The Collision Channel Without Feedback

JAMES L. MASSEY, FELLOW, IEEE, AND PETER MATHYS, MEMBER, IEEE

Abstract—A model is proposed for the situation where M users share a common communication resource but, because of unknown time offsets among their clocks, cannot transmit their data packets in a time-sharing mode and, because of the lack of a feedback link, can never determine these time offsets and also can never be sure of the outcomes of their individual packet transmissions. Each user is required to make his packet transmissions at times determined by a protocol signal that is independent of the data to be sent.

The capacity and zero-error capacity regions of this channel are determined for both the unsynchronized and slot-synchronized cases; these four regions are shown to coincide. It is further shown that a dense set of rate points on the outer boundary of this region can be achieved in the slot-synchronized case. Specific constructions of protocol sequences for achieving these points are given, and the technique of "decimation decoding" is introduced for identifying the sender of each successfully transmitted packet. Maximum-erasure burst-correcting codes over an alphabet of arbitrary size are constructed and shown to suffice for reconstructing the packets lost in "collisions" when these protocol sequences are used.

I. Introduction

THE USUAL PURPOSE of "random accessing" is to reduce the large message delay that would otherwise result if many senders, who only infrequently had messages, shared a common communications resource on a time-division multiple-access (TDMA) basis. Sometimes, however, random accessing is necessitated where TDMA might be preferred but is impractical because of the difficulty in synchronizing transmission from the senders. Satellite relay systems and mobile radio systems are instances where such synchronization of data packets may be well-nigh impossible.

Random accessing leads inevitably to "collisions" when two or more senders simultaneously transmit. It is often thought that "feedback" is required in such systems so that senders can retransmit packets after being notified via feedback of their loss in collisions.

The purpose of this paper is to explore how much loss of transmission capacity occurs when M senders are *forced* to use random accessing because they cannot synchronize their transmissions. This viewpoint requires us to rule out the presence of a feedback link, as such feedback could otherwise be exploited by the users to bring their transmissions eventually into any desired synchronism. We shall demonstrate that reliable random-access communications is indeed possible without a feedback link.

In Section II, we describe the channel model that will be used through this paper. Section III introduces four different capacity regions and states the main result of this paper, viz. that these four regions coincide. Section IV gives the required proof that reliable communication outside the capacity region is impossible. Section V gives a constructive scheme for signaling without error at rates on the outer boundary of the capacity region when the senders are slot-synchronized. Section VI gives a similar constructive scheme for signaling without error at all rates in the interior of the capacity region in the fully unsynchronized case. Finally, in Section VII, we place the results of this paper in historical perspective, and we make some remarks about the significance and proper interpretation of these results.

II. The Channel Model

Channel models generally have two distinct features: 1) specification of the conditional probability law (or deterministic rule) for the channel output(s) given the channel input(s); and 2) specification of constraints on channel usage. The first of these two specification might well be called the "basic channel model." For instance, the basic channel model might be a discrete-time memoryless additive Gaussian noise channel. The constraint on channel usage then might be a specified upper bound on the second moment of the input variable or a specified upper bound on the magnitude of the input variable. Note that the channel model is not complete (and in particular the capacity is not computable) until the constraints on the channel usage are specified.

A. The Basic Channel Model

The basic channel model for the *collision channel without feedback* (CCw/oFB) is illustrated in Fig. 1. Our intent is to model the situation in which there are M channel users,

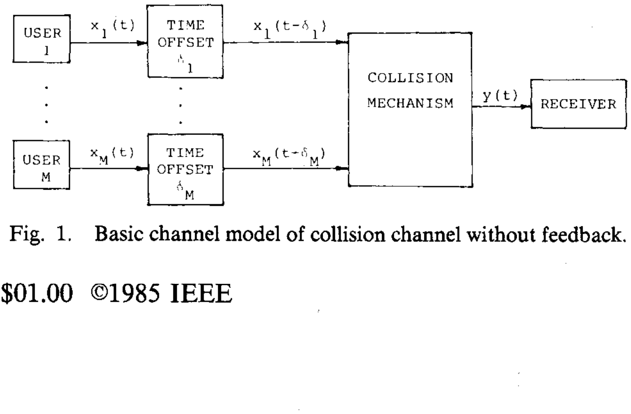

Fig. 1. Basic channel model of collision channel without feedback.

Manuscript received February 11, 1985. This work was supported in part by the Swiss National Fund. This paper was presented in part at the Information Theory Conference, Oberwolfach, Germany, April 5–10, 1982, and in part at the IEEE International Symposium on Information Theory, Les Arcs, France, June 21–25, 1982.

J. L. Massey is with the Institute for Signal and Information Processing, Swiss Federal Institute of Technology CH-8092 Zurich, Switzerland.

P. Mathys was with the Swiss Federal Institute of Technology, Zurich, Switzerland. He is now with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

0018-9448/85/0300-0192$01.00 ©1985 IEEE

each of which occasionally sends a "packet" of some fixed duration, say T seconds, but otherwise is silent. Thus, the input signal $x_i(t)$ from user i in Fig. 1 will be zero except in those intervals of length T seconds where user i is actually sending a packet, in which intervals we assume only that $x_i(t)$ is some suitably recognizable nonzero waveform. We assume that a packet has Q possible values for some fixed integer Q, $Q \geq 2$, and we define $\log_2 Q$ bits of information to be a *packet* of information.

Our intent further is to model the situation where there is no common time reference between any of the users or the receiver. To achieve this, we introduce the *time offsets* $\delta_1, \delta_2, \cdots, \delta_M$ as shown in Fig. 1. In Fig. 1, $x_i(t)$ denotes user-i's transmitted signal at his own local time t, while $y(t)$ denotes the received signal at the receiver's local time t.

The time offset δ_i should be interpreted as the difference between the time shown on the receiver's clock and the time shown on user i's clock so that a signal from user i, received at time t on the receiver's clock, was actually sent at time $t - \delta_i$ on user i's clock. (Note that this situation is entirely equivalent to assuming that the clocks at user i and at the receiver are perfectly synchronized, but that the signal from user i is delayed by δ_i before reaching the receiver. It does not hurt to think of δ_i as the propagation delay for user i's signal—provided one is willing to allow negative delays.) The key point in our model is that *all time offsets are unknown to all users*, and can never be learned as the users receive no feedback from the channel, and are also unknown in advance to the receiver.

Our intent next is to model the situation in which packets that overlap at the receiver, partially or completely, are completely destroyed by such "collision," but are received error-free in the absence of a collision. A packet sent by user i starting at time t_a will be assumed to collide with a packet sent by user j starting at time t_b if and only if

$$\left|(t_a - \delta_i) - (t_b - \delta_j)\right| < T, \tag{1}$$

i.e., if and only if the time difference between receipt of their leading edges is less than the packet duration. We assume that the received signal $y(t)$ at the output of the "collision mechanism" in Fig. 1: 1) coincides with the corresponding recognizable packet waveform during receipt of a noncollided packet; 2) is recognizable as a "garble" (and nothing more) during receipt of any collided packets; 3) is recognizable as "silence" during periods when no packet (collided or not) is received. (To be fully precise, we need also to assume that the receiver is able to recognize the boundary between packets received successfully and precisely adjacent to one another.)

We complete our basic channel model by distinguishing two cases for the possible values of the unknown time offsets, namely

1) the *slot-synchronized* case in which the time offsets $\delta_1, \delta_2, \cdots, \delta_M$ are arbitrary integer multiples of T,
2) the *unsynchronized* case in which the time offsets $\delta_1, \delta_2, \cdots, \delta_M$ are arbitrary real numbers.

