(i=1)\sum_{k=1}^{\infty} P(j=k)(1 + M_{N-1} + M_k) \\ &+ \sum_{r=2}^{N} P(i=r)M_r \end{aligned} \tag{4.3}$$

where

$$P(i=r) = \binom{N}{r} q^r (1-q)^{N-r}, \qquad r = 0,1,\cdots,N \tag{4.4a}$$

$$P(j=k) = \frac{e^{-\lambda y}(\lambda y)^k}{k!} \triangleq \frac{e^{-z}z^k}{k!}, \qquad k \geq 0. \tag{4.4b}$$

Equation (4.2) is explained as follows. The 1 is the first slot of the CRI in which the initial collision is detected. When $i = j = 0$ or $i = 0$ and $j = 1$, the CRI lasts $1 + (L_N - 1)$ slots more on average—one slot to detect i and j plus $L_N - 1$ slots on average to resolve the collision among the N messages contained in $(E - E')$ (one slot less since the collision is known in advance). When $i = 1$ and $j = 0$, the CRI lasts $1 + L_{N-1}$ slots more on average—one slot to detect i and j plus L_{N-1} slots on average to resolve the $N-1$ messages contained in $(E - E')$ $(N - 1 \geq 1$, and therefore it is also required to enable $(E - E')$ itself). When $i = 0$ and $j = k \geq 2$, the CRI lasts $2 + (L_N - 1) + (L_k - 1)$ slots more on average—two slots to detect that $i + j \geq 2$ and $i = 0$, plus $L_N - 1$ and $L_k - 1$ slots on average to resolve the collisions (which are known in advance) among the N messages contained in $(E - E')$ and the k messages contained in F, respectively. The case when $i = 1$ and $j = k \geq 1$ is explained similarly, except that now it is required to enable the intervals themselves before resolving them as in the case when $i = 1$ and $j = 0$. Finally, when $i = r \geq 2$, the CRI lasts $2 + (L_r - 1)$ slots more on average—two slots to detect that $i + j \geq 2$ and $i \geq 2$ plus $L_r - 1$ slots on average to resolve the collision (which is known in advance) in E'. Equation (4.3) is explained similarly.

By rearranging the terms in (4.2) and (4.3), we obtain that each of the equations is of the following form:

$$x_N = \sum_{r=2}^{N} P(i=r)x_r + P(i=1)x_{N-1} + P(i=0)x_N + [P(i=0)+P(i=1)] \cdot \sum_{k=2}^{\infty} P(j=k)x_k + b_N, \quad N \geq 2 \tag{4.5}$$

where in (4.2),

$$b_N = 2 - P(i=0) + P(i=1) \cdot [1 - P(j=0) + P(j=1)] \triangleq b_N^{(L)} \tag{4.6a}$$

and in (4.3),

$$b_N = P(i=1) + P(j=1)[P(i=0)+P(i=1)] \triangleq b_N^{(M)}. \tag{4.6b}$$

Equation (4.5) defines an infinite system of equations in x_N that can be rewritten as

$$x_N = \sum_{i=2}^{\infty} a_{N,i}x_i + b_N \triangleq F_N(X), \qquad N \geq 2 \tag{4.7}$$

where $X = \{x_N\}_{N=2}^{\infty}$ and $a_{N,i}$ and b_N are defined by (4.5) and (4.6).

Clearly, each of the sequences $\{L_N\}_{N=2}^{\infty}$ and $\{M_N\}_{N=2}^{\infty}$ is a solution of the respective system of equations (4.7). Employing a similar approach to that in [8], we state the following theorem.

Theorem: If

$$2q(1-q) > z(1-e^{-z}), \tag{4.8}$$

then the following hold.

a) The system of equations (4.7) has a solution $X' = \{x'_N\}_{N=2}^{\infty}$ which satisfies $0 \leq x'_N \leq \alpha N$, $N \geq 2$ where $\alpha = 4/[2q(1-q) - z(1-e^{-z})]$.

b) X' is unique in the class of sequences X that satisfy

$$\lim_{k \to \infty} \max_{N<k} \sum_{i=k}^{\infty} a_{N,i}|x_i| = 0. \tag{4.9}$$

c) If for all $N \geq 2$, $b_N = b_N^{(L)}$ ($b_N = b_N^{(M)}$), then for all $N \geq 2$, $L_N = x'_N$ ($M_N = x'_N$).

The proof of this theorem is given in the Appendix.