We define *time slot* n to be the semi-open interval $nN \leq t < (n+1)T$, where local time is understood. In the slot-synchronized case, if user i sends a packet precisely within his own time slot n, then it will be received precisely within the receiver's time slot $n + \delta_i/T$. Thus, if all users align their packet transmissions within time slots, collisions will result only when received packets completely overlap. In the unsynchronized case, however, the users have no way to avoid collisions that result from only partial overlapping of packets.

B. The Constraints on Channel Usage

The constraints on channel usage for the CCw/oFB are illustrated by Fig. 2 which shows the detailed structure by which user i is permitted to use the basic channel of Fig. 1. Each user has an independent information source which, upon demand, produces a Q-ary symbol to be transmitted to the destination.

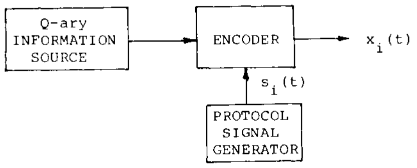

Fig. 2. Constraint on channel usage for collision channel without feedback.

In actual random-access systems, "information" is transmitted only via the contents of packets and not also via the timing of access attempts. To say this in another way, the randomness of the "information" is not used in the selection of transmission times. Such a prohibition has the desirable effect that system performance does not vary with the statistical nature of the information transferred. We wish to impose such a prohibition against the dependence of starting times on information to be transmitted in our channel model. We do this by requiring that each user have a *protocol signal* generator as shown in Fig. 2 whose output is a predetermined periodic waveform that completely specifies the transmission times for that user. This protocol signal $s_i(t)$ for user i has period τ_i, has value either zero or one for all t, and takes on value one only over semi-open intervals whose lengths are integer multiples of T. The encoder for user i is required to emit packets whenever $s_i(t) = 1$ and is required to be silent (i.e., to emit the zero waveform) whenever $s_i(t) = 0$. We assume that the users may jointly choose their protocol signals and that their choice is known by the receiver.

It may seem strange that we have included an "on demand" information source in our model, as one usually thinks of a random-access system as the appropriate way to transmit many sources each of which only infrequently has something to say. However, it seems desirable when possible to decouple the channel model from the source model so that "capacity" does not depend on the source. One might view our "on demand" source model as a kind of worst-case assumption that all of the sporadic sources are active and hence each has a nonempty queue of messages awaiting transmission. Capacity can then be interpreted as the best possible performance for heavy loading of the system. Effectively, one makes such an "on demand" source assumption when one asserts, for instance, that the capacity of a time-division multiple-access system is one packet per slot. At bottom however, we must admit that our channel model is aimed primarily at determining how much loss results when M senders share a common channel but are prevented from time-sharing this channel by what appears to be the mildest possible assumption that prevents such time-sharing, namely, lack of a common time reference.

It may also seem strange that we have required determinisitic protocol signals (and indeed periodic ones) to control access in our model of a random-access system. Note, however, that nothing prevents user i from choosing the first period of $s_i(t)$ as a realization of some appropriate random process. The point is that the random process that controls transmission time should not depend on the information source; thus we can conveniently consider that any random experiment used to produce the protocol signals has been carried out in advance of performing the random experiment used to produce the output sequences of the information sources. We have required the protocol sequences to have finite periods for analytical convenience, but we have placed no finite bound on these periods so this is no real limitation on the model.

The purpose of the encoder in Fig. 2 is to code the output of the Q-ary source into packets for transmission so that the receiver will be able to reconstruct the output of this source from the received signal $y(t)$ with an acceptably small error probability. The receiver must, of course, so reconstruct each of the M sources.

III. Capacity Regions and Main Results

In proving coding theorems for the CCw/oFB, we shall always assume that the "on demand" source for each user is a *Q-ary symmetric source* (QSS), i.e., a source whose next output digit is equally likely to be any of the Q possible values, independent of its past history. The QSS has an information rate of $\log_2 Q$ bits per symbol or, equivalently, one packet per symbol.

For the CCw/oFB, it is convenient to define the *duty factor* p_i for user i as that fraction of its period during which the protocol signal $s_i(t)$ is nonzero, i.e., the fraction of time during which user i is actually transmitting packets. Of course, $0 \leq p_i \leq 1$. Note that if user i is transmitting information from his QSS at a rate R_i packets/slot, then he is actually transmitting information at a rate R_i/p_i packets/slot during those times that he is *actively using* the channel.

In general, by the "capacity region" of any multiuser channel, one means the set of all joint user rates such that it is possible to communicate with arbitrarily small (positive) error probability at any joint rate inside this set, but it is impossible to do so at any joint rate outside this set. By the "zero-error capacity region," one means the joint rate region where zero-error probability is possible. To define such regions precisely, it is convenient to make use of Shannon's concept of an "approachable" rate [1, p. 614] so that the capacity regions are always closed sets.

We now define the *capacity region* $\mathscr{C}_u$ of the M-user unsynchronized CCw/oFB as the set of all rate vectors $\boldsymbol{R} = (R_1, R_2 \cdots, R_M)$, with $R_i \geq 0$ for $1 \leq i \leq M$, that are *approachable* in the sense that, given any positive numbers δ and ϵ, there exist a protocol signal $s_i(t)$ and a block code of length n_i packets for each user i such that

1) blocks of at least $(R_i/p_i - \delta)n_i$ packets from the QSS for user i are encoded into blocks of n_i packets for transmission during successive slots in which user i actually uses the channel; and
2) a decoder can, from the channel output signal, reconstruct the output sequence of user i's QSS with average packet error probability at most ϵ, *regardless of the values of the time offsets $\delta_1, \delta_2, \cdots, \delta_M$.*

The zero-error capacity region, $\mathscr{C}_{u0}$, of the unsynchronized CCw/oFB is defined in the same way as $\mathscr{C}_u$ except that $\epsilon = 0$ is specified. The capacity region and zero-error capacity region of the slot-synchronized CCw/oFB, $\mathscr{C}_s$ and $\mathscr{C}_{s0}$, respectively, are similarly defined.

It proves convenient here to introduce the concept of the "outer boundary" of a capacity (or zero-error capacity) region. We shall write an inequality between vectors, e.g., $\boldsymbol{R}' \leq \boldsymbol{R}$, to denote the corresponding inequality between each of their components. By definition, any point $\boldsymbol{R}$ in any of the capacity (or zero-error capacity) regions defined above satisfies $\boldsymbol{R} \geq \boldsymbol{0}$, where $\boldsymbol{0}$ denotes the all-zero vector with M components. We also see immediately that if $\boldsymbol{R}$ is in a capacity (or zero-error capacity) region and $\boldsymbol{0} \leq \boldsymbol{R}' \leq \boldsymbol{R}$, then $\boldsymbol{R}'$ is also in this capacity (or zero-error capacity) region. Thus, we can define the *outer boundary* of a capacity (or zero-error capacity) region as the set of all points $\boldsymbol{R}$ of this region such that there is no other point $\boldsymbol{R}'$ in this region for which $\boldsymbol{R} \leq \boldsymbol{R}'$. Note that *specification of a capacity (or zero-error capacity) region is equivalent to specification of its outer boundary.* As a trivial illustration of these concepts, we remark that the capacity region of a single-sender single-receiver discrete memoryless channel (DMC) with capacity C is the closed interval $[0, C]$ and its outer boundary is the singleton set $\{C\}$. Because the outer boundary of a capacity (or zero-error-capacity) region for a multi-user channel is the natural generalization of the capacity (or zero-error capacity) of a DMC, we shall denote points on the outer boundary of such a region by $\boldsymbol{C} = (C_1, C_2, \cdots, C_M)$.

The following inclusions are an immediate consequence of the definitions of the corresponding regions and the fact that the allowable values of $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_M)$ for the slot-synchronized CCw/oFB are a subset of those for the unsynchronized CCw/oFB:

$$\mathscr{C}_{u0} \subset \mathscr{C}_u \subset \mathscr{C}_s \tag{2a}$$

$$\mathscr{C}_{u0} \subset \mathscr{C}_{s0} \subset \mathscr{C}_s. \tag{2b}$$

Our main result, which is proved in the following two sections, is that these four regions in fact coincide.

Theorem 1: For the M-user CCw/oFB,

$$\mathscr{C}_{u0} = \mathscr{C}_u = \mathscr{C}_{s0} = \mathscr{C}_s.$$

Moreover, the outer boundary of this common capacity and zero-error-capacity region $\mathscr{C}$ is the set of all points $\boldsymbol{C} = (C_1, C_2, \cdots, C_M)$ such that

$$C_i = p_i \prod_{\substack{j=1 \\ j \neq i}}^{M} (1 - p_j) \tag{3}$$

where $\boldsymbol{p} = (p_1, p_2, \cdots, p_M)$ is a vector satisfying

$$\boldsymbol{p} \geq \boldsymbol{0} \tag{4a}$$

and

$$\sum_{i=1}^{M} p_i = 1; \tag{4b}$$

and each such $\boldsymbol{C}$ is determined by a unique such $\boldsymbol{p}$.

We remark that conditions (4a) and (4b) are equivalent to saying that $\boldsymbol{p}$ is a *probability vector*. Thus, Theorem 1 states that there is a simple one-to-one correspondence between probability vectors and points on the outer boundary of $\mathscr{C}$.

Fig. 3 shows the region $\mathscr{C}$ for the $M = 2$ user CCw/oFB. For $M = 2$, a probability vector has the form $\boldsymbol{p} = (\gamma, 1 - \gamma)$ where $0 \leq \gamma \leq 1$. Equation (3) then gives $C_1 = \gamma^2$ and $C_2 = (1 - \gamma)^2$. Thus, the outer boundary of $\mathscr{C}$ is just the set of all points $\boldsymbol{C} = (C_1, C_2)$ such that $\boldsymbol{C} \geq \boldsymbol{0}$ and

$$\sqrt{C_1} + \sqrt{C_2} = 1.$$

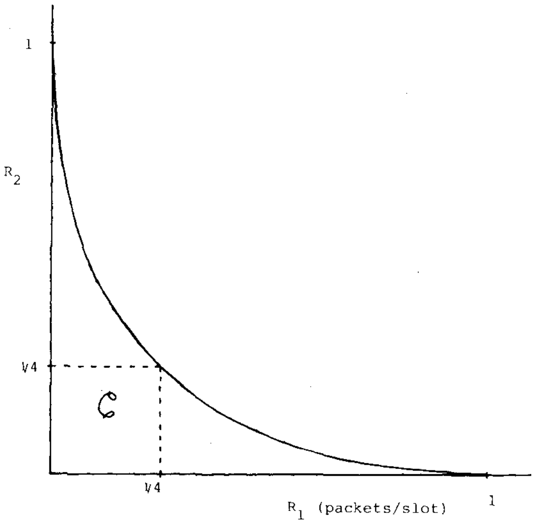

Fig. 3. Capacity region of two-user collision channel without feedback.

The region $\mathscr{C}$ in Fig. 3 is not convex, but it is easy to check that its *complement* in the first quadrant (i.e., the set of all $\boldsymbol{R}$ such that $\boldsymbol{R} \geq \boldsymbol{0}$ but $\boldsymbol{R} \notin \mathscr{C}$) is convex. This led us to conjecture earlier [2] that for every $M \geq 2$, the complement of $\mathscr{C}$ in the first "orthant" is convex—the correctness of this conjecture has been proved by Post [3].

The region $\mathscr{C}$ is not convex for any $M \geq 2$, as follows from the $M = 2$ case by consideration of that portion of the outer boundary corresponding to probability vectors with $p_i = 0$ for $3 \leq i \leq M$. This is the first instance known to us of a capacity (or zero-error capacity) region that is not convex.[1] As Shannon has pointed out [1], all capacity (and zero-error capacity) regions are convex if it is possible to time-share the coding schemes used to approach individual rate points of the region. The fact that $\mathscr{C}$ is not convex for the M-user CCw/oFB when $M \geq 2$ must thus be seen as a consequence of the fact that the lack of a common time reference prevents the users from time-sharing different coding schemes.

A rate vector $\boldsymbol{R}$ in a capacity (or zero-error capacity) region is said to be *achievable* (cf. [5, p. 5]) if $\boldsymbol{R}$ satisfies the above definition of an approachable rate with the change that $\delta = 0$. Interior points of a capacity (or zero-error capacity) region are always achievable, but boundary points may or may not be achievable. Our second main result is that, for the slot-synchronized CCw/oFB, the outer boundary is everywhere dense with achievable rates.

Theorem 2: Every open neighborhood of every point on the outer boundary of the capacity regions $\mathscr{C}_s$ and $\mathscr{C}_{s0}$ contains achievable rates that also lie on the outer boundary.

In a random-access system, one is usually most interested in the "symmetric case" where all users are signaling at the same rate. Thus, we define the *symmetric capacity*, C_{sym}, of the M-user CCw/oFB to be the maximum rate r such that $\boldsymbol{R} = (r/M, r/M, \cdots, r/M)$ is in $\mathscr{C}$. Note that if there is an r such that $\boldsymbol{C} = (r/M, r/M, \cdots, r/M)$ is on the outer boundary of $\mathscr{C}$, then $C_{\text{sym}} = r$. But, from (3) and (4), we see that the choice $\boldsymbol{p} = (1/M, 1/M, \cdots, 1/M)$ gives such a $\boldsymbol{C}$. This proves all but the final part of the following corollary.

Corollary to Theorem 1: The symmetric capacity of the M-user CCw/oFB with $M \geq 2$ (whether unsynchronized or slot-synchronized and whether for arbitrarily small positive error probability or for zero-error probability) is

$$C_{\text{sym}} = \left(1 - \frac{1}{M}\right)^{M-1} \text{packets/slot.} \tag{5}$$

Moreover, the rate point $(C_{\text{sym}}/M, C_{\text{sym}}/M, \cdots, C_{\text{sym}}/M)$ is achievable in the slot-synchronized case.

From (5) one calculates, for instance,

$$C_{\text{sym}} = \begin{cases} 1/2, & M = 2 \\ 4/9 \approx .444, & M = 3 \\ \approx .3874, & M = 10 \\ \approx .3678, & M = 100. \end{cases}$$

Moreover, C_{sym} decreases monotonically as M increases and

$$C_{\text{sym}} \to \frac{1}{e}, \qquad \text{as } M \to \infty. \tag{6}$$

[1] The "achievable region" of Wyner's wire-tap channel [4] is not convex, but this is not actually a capacity region as one of the coordinates is not an information rate.

The quantity $1/e$ is, of course, the well-known maximum throughput of the slotted ALOHA algorithm [6] for infinitely many identical users. Thus, (6) could perhaps be expected for the slot-synchronized case, although ALOHA algorithms make essential use of the feedback that is not present in our model. That (6) holds for the unsynchronized case seems truly surprising because the maximum throughput of the "pure" ALOHA algorithm [7] is only $1/2e$.