It follows from the theorem that for all q and z satisfying (4.8), L_i and M_i do not increase with i more rapidly than linearly. On the other hand, for z not large, $a_{N,i}$ decrease very rapidly with increasing i for $i > N$ (like $i! \simeq i^i$). Hence an approximate solution for L_N and M_N for $2 \leq N \leq K$, where K is some integer, can be obtained by solving the finite system of equations obtained from the given infinite one by discarding all equations and unknowns commencing with $K+1$ (K is determined according to the desired accuracy).

L_N and M_N discussed earlier correspond to a CRI in which exactly N messages were transmitted in its first slot; that is, the new interval E_i (whose length is x) that corresponds to the CRI contains exactly N messages. We denote the corresponding averages over N (which is Poisson distributed with parameter $W \triangleq \lambda x$) by

$$L = \sum_{N=0}^{\infty} \frac{e^W W^N}{N!} L_N \qquad M = \sum_{N=0}^{\infty} \frac{e^W W^N}{N!} M_N. \tag{4.10}$$

Let $M(s)$ denote the random variable which is the number of successfully transmitted messages during the first s CRI's, and let $L(s)$ denote the random variable which is the number of slots required to transmit them. The throughput of the algorithm is defined as $T = \lim_{s \to \infty}\{E[M(s)]/E[L(s)]\}$. By using the fact that Poisson arrivals in disjoint intervals are independent, it is easy to show that $T = M/L$. Numerical optimization over q, z, W for achieving maximum throughput yields the following results: throughput $= 0.4892$ for $W = 1.24$, $q = 0.26$, $z = 0.66$. Note that the throughput achieved is higher than 0.4878. Note also that q and z do satisfy (4.8).

V. Discussion and Summary

We introduced a new scheme in which extra information added to the message is exploited when the message is received successfully. We proposed a type of information to be added to the message and a new algorithm which exploits this information obtained during successful transmissions. This algorithm mixes a collision interval with a new interval and achieves a throughput of 0.4892. The highest throughput previously achieved for the model discussed was 0.4878. An algorithm in which a new interval is mixed with an interval that is known to contain at least one message, and an algorithm in which two collision intervals are mixed have also been considered, but in both cases the throughput achieved was less than 0.4878. In [10] it is shown that by enabling a small interval that is known to contain at least one message together with a new interval, the throughput of the 0.487 algorithm [4] can be increased by 3.6×10^{-7}.

Note that our scheme can be viewed as a channel coding scheme in the sense that extra information added to the message is exploited to improve the performance of the system.

In the algorithm presented the bookkeeping concerning collision intervals and new intervals requires an unbounded amount of memory. To overcome this difficulty, we first observe that the bookkeeping concerning new intervals differs substantially from that for collision intervals. The memory needed for new intervals can be reduced to that needed for keeping a single point of the arrival axis simply by appropriate shifting of clocks (without any degradation in the throughput). To bound the amount of memory required for collision intervals, one must limit the size of the CIL and resolve collision intervals that otherwise would have been appended to the CIL by the 0.487 algorithm [4] whenever the CIL becomes full. This leads to a degradation in the throughput; in the worst case (i.e., CIL of size zero) it reduces to 0.4881, which is still higher than 0.4878.

An algorithm that exploits the same type of extra information obtained during successful transmissions, when the number of colliding messages in each collision is known, is introduced in [11]. It is shown there that the throughput achieved by this algorithm is 0.5533, which exceeds the highest throughput previously achieved for this case (0.5324).

Throughout the computation of the throughput the set marker has not been separated from the message. Therefore, the result obtained is asymptotic with the message length. However, since the message length in the model is arbitrary, this presents no difficulty. Moreover, since the set marker is only a single bit, this does not cause any practical difficulty either.

The improvement achieved in the throughput is small, but note that the set marker and the algorithm proposed are only an example of a scheme in which extra information added to the message is exploited when the message is received successfully; they are not necessarily the optimal choice. The importance of our result resides mainly in the fact that it proves that the throughput can be increased by such a scheme.

Acknowledgment

The authors would like to thank Dr. Israel Cidon for helpful discussions.

Appendix

Here we give the proof of the theorem stated in Section IV, employing a similar approach to that in [8].