We remark here that C_{sym} is also the minimum of $C_1 + C_2 + \cdots + C_M$ for any point $C = (C_1, C_2, \cdots, C_M)$ on the outer boundary of $\mathscr{C}$. This confirms the intuition about random-access systems which claims that a given total amount of traffic is most difficult to serve when it is equally apportioned among the M users.

IV. Nonapproachability of Rates Outside $\mathscr{C}$

We now wish to show that rates outside the region $\mathscr{C}$, as defined in Theorem 1, cannot be approached for the CCw/oFB in either the slot-synchronized or unsynchronized case and for either error probability criterion. From (2), we see that it suffices to show that points outside $\mathscr{C}$ cannot be approached with arbitrarily small positive error probability in the slot-synchronized case. Thus, for the rest of this section, we consider only the slot-synchronized case.

Consider now any choice of protocol signals and codes for the users. Without loss of essential generality, we may assume that the period τ_i of the protocol signal $s_i(t)$ is a rational multiple of the slot length T, for $1 \le i \le M$. Thus, we can write $\tau_i = (m_i/m)T$ where m_i and τ_i are integers. Then NT, where $N = m_1 m_2 \cdots m_M$ is an integer multiple of each τ_i. Thus, for all t,

$$s_i(t + NT) = s_i(t) \tag{7}$$

for $1 \le i \le M$.

For purposes only of our proof, we now impose a fictitious probability distribution on the time offsets $\delta_1, \delta_2, \cdots, \delta_M$; namely, we specify that these are independent and identically distributed (IID) random variables that are equally likely to take on any of the N values $0, T, 2T, \cdots, (N-1)T$. It follows from (7), from the definition of the duty factor p_i, and from the fact that $s_i(t)$ is nonzero only over semi-open intervals of lengths which are integer multiples of T, that

$$E[s_i(t - \delta_i)] = p_i \tag{8}$$

for every time instant t.

At any given time instant t on the receiver's clock, user i will be the only user in the act of transmission if and only if

$$s_i(t - \delta_i) \prod_{j \ne i} [1 - s_j(t - \delta_j)] = 1; \tag{9}$$

Moreover, the left side of (9) will otherwise be zero. Thus, defining T_i as the total time within an arbitrary semi-open interval, $[t_0, t_0 + NT)$, on the receiver's clock of length NT during which the receiver is receiving noncollided packets from user i, we have

$$T_i \le \int_{t_0}^{t_0 + NT} s_i(t - \delta_i) \prod_{j \ne i} [1 - s_j(t - \delta_j)]\, dt; \tag{10}$$

the inequality is required by the fact that the satisfaction of (9) for some t does not ensure that the packet being sent at receiver time t by user i will not experience a "partial" collision. (Note that the users need not align their packets with a time slot even though the channel is slot-synchronized.) Taking expectations in (9) and making use of (8) and of the independence of $\delta_1, \delta_2, \cdots, \delta_M$ gives

$$E\left[s_i(t - \delta_i) \prod_{j \ne i} [1 - s_j(t - \delta_j)]\right] = p_i \prod_{j \ne i} (1 - p_j). \tag{11}$$

Now taking expectations in (10) and using (11) gives

$$E[T_i] \le NTp_i \prod_{j \ne i} (1 - p_j). \tag{12}$$

For any given i, it follows from (12) that there must be some specific choice of $\delta_1, \delta_2, \cdots, \delta_M$ such that

$$T_i \le NTp_i \prod_{j \ne i} (1 - p_j) \tag{13}$$

and, indeed, it was only to arrive at this conclusion that we introduced the fictitious probability distribution on $\delta_1, \delta_2, \cdots, \delta_M$.

We now recall that, according to the model of the CCw/oFB as given in Section II, the specific time intervals over which the received signal is indicating either "idle" or "collision" are determined entirely by the protocol signals and the time offsets. Thus, the information from the QSS of user i can affect the received signal at most during the T_i seconds of the interval $[t_0, t_0 + NT]$ when the receiver is receiving noncollided packets from user i. It follows from (12) that, given i, there is a specific choice of $\delta_1, \delta_2, \cdots, \delta_M$ such that the receiver receives noncollided packets from user i at a "rate" of at most $p_i \prod_{j \ne i}(1 - p_j)$ packets/slot. Suppose further that there is a friendly genie who identifies in advance, for both user i and the receiver, each interval in which user i sends a noncollided packet. Then user i has, with this extra help, a noiseless Q-ary DMC to the receiver with a capacity of one packet per use and with at most $p_i \prod_{j \ne i}(1 - p_j)$ uses per slot. Thus, by the usual coding theorem for a DMC, user i cannot send information from his QSS at a rate R_i with arbitrarily small positive error probability, regardless of the values $\delta_1, \delta_2, \cdots, \delta_M$, unless

$$R_i \le p_i \prod_{j \ne i} (1 - p_j) \text{ packets/slot.} \tag{14}$$

It follows that $\boldsymbol{R} = (R_1, R_2, \cdots, R_M)$ cannot be approached with arbitrarily small positive error probability, independent of $\boldsymbol{\delta}$, unless (14) is satisfied for $i = 1, 2, \cdots, M$.

To complete the proof that points outside $\mathscr{C}$ cannot be achieved, we need only show that every $\boldsymbol{R} \ge 0$ that satisfies (14) for $1 \le i \le M$ lies in the region $\mathscr{C}$ defined in Theorem 1, i.e., that if $\boldsymbol{R}$ satisfies (14) for $1 \le i \le M$ for some duty factor vector $\boldsymbol{p}$, then $\boldsymbol{R}$ also satisfies (14) for $1 \le i \le M$ for some probability vector $\boldsymbol{p}$. In fact, Abramson has already proved this last statement in his determination of the "achievable throughput region "for an M-user slotted ALOHA system [8] (cf. [9, pp. 365–369]). Nonetheless, in Appendix A, we give an elementary proof of the following lemma that also establishes this result; our proof also shows that each point $\boldsymbol{C}$ on the outer boundary of $\mathscr{C}$ is determined by a unique probability vector $\boldsymbol{p}$, as stated in Theorem 1. [We write $\mathbf{1}$ to denote the all-one vector $(1,1,\cdots 1)$ with M components.]

Lemma 1: For any $\boldsymbol{p}' = (p_1', p_2', \cdots, p_M')$ with $\mathbf{0} \leq \boldsymbol{p}' \leq \mathbf{1}$, there is a probability vector $\boldsymbol{p} = (p_1, p_2, \cdots, p_M)$ such that

$$p_i' \prod_{j \neq i} \left(1 - p_j'\right) \leq p_i \prod_{j \neq i} \left(1 - p_j\right), \qquad 1 \leq i \leq M.$$

V. Achievability of Rates on the Outer Boundary of $\mathscr{C}$

In this section, we will give a constructive proof of Theorem 2. In the following section, we shall use the results of this section to obtain a simple proof of the direct part of Theorem 1.

A. Preliminaries

Throughout this section, we will consider only the slot-synchronized case. As we are now dealing with constructive schemes, we can and do adopt the restriction that all users align their packet transmissions to fall within time slots on their local clocks, and hence also within time slots on the receiver's clock, since the time offsets are integer multiples of the slot length T. With no loss of generality, we take $T = 1$ so that each time offset δ_i is an integer. The period of each protocol sequence is now also an integer that we denote by N_i for user i, and we write N for the least common multiple of $N_1, N_2, \cdots, N_M$.