Proof of a)

First we show that the existence of $X^{(0)} = \{x_N^{(0)}\}_{N=2}^{\infty}$ with $x_N^{(0)} \geq 0$, $N \geq 2$ that satisfies (4.9) and satisfies $F_N(X^{(0)}) \leq x_N^{(0)}$, $N \geq 2$, implies that (4.7) has a solution $X' = \{x'_N\}_{N=2}^{\infty}$ satisfying $0 \leq x'_N \leq x_N^{(0)}$, $N \geq 2$. Indeed, consider the sequence $X^{(j)}$, $j = 0,1,2,\cdots$ defined via $x_N^{(j+1)} = F_N(X^{(j)})$, $N \geq 2$. It follows by induction that for every $N \geq 2$ the sequence $x_N^{(j)}$, $j = 0,1,2,\cdots$ is nonincreasing, and since $a_{N,i}$, $b_N \geq 0$, we have that for every $N \geq 2$, $x_N^{(j)} \geq 0$ for all $j \geq 0$. Hence the following exists:

$$\lim_{j \to \infty} x_N^{(j)} = x'_N \qquad 0 \leq x'_N \leq x_N^{(0)}, \qquad N \geq 2.$$

We will show that $X' = \{x'_N\}_{N=2}^{\infty}$ is a solution of (4.7). Consider the following equality:

$$x_N^{(j+1)} = \sum_{i=2}^{\infty} a_{N,i} x_i^{(j)} + b_N, \qquad N \geq 2.$$

In passing to the limit as $j \to \infty$, in the right side a term-by-term passage to the limit is admissible, for the series at the right converges uniformly as regards j since it is majored by the series with constant terms $\sum_{i=2}^{\infty} a_{N,i} x_i^{(0)}$ ([9], p. 22). Thus on effecting this passage we find that

$$x'_N = \sum_{i=2}^{\infty} a_{N,i} x'_i + b_N, \qquad N \geq 2;$$

i.e., X' is indeed a solution of (4.7).

Next we show that if (4.8) holds, then $X^{(0)} = \{x_N^{(0)}\}_{N=2}^{\infty}$ with $x_N^{(0)} = \alpha N$, $N \geq 2$ where $\alpha = 4/[2q(1-q) - z(1-e^{-z})]$, satisfies the previous conditions on $X^{(0)}$. Clearly, this completes the proof of a).

To show this, let $x_N^{(0)} = cN$, $N \geq 2$ with arbitrary c. Substituting this into (4.5), we obtain after some calculations that

$$F_N(X^{(0)}) = x_N^{(0)} - [c(B_N - A_N) - b_N], \qquad N \geq 2 \quad \text{(A.1)}$$

where

$$B_N \triangleq N[1 - q - P(i=0) - P(i=1)] + 2P(i=1)$$
$$A_N \triangleq [P(i=0) + P(i=1)] z(1 - e^{-z}).$$

It is clear now that if $B_N - A_N > 0$ and $c \geq b_N/(B_N - A_N)$ for all $N \geq 2$, then all the required conditions are met, i.e., $x_N^{(0)} \geq 0$, $N \geq 2$, $X^{(0)}$ satisfies (4.9), and $F_N(X^{(0)}) \leq x_N^{(0)}$, $N \geq 2$. Clearly, this is ensured if $B_l > 0$ and $c \geq B_u$, where B_l and B_u are a lower bound on $B_N - A_N$ and an upper bound on $b_N/(B_N - A_N)$, respectively, over all $N \geq 2$.

B_l and B_u are obtained as follows:

$$\begin{aligned} B_N - A_N &= N[1 - q - P(i=0) - P(i=1)] + 2P(i=1) \\ &\quad - [P(i=0) + P(i=1)] z(1 - e^{-z}) \\ &\geq N[1 - q - P(i=0) - P(i=1)] + 2P(i=1) \\ &\quad - z(1 - e^{-z}) \triangleq h(N) - z(1 - e^{-z}), \qquad N \geq 2 \quad \text{(A.2)} \end{aligned}$$

$$\begin{aligned} h(N) &= N\left[1 - q - (1-q)^N - Nq(1-q)^{N-1}\right] \\ &\quad + 2Nq(1-q)^{N-1} \\ &= N(1-q)\left\{1 - (1-q)^{N-2}[(N-3)q + 1]\right\} \\ &\triangleq N(1-q) g(N), \qquad N \geq 2. \end{aligned}$$