We can now equivalently describe the protocol signal $s_i(t)$ by the *protocol sequence* $\boldsymbol{s}_i = [s_{i1}, s_{i2}, \cdots, s_{iN}]$ in the manner that s_{in} is the value of $s_i(t)$ in the nth time slot $n \leq t < n + 1$. We further assume that a transmitted packet takes values in the set $\{0, 1, 2, \cdots, Q - 1\}$, and we write Λ to denote the silent signal ("idle") in a slot. Thus, we can denote the transmitted signal from user i in his own nth time slot by the discrete random variable $X_i(n)$ in the manner that

$$\begin{aligned} X_i(n) &= \Lambda, && \text{if } s_{in} = 0 \\ X_i(n) &\in \{0, 1, \cdots, Q - 1\}, && \text{if } s_{in} = 1. \end{aligned}$$

Similarly, we can denote the received signal in the nth time slot by the discrete random variable $Y(n)$ in the manner that

$$\begin{aligned} Y(n) &= \Lambda, && \text{if } X_i(n - \delta_i) = \Lambda \text{ for } 1 \leq i \leq M \\ Y(n) &= X_i(n - \delta_i), && \text{if } X_j(n - \delta_j) = \Lambda \text{ for all } j \neq i \\ Y(n) &= \Delta, && \text{otherwise,} \end{aligned}$$

where Δ denotes a collision of two or more packets. In this manner, we obtain a fully discrete representation for the slot-synchronized CCw/oFB. Note that the channel input alphabet of each user contains $Q + 1$ letters and that the channel output alphabet contains $Q + 2$ letters. Henceforth, we shall sometimes speak of "time instant n," rather than the "nth time slot."

We assume for convenience that the output alphabet of the QSS of each user is also the set $\{0, 1, \cdots, Q - 1\}$. We seek then to choose a protocol sequence and block code for each user such that, regardless of the values of the time offsets, the receiver can reconstruct each source output sequence without error. Moreover, we must show that the joint rates $\boldsymbol{R}$ that can be so achieved are dense on the outer boundary of the region $\mathscr{C}$ defined in Theorem 1.

B. Protocol Matrices and an Example

We define the *protocol matrix* S as the $M \times N$ binary matrix whose ith row is the protocol sequence $\boldsymbol{s}_i$ of user i. For instance, with $M = 2$ users, and protocol sequence periods $N_1 = 2$ and $N_2 = 4$ with least common multiple $N = 4$, we could choose

$$S = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \tag{15}$$

We shall be interested in the received sequence over a span of N consecutive time instants, which, with no loss of essential generality, we can take to be time instants $1, 2, \cdots, N$. We write

$$\boldsymbol{Y} = [Y_1, Y_2, \cdots Y_N]$$

to denote this received N-tuple. We see from (15) that, in case the time offsets are $\delta_1 = \delta_2 = 0$,

$$\boldsymbol{Y} = [\Delta, P_A, P_B, \Lambda],$$

i.e., that slot 1 is a collision slot, that slot 4 is idle, and that slots 2 and 3 contain packets. From (15), we see further that packet P_A was sent by user 2 whereas packet P_B was sent by user 1.

Suppose next that $\delta_1 = 5$. This delays the periodic protocol sequence of user 1 by 5 slots, so that it will appear to the receiver that user 1 is actually using the protocol sequence $[0, 1, 0, 1]$ in slots 1 through 4. Similarly, if $\delta_2 = 3$, it will appear to the receiver that user 2 is actually using the protocol sequence $[1, 0, 0, 1]$. Thus, it will appear to the receiver as if the modified protocol matrix

$$S[\boldsymbol{\delta}] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

is actually in use. In particular, we see that

$$\boldsymbol{Y} = [P_A, P_B, \Lambda, \Delta]$$

where the packets P_A and P_B are from users 2 and 1, respectively.

As we have observed from this example, a time offset (or "delay") of δ_i slots corresponds to δ_i right cyclic shifts of the protocol sequence $\boldsymbol{s}_i$. We write $\boldsymbol{s}_i[\delta_i]$ to denote the sequence obtained from $\boldsymbol{s}_i$ after δ_i right cyclic shifts and, as we have already done in (16), we write $S[\boldsymbol{\delta}]$ for the *effective protocol matrix* whose ith row is $\boldsymbol{s}_i[\delta_i]$. Note that $S = S[\mathbf{0}]$. Because $s_{in} = s_{i,n+N}$ for all i and n, it follows that $\boldsymbol{s}_i[\delta_i] = \boldsymbol{s}_i[\delta_i + N]$. Thus, given N, we can and do hereafter restrict ourselves to the condition

$$0 \leq \delta_i < N \tag{17}$$

without loss of essential generality. Because of (17), we see that there are only N^M values of $\boldsymbol{\delta} = [\delta_1, \delta_2, \cdots, \delta_M]$ to be considered, and hence at most this many distinct effective protocol matrices.

For the protocol matrix S of (15), the reader can easily check that all 16 choices of $\boldsymbol{\delta}$ result in an $S[\boldsymbol{\delta}]$ such that the resulting $\boldsymbol{Y} = [Y_1, Y_2, Y_3, Y_4]$ always contains one collision slot, one idle slot, and one packet from each of the two users. Moreover, the packet from user 2 is always adjacent to a collision slot [provided we count slot 1 as adjacent to slot N] whereas the packet from user 1 is never adjacent to a collision slot. Thus, the receiver can, from examination of $\boldsymbol{Y} = [Y_1, Y_2, Y_3, Y_4]$, uniquely identify the sender of each of the two successfully received packets in this sequence regardless of the values of the time offsets. Suppose further that each user employs the simple rate $r = 1/2$ packets/slot repeat code in which each information packet from his QSS is sent twice. Precisely one of these two packets will be correctly received, and its sender identified, as the other packet will be lost in a collision. Hence, the receiver can perfectly reconstruct the output sequence from each of the two QSS's. Note that user i is sending information packets at the rate $R_i = 1/4$ packets/slot for $i = 1$ and 2. (We note from (15) that the duty factors are $p_1 = p_2 = 1/2$, which, since the code rates are $r_1 = r_2 = 1/2$ packets/slot, also implies $R_i = p_i r_i = 1/4$ packets/slot for $i = 1$ and 2.) Thus, we have demonstrated a coding scheme that achieves the equi-rate point $\boldsymbol{C} = (1/4, 1/4)$ on the outer boundary of the zero-error capacity region of the two-user slot-synchronized CCw/oFB.