$g(N)$ satisfies

$$g(N) - g(N-1) = (1-q)^{N-3} q^2 (N-3), \qquad N \geq 3$$

and $g(2) = q$, and therefore, for all $0 < q < 1$, $g(N)$ is a positive nondecreasing function for $N \geq 2$. Hence $h(N)$ is an increasing function for $N \geq 2$ for all $0 < q < 1$, and therefore, $h(N) \geq h(2) = 2q(1-q)$ for all $N \geq 2$. Substituting this into (A.2), we obtain that for all $N \geq 2$,

$$B_N - A_N \geq 2q(1-q) - z(1 - e^{-z}) \triangleq B_l.$$

To obtain an upper bound on $b_N/(B_N - A_N)$, we note that $b_N \leq 4$ for all $N \geq 2$, and therefore, for $B_l > 0$, i.e., $2q(1-q) > z(1 - e^{-z})$,

$$\frac{b_N}{B_N - A_N} \leq \frac{4}{2q(1-q) - z(1-e^{-z})} = \alpha \triangleq B_u.$$

Hence if (4.8) holds and $c \geq \alpha$, in particular, $c = \alpha$, then $X^{(0)} = \{x_N^{(0)}\}_{N=2}^{\infty}$ with $x_N^{(0)} = cN$, $N \geq 2$, satisfies the required conditions. Q.E.D.

Proof of b)

Assume that (4.7) has two solutions $\{x_N^{(1)}\}_{N=2}^{\infty}$ and $\{x_N^{(2)}\}_{N=2}^{\infty}$ satisfying (4.9). Then their difference $d_N \triangleq x_N^{(1)} - x_N^{(2)}$, $N \geq 2$, satisfies $d_N = \sum_{i=2}^{\infty} a_{N,i} d_i$, $N \geq 2$. Since $\{x_N^{(j)}\}_{N=2}^{\infty}$, $j = 1,2$ satisfy (4.9), it follows that for all $\epsilon > 0$ and $k_1 \geq 2$, there exists $k \geq k_1$ large enough such that for all $N < k$,

$$\sum_{i=k}^{\infty} a_{N,i} |x_i^{(j)}| \leq \frac{1}{2}\epsilon, \qquad j = 1,2,$$

and therefore,

$$\sum_{i=k}^{\infty} a_{N,i} |d_i| \leq \sum_{i=k}^{\infty} a_{N,i} \left(|x_i^{(1)}| + |x_i^{(2)}|\right) \leq \epsilon.$$

Hence

$$|d_N| \le \sum_{i=2}^{\infty} a_{N,i}|d_i| \le \sum_{i=2}^{k-1} a_{N,i}|d_i| + \epsilon.$$

We now construct an upper bound $x_N^{(0)}$ on $|d_N|$ $2 \le N < k$ by letting

$$x_N^{(0)} = \alpha_\epsilon N, \qquad N \ge 2, \ \alpha_\epsilon \ge 0$$

where α_ϵ is chosen such that $x_N^{(0)} \ge |d_N|$ for $2 \le N < k$, and such that equality holds for some $2 \le N_0 < k$. Then for all $2 \le N < k$,

$$|d_N| \le \sum_{i=2}^{k-1} a_{N,i}|d_i| + \epsilon \le \sum_{i=2}^{k-1} a_{N,i}|x_i^{(0)}| + \epsilon \le \sum_{i=2}^{\infty} a_{N,i}|x_i^{(0)}| + \epsilon$$

$$= \sum_{i=2}^{\infty} a_{N,i}x_i^{(0)} + \epsilon = x_N^{(0)} - [\alpha_\epsilon(B_N - A_N) - \epsilon]$$

where the last equality is obtained similarly to (A.1), and A_N and B_N are defined there. It is clear that for $2q(1-q) > z(1-e^{-z})$, if $\alpha_\epsilon > \epsilon/[2q(1-q) - z(1-e^{-z})]$, then $|d_N| < x_N^{(0)}$ for all $2 \le N < k$, and this contradicts our choice of α_ϵ (such that equality holds for some $2 \le N_0 < k$). Hence $\alpha_\epsilon \le \epsilon/[2q(1-q) - z(1-e^{-z})]$, and since ϵ and k_1 are arbitrary, it follows that $|d_N| \equiv 0$ for all $N \ge 2$. Q.E.D.

Proof of c)

We prove here only for L_N. The proof for M_N is entirely similar.

Let l_N denote the random variable which is the length of a CRI that exactly N messages were transmitted in its first slot. Clearly, $L_N = E[l_N]$.