In the following subsections, we develop the appropriate generalization of this example. We shall show, in fact, how to achieve, with zero-error, any $\boldsymbol{C} = (C_1, C_2, \cdots, C_M)$ *on the outer boundary of $\mathscr{C}$ for which the corresponding probability vector $\boldsymbol{p} = (p_1, p_2, \cdots, p_M)$ of* (3) *has only rational components.*

C. Construction of Protocol Sequences

Any duty factor vector $\boldsymbol{p} = (p_1, p_2, \cdots, p_M)$ with only rational components may be written as $\boldsymbol{p} = (q_1/q, q_2/q, \cdots, q_M/q)$, where $q_1, q_2, \cdots, q_M$ are nonnegative integers and q is a positive integer that we assume is chosen as small as possible. We shall construct a special protocol matrix S for this $\boldsymbol{p}$, using as an intermediary a matrix with q-ary components. We write A_{Mq} to denote the $M \times q^M$ matrix whose jth column is the M place radix-q representation of the integer $q^M - j$, with the least significant digit at the top. For example, with $\boldsymbol{p} = (1/3, 2/3)$, we have $M = 2$ and $q = 3$ so we first construct

$$A_{23} = \begin{bmatrix} 2 & 1 & 0 & 2 & 1 & 0 & 2 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (18a)$$

We then obtain the desired protocol matrix, which we denote simply by S_{Mq} [although it also depends on the values of $q_1, q_2, \cdots, q_M$], by mapping, within the ith row of A_{Mq}, the q-ary digits $q-1, q-2, \cdots, q-q_i$ to 1's and mapping the q-ary digits $q - q_i - 1, \cdots, 1, 0$ to zeros. Continuing our example, we obtain (from (18a) and the fact that $q_1 = 1$ and $q_2 = 2$) the protocol matrix

$$S_{23} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (18b)$$

We write $A_{Mq}[\boldsymbol{\delta}]$ to denote the matrix whose ith row is the vector obtained by δ_i right cyclic shifts of the ith row of A_{Mq}. Note that $S_{Mq}[\boldsymbol{\delta}]$ is obtained if we apply the above q-ary to binary digit mapping to the entries of $A_{Mq}[\boldsymbol{\delta}]$.

Lemma 2: For every $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_M)$, the matrix $S_{Mq}[\boldsymbol{\delta}]$ can be obtained from S_{Mq} by a permutation of columns.

Proof: It suffices to prove that the columns of $A_{Mq}[\boldsymbol{\delta}]$ are a permutation of those of A_{Mq}. To prove this, we note that the first row of A_{Mq} (periodically repeated—as we shall always mean when we speak of the "period" of finite sequences) is a sequence of period q in which each q-ary symbol appears. But the symbols in the second row of A_{Mq} occur in runs of q identical symbols and this row has period q^2. Thus, no matter how the first and second rows are cyclically shifted, the first two rows of the resulting $A_{Mq}[\boldsymbol{\delta}]$ must, like A_{Mq}, form a submatrix in which each 2-place q-ary number appears as a column and in which the columns are periodic with period q^2. But the symbols in the third row of A_{Mq} occur in runs of q^2 identical symbols and this row has period q^3. Hence, the first three rows of $A_{Mq}[\boldsymbol{\delta}]$ must, like A_{Mq}, form a submatrix whose columns have period q^3 and in which each possible column appears. By a simple induction, we conclude that every q-ary M-tuple must appear exactly once as a column of $A_{Mq}[\boldsymbol{\delta}]$, and hence that the columns of $A_{Mq}[\boldsymbol{\delta}]$ are indeed just a permutation of those of A_{Mq}.

The practical import of Lemma 2 is that $\boldsymbol{Y} = [Y_1, Y_2, \cdots, Y_N]$, $N = q^M$, will contain the same number of collisions, the same number of idle slots and the same number of successes from user i, $1 \le i \le M$, regardless of the time offset $\boldsymbol{\delta}$, when we use the protocol matrix S_{Mq}. For example, from (18b), we see that for $\boldsymbol{\delta} = \boldsymbol{0}$

$$\boldsymbol{Y} = [\Delta, P_A, P_B, \Delta, P_C, P_D, P_E, \Lambda, \Lambda]$$

where the packets P_A, P_B, P_C and P_D are from user 2 whereas packet P_E is from user 1. Lemma 2 implies that $\boldsymbol{Y}$ will always contain two collisions, two idle slots, four successful packets from user 2 and one successful packet from user 1, regardless of the time offset $\boldsymbol{\delta}$, when the protocol matrix S_{23} of (18b) is used. Our next step is to show that the receiver can identify the sender of each successfully received packet, regardless of the time offset $\boldsymbol{\delta}$.

D. Decimation Decoding

By the kth *phase* of the dth *decimation* of a sequence $[a_1, a_2, \cdots, a_N]$, where d is a divisor of N, we shall mean the subsequence $[a_k, a_{k+d}, a_{k+2d}, \cdots]$ of length N/d obtained by selecting every dth digit of the sequence, commencing with the kth digit. The following lemma is the key to recognizing the sender of successfully received packets.

Lemma 3: For every $\boldsymbol{\delta}$, the effective protocol matrix $S_{Mq}[\boldsymbol{\delta}]$, which has $N = q^M$ columns, is such that, for every $d = q^j$ with $1 \le j < M$ and for every k with $1 \le k \le d$, the kth phase of the dth decimation of the received vector $\boldsymbol{Y} = [Y_1, Y_2, \cdots, Y_N]$ has the following two properties.

1) If $i \le j$, then user i is either active in each of the q^{M-j} slots of this phase or is silent in each of the q^{M-j} slots of this phase.

2) If $q_i < q$ for all i, then there is at least one slot of this phase where all users i with $i > j$ are silent.

Proof: Property 1) follows from the fact that the ith row of $A_{Mq}[\boldsymbol{\delta}]$ has period q^i, and hence the ith row of $S[\boldsymbol{\delta}]$ also has period q^i. If $i \leq j$, then $d = q^j$ is divisible by q^i so the kth phase of the dth decimation of the ith row of $S[\boldsymbol{\delta}]$ must have the same entry [0 or 1] in every position, and this is precisely the meaning of property 1).

To prove property 2), we note that, if $d = q^j$ where $i < j < M$, then any phase of the dth decimation of the ith row of A_{Mq} is just the $(j-i)$th row of $A_{M-j,q}$. The matrix $A_{Mq}[\boldsymbol{\delta}]$ inherits this property in the sense that such a dth decimation of its ith row is the $(j-i)$th row of $A_{M-j,q}[\boldsymbol{\delta}']$ for some offset $\boldsymbol{\delta}'$. It follows that there must be some slot in the dth decimation, $d = q^j$, where the corresponding column of $A_{Mq}[\boldsymbol{\delta}]$ has only zeros in rows $j+1, j+2, \cdots, M$. Because $q_i < q$ for all i, all zeros in A_{Mq} are converted to zeros in S; hence users $j+1, j+2, \cdots, M$ must all be silent in this slot.

Note that decimating j times by q is the same as one decimation by q^j. This fact, together with Lemma 3, establishes the validity of the following procedure, which we call *decimation decoding*, by which the receiver can identify the sender of each successfully received packet contained in $\boldsymbol{Y} = [Y_1, Y_2, \cdots, Y_N]$ when the protocol matrix S_{Mq} is used. We assume that $q_i < q$ for all i. The case $q_i = q$, and hence $p_i = q_i/q = 1$, is interesting only for those trivially obtained points $\boldsymbol{C}$ on the outer boundary of $\mathscr{C}$ where $C_i = 1$ and $C_j = 0$ for $j \neq i$.

1) Form the q distinct phases of the qth decimation of $\boldsymbol{Y}$. Identify as coming from user 1 all successfully received packets in phases containing no idle slot. Set $i = 2$.
2) Form the q distinct phases of the qth decimation of each of the phases with an idle slot that were formed in the previous step. Identify as coming from user i all successfully received packets in those newly formed phases containing no idle slot. If $i = M$, stop. Otherwise, increase i by 1 and repeat this step.

As an example, suppose that $M = 2$ users have the protocol matrix S_{23} of (18b) and that $\boldsymbol{\delta}$ is such that the received vector is

$$\boldsymbol{Y} = [\Lambda, P_A, \Lambda, P_B, \Delta, P_C, P_D, \Delta, P_E].$$

Decimating by $q = 3$, as called for in step 1), yields the three phases

$$[\Lambda, P_B, P_D], [P_A, \Delta, \Delta], [\Lambda, P_C, P_E].$$

Only the second of these phases contains no idle slot; thus, only packet P_A is identified as having been sent by user 1. Of course, because $M = 2$, the other four packets in $\boldsymbol{Y}$ were sent by user 2. But, in principle, we find this out according to step 2) by decimating by $q = 3$ the first and third of the above phases to obtain the six phases

$$[\Lambda], [P_B], [P_D], [\Lambda], [P_C], [P_E].$$

The packets in these phases with no idle slot, i.e., packets P_B, P_D, P_C, and P_E, are now identified as having been sent by user 2.

It must be pointed out that, although decimation decoding identifies the sender of all successfully received packets, it does not in general identify the position of such a packet in the (unshifted) protocol sequence of the sender. This knowledge is not always required to decode the block code that the user has employed for his packets, as the example in Subsection V-B shows, but it is required in general. Thus, we need some scheme by which such packet location information can be obtained by the receiver. Note that it is necessary to locate only one packet for each user, and note that this packet location information allows the receiver to construct the effective protocol matrix $S_{Mq}[\boldsymbol{\delta}]$ and thus to identify the subset of users participating in each collision contained in $\boldsymbol{Y}$.

E. Finding Packet Locations

We assume that, for each user, there is some finite time in the past when that user first began to transmit information from his QSS, and we further assume that this user transmitted the zero packet, $P = 0$, in all previous slots into the infinite past in which he was required by his protocol sequence to send a packet. Note that user i sends Np_i packets during one cycle of his protocol sequence $\boldsymbol{s}_i$; we shall call these Np_i packets a *frame*. When user i is ready to send information from his QSS, he first sends the frame $[1, 1, \cdots, 1]$ consisting only of packets $P = 1$. He then sends successively the following Np_i frames containing one 1 packet: $[1, 0, \cdots, 0, 0]$, $[0, 1, \cdots, 0, 0], \cdots$, $[0, 0, \cdots, 1, 0]$, $[0, 0, \cdots, 0, 1]$.

The receiver will see only 0 packets, idle slots and collisions into the infinite past. As soon as the receiver identifies, by decimation decoding, a packet $P = 1$ from user i, he begins to count the number of slots, taken at intervals of N slots, until packet $P = 1$ again appears in this slot. This number is the location of this slot in user i's frame, and this allows the receiver to locate this slot in user i's protocol sequence.

It remains only to formulate an appropriate coding scheme for user i so that he can code the information packets from his QSS into his transmitted frame at the desired rate and in such a way that the receiver can always correctly decode these packets.

F. Coding the Packets

The matrix A_{Mq} has $q_i \prod_{j \neq i}(q - q_j)$ columns in which the entry in row i is a digit equal to or greater than $q - q_i$ but in which the entry in each other row j is an integer less than $q - q_j$. It follows then from the construction of S_{Mq} and Lemma 2 that, for every $\boldsymbol{\delta}$, the matrix $S_{Mq}[\boldsymbol{\delta}]$ will be such that $\boldsymbol{Y}$ contains exactly $q_i \prod_{j \neq i}(q - q_j)$ successfully received packets from user i. Thus, *provided we can find a coding scheme that allows each user i to send one information packet without error to the receiver for each successfully received packet from that user*, user i will be transmitting with zero-error probability at the rate

$$\begin{aligned} R_i &= \frac{1}{N} q_i \prod_{j \neq i} (q - q_j) \\ &= p_i \prod_{j \neq i} (1 - p_j) \text{ packets/slot}, \end{aligned} \tag{19}$$

where we have used the fact that $N = q^M$ and that the duty factor p_i is given by $p_i = q_i/q$. Thus, our proof of Theorem 2 will be complete if we can find a coding scheme that meets this proviso.

Note that the packets from user i that are involved in collisions are equivalent to "erasures" for the decoding of packets from user i. We assume that user i employs a block code of length $n_i = Np_i$ to code his $k_i = NR_i$ packets. This (n_i, k_i) code must be capable of correcting any of the patterns of $n_i - k_i$ erasures within a block that are consistent with $S_{Mq}[\boldsymbol{\delta}]$ for some $\boldsymbol{\delta}$.

If $Q = 2^m$ for some m such that $Q \geq n_i - 1$ (as might be expected in practice), the coding problem for user i is trivial in principle. He can simply use a Reed–Solomon (RS) code over GF(2^m), possibly extended to length $Q + 1$ [10]. Such an (n_i, k_i) RS code has minimum distance $n_i - k_i + 1$ and hence can correct *every* pattern of $n_i - k_i$ erasures. But this is not a satisfactory solution for our purposes, since we have insisted that the packet alphabet size Q can be as small as 2. We thus must construct appropriate codes over the alphabet $\{0, 1, \cdots, Q-1\}$ for any $Q \geq 2$. In fact, we shall see that it suffices to correct *bursts* of consecutive erasures. The following lemma is the key to our coding scheme.

Lemma 4: For any integers Q, n, and k with $Q \geq 2$ and $1 \leq k \leq n$, there exists an (n, k) systematic linear code over Z_Q, the ring of integers modulo Q, that corrects all closed-loop erasure bursts of length $n - k$.

By a "closed-loop" erasure burst, we mean that position 1 in the block is assumed to follow position n so that a burst can begin near the end of the block and continue into the beginning of this same block. By an (n, k) systematic linear code over Z_Q, we mean that the first k symbols in the block can be arbitrarily chosen (the "information symbols") and the remaining $n - k$ symbols (the "parity symbols") computed as linear combinations in Z_Q of these information symbols. We shall prove Lemma 4 by constructing the code whose existence is asserted. First, however, we show how these codes can be used to obtain the information rate R_i of (19) when the protocol matrix S_{Mq} is used.

For convenience, we refer to the code described in Lemma 4 as a *maximum-erasure-burst-correcting* (*MEBC*) *code*. We now show inductively how to nest such MEBC codes to obtain the desired coding system for user 1.

User 1 will actually use q_1 independent, but identical, codes of block length $n = q^{M-1}$, one of which will be used to code the packets sent during each of the q_1 phases of the qth decimation of $\boldsymbol{s}_1$ that consist only of ones. We describe the code used by user 1 for the packets sent during phase 1 of the qth decimation of his protocol sequence.

We see, from the fact that each phase of the qth decimation of the second row of A_{Mq} has period q, that, in any q successive slots during phase 1 of the qth decimation of user 1's protocol sequence, the packets from user 2 will occur as a (closed-loop) burst of length q_2 packets occurring with a period of q slots. Thus, if, as we now assume to be the case, the packets from user 1 in each successive q slots of phase 1 of his protocol sequence form a codeword in an $(n = q,\ k = q - q_2)$ MEBC, then the decoder will be able to determine all user 1's packets correctly, *provided that users 1 and 2 are the only active users.*

We now assume that an $(n = q^{M-2},\ k = \prod_{j=2}^{M-1}(q - q_j))$ code has been found for user 1's packets during phase 1 of his protocol sequence that allow the decoder to correct all erasure patterns that can result when users $1, 2, \cdots, M-1$ are the only active users. We must show that we can extend this to an $(n = q^{M-1},\ k = \prod_{j=2}^{M}(q - q_j))$ code that will correct all erasure patterns that are possible when users $1, 2, \cdots, M$ are all active. We construct this code by specifying: that a codeword be the concatenation of q codewords from the former code of length $n = q^{M-2}$; that the first $q - q_M$ of these q codewords can be arbitrarily selected; and that the last q_M of these q codewords be determined by the rule that the digits in every phase of the q^{M-2}th decimation of the entire codeword must be a codeword in an $(n = q,\ k = q - q_M)$ code. This new code obviously has the claimed length $n = q^{M-1}$ and claimed number of information packets $k = \prod_{j=2}^{M}(q - q_j)$. It remains only to show that the decoder can correct any pattern of erased packets that can occur with all M users active.