Let $k \ge 1$ be some integer. For all $N \ge 2$ define

$$l_N^k \triangleq \min(l_N, k)$$
$$L_N^k \triangleq E[l_N^k].$$

The following holds for every $N \ge 2$ and $k \ge 1$:

$$L_N^k \le L_N^{k+1} \le L_N \qquad \lim_{k \to \infty} L_N^k = L_N \tag{A.3}$$

$$L_N^k \le k \le \beta \le \beta N \tag{A.4}$$

where $\beta \triangleq \max(k, \alpha)$, and α is defined in part a) of the theorem. From the operation of the algorithm, it follows that for every $k \ge 1$,

$$L_N^k \le F_N(\mathbf{L}_k), \qquad N \ge 2 \tag{A.5}$$

where

$$\mathbf{L}_k \triangleq \{L_N^k\}_{N=2}^{\infty}.$$

Lemma: For every $k \ge 1$,

$$L_N^k \le x'_N, \qquad N \ge 2.$$

Proof: For every $k \ge 1$, consider the sequence $\mathbf{L}_k^{(j)} = \{L_N^{k(j)}\}_{N=2}^{\infty}$, $j = 0,1,2,\cdots$ defined via $\mathbf{L}_k^{(0)} = \mathbf{L}_k$; $L_N^{k(j+1)} = F_N(\mathbf{L}_k^{(j)})$, $N \ge 2$. It follows from (A.5) by induction that for every $N \ge 2$ the sequence $L_N^{k(j)}$, $j = 0,1,\cdots$ is nondecreasing. On the other hand, it is clear from the last part of the proof of a) that $F_N(Y) \le \beta N$, where $Y = \{y_N\}_{N=2}^{\infty}$ with $y_N = \beta N$, $N \ge 2$. It follows from this and from (A.4) by induction that for every $N \ge 2$, $L_N^{k(j)} \le \beta N$ for all $j \ge 0$. Hence the following exists:

$$\lim_{j \to \infty} L_N^{k(j)} = x''_N \qquad L_N^k \le x''_N \le \beta N, \qquad N \ge 2.$$

It can be shown in a similar way to that in the proof of a) that $X'' = \{x''_N\}_{N=2}^{\infty}$ is a solution of (4.7). Since $0 \le x''_N \le \beta N$, $N \ge 2$, it follows that X'' satisfies (4.9). It follows now from the uniqueness of the solution that $x''_N = x'_N$ for all $N \ge 2$; hence $L_N^k \le x'_N$ for all $N \ge 2$. This completes the proof of the Lemma.

It follows from the Lemma and from (A.3) that $L_N \le x'_N$ for all $N \ge 2$, and therefore, $\{L_N\}_{N=2}^{\infty}$ satisfies (4.9). On the other hand, $\{L_N\}_{N=2}^{\infty}$ is a solution of (4.7). It follows now from the uniqueness of the solution that $L_N = x'_N$ for all $N \ge 2$. Q.E.D.

References

[1] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.

[2] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Prob. Inform. Trans.*, vol. 14, pp. 259–280, Oct.–Dec. 1978.

[3] J. L. Massey, "Collision resolution algorithms and random-access communications," in *Multi-User Communications* (CISM Courses and Lecture Series), G. Longo, Ed. New York: Springer-Verlag, 1981.

[4] R. G. Gallager, "Conflict resolution in random access broadcast networks," in *Proc. AFOSR Workshop Communication Theory and Applications*, Provincetown, MA, Sept. 1978, pp. 74–76.

[5] B. S. Tsybakov and V. A. Mikhailov, "Random multiple access of packets: Part-and-try algorithm," *Prob. Inform. Trans.*, vol. 16, Oct.–Dec. 1980.

[6] J. Mosley and P. A. Humblet, "A class of efficient contention resolution algorithms for multiple access channels," *IEEE Trans. Commun.*, vol. COM-33, pp. 145–151, Feb. 1985.

[7] S. S. Panwar, D. Towsley, and J. K. Wolf, "On the throughput of degenerate intersection and first-come first-served collision resolution algorithms," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 274–279, Mar. 1985.

[8] B. S. Tsybakov and N. D. Vvedenskaya, "Random multiple-access stack algorithm," *Prob. Inform. Trans.*, vol. 16, pp. 230–243, July–Sept. 1980.

[9] L. V. Kantarovich and V. I. Krylov, *Approximate Methods of Higher Analysis.* Groningen, The Netherlands: P. Noordhoff, 1958.

[10] N. D. Vvedenskaya and M. S. Pinsker, "Non-optimality of the part-and-try algorithm," in *Abst. Int. Workshop on Convolutional Codes and Multiuser Communication*, Sochi, USSR, 1983, pp. 141–148.

[11] I. Kessler and M. Sidi, "Mixing collision resolution algorithms exploiting information of successful messages," in *Proc. 24th Ann. Allerton Conf. Communication, Control, and Computing*, Urbana, IL, Oct. 1986, pp. 844–853.