The decoder, for the full codeword of length $n = q^{M-1}$ just described, first forms each phase of the q^{M-2}th decimation of the received codeword. But each such phase is also a phase of the q^{M-1}th decimation of the vector $\boldsymbol{Y}$ received over the channel in some $N = q^M$ consecutive slots. Lemma 3 thus implies that the *only* collisions in any such phase that would not also be collisions when only users $1, 2, \cdots, M-1$ were active must occur within phases where *all* the collisions are only between user 1 and user M. From the structure of A_{Mq}, we see that user M causes an erasure burst of length q_M in such a phase. But, by our construction, the packets from user 1 form a codeword in an $(n = q,\ k = q - q_M)$ code in such a phase. Thus, the decoder can at the outset correct all erasures resulting from collisions involving user M that would not also have been erasures when only users $1, 2, \cdots, M-1$ were active. The decoding problem then reduces to that for decoding q codewords of the length $n = q^{M-2}$ code to correct the erasures caused by users $2, 3, \cdots, M-1$, which, by hypothesis, can be done.

We now show that essentially the same coding strategy just developed for user 1 can be used by all M users. To see this, let α denote any chosen q-ary digit, i.e., any digit in $\{0, 1, \cdots, q-1\}$. If one modifies the matrix A_{Mq} first by deleting all columns in which the entry in a chosen row, say row i, is not the digit α, and next by deleting row i, then one obtains the matrix $A_{M-1,q}$. This follows from the fact that the columns of A_{Mq} contain the M-place q-ary numbers in natural order. It is then easy to see that the coding strategy just developed for user 1 applies directly to user i with the roles of users $2, 3, \cdots, M$ in the former scheme being played by users $1, 2, \cdots, i-1, i+1, \cdots M$, respectively, in the coding scheme for user i. It is only necessary that each codeword of the length $n = q^{M-1}$ code for user i be placed in those slots in which the ith row of A_{Mq} contains the same digit α.

It remains finally to construct the MEBC code described in Lemma 4. We do this by specifying the systematic generator matrix G for this code, where G is a $k \times n$ matrix with entries in $Z_Q = \{0, 1, \cdots, Q-1\}$ whose first k columns form an identity matrix. In fact, the entries in G will take values only in the subset $\{0, 1\}$ of Z_Q so that the form of G does not depend on the particular value of Q.

Our construction is perhaps best explained by an example, for which we choose $n = 64$ and $k = 27$. We first divide $k = r_0$ into n to obtain the quotient $q_0 = 2$ and remainder $r_1 = 10$. We then divide r_1 into r_0 to get a new quotient $q_1 = 2$ and a new remainder $r_2 = 7$. Continuing in this manner until the remainder is zero, we obtain in this instance the following values:

$$r_0 = 27, \; r_1 = 10, \; r_2 = 7, \; r_3 = 3, \; r_4 = 1$$
$$q_0 = 2, \quad q_1 = 2, \quad q_2 = 1, \; q_3 = 2, \; q_4 = 3.$$

These values specify the following 27×64 systematic generator matrix:

$$G = \left[\begin{array}{c|c|c} I_{27} & I_{27} & \begin{array}{c} I_{10} \\ \hline I_{10} \\ \hline \begin{array}{c|c} I_7 & \begin{array}{c} I_3 \\ \hline I_3 \\ \hline \begin{array}{c|c|c} I_1 & I_1 & I_1 \end{array} \end{array} \end{array} \end{array} \end{array}\right]$$

where I_m denotes an $m \times m$ identity matrix. The matrix G is formed by starting with a row of q_0 matrices I_{r_0}; then, starting from the top, adding a column of q_1 matrices I_{r_1}; then, starting at the left, adding a row of q_2 matrices I_{r_2}, etc. The proof that this construction always yields the systematic generator matrix of a Q-ary (n, k) code that can correct all closed-loop erasure bursts of length $n - k$ is not especially insightful, and thus is deferred to Appendix B.

G. Completing the Proof of Theorem 2

In the previous subsections, we have shown that the M users of the slot-synchronized CCw/oFB can send information without error at the joint rate $\boldsymbol{R} = \boldsymbol{C}$ for every point $\boldsymbol{C}$ on the outer boundary for $\mathscr{C}$ for which the corresponding probability vector $\boldsymbol{p}$ has only rational components. But every open neighborhood of every probability vector contains probability vectors with only rational components. Moreover, the mapping (3) from probability vectors $\boldsymbol{p}$ to points $\boldsymbol{C}$ on the outer boundary of $\mathscr{C}$ is continuous. It follows that every open neighborhood of every point $\boldsymbol{C}$ of the outer boundary of $\mathscr{C}$ contains outer boundary points that correspond to probability vectors with only rational components and that thus are achievable with zero-error probability, which is the assertion of Theorem 2.

VI. Approachability of Rates in $\mathscr{C}$

To prove the direct part of Theorem 1, we see from (2) that it suffices to show that any rate vector $\boldsymbol{R}$ in the region $\mathscr{C}$ defined in Theorem 1 can be approached without error for the unsynchronized CCw/oFB. However, we first show that such $\boldsymbol{R}$ can be approached without error for the slot-synchronized CCw/oFB, and then we give a simple argument that reduces the unsynchronized case to the slot-synchronized case.

A. The Slot-Synchronized Case

Let $\boldsymbol{R}$ be any vector in $\mathscr{C}$ as defined in Theorem 1. Note that $\boldsymbol{R}$ can be on the boundary or even on the outer boundary of $\mathscr{C}$. But, in any case, there must exist a point $\boldsymbol{C}'$ (possibly $\boldsymbol{R}$ itself) on the outer boundary of $\mathscr{C}$ such that $\boldsymbol{R} \leq \boldsymbol{C}'$. Hence, for any given positive δ, $\boldsymbol{R} - \delta\mathbf{1} < \boldsymbol{C}'$. It now follows from Theorem 2 that there is a point $\boldsymbol{C}$ on the outer boundary of $\mathscr{C}$ that is achievable with zero error in the slot-synchronized case and for which $\boldsymbol{R} - \delta\mathbf{1} < \boldsymbol{C}$. Therefore, $\boldsymbol{R}$ is indeed approachable in the slot-synchronized case.

B. The Unsynchronized Case

Since we are dealing with constructive coding schemes, we can and do enforce the provision that all users must align their packet transmissions to fall within time slots on their local clocks, even in the unsynchronized case that we now consider. Of course, because the components of $\boldsymbol{\delta}$ are now arbitrary real numbers, received packets will in general not fall into time slots on the receiver's clock.

By virtue of our restriction on packet transmission, we can still describe the protocol signals in the unsynchronized case by protocol sequences and protocol matrices as in Section V. (The slot length will again be taken for convenience as $T = 1$.) The following result, because of the arbitrariness of m, shows that any rate approachable without error in the slot-synchronized case is also approachable without error in the unsynchronized case. We write 0^m and 1^{m-1} to denote, respectively, a string of m zeroes and a string of $m-1$ ones.

Lemma 5: Suppose that the protocol matrix S, together with a given code for each user, yields error-free operation at the joint rate $\boldsymbol{R}$ on the M-user, slot-synchronized, CCw/oFB. Let the protocol matrix $S^{(m)}$ be constructed from S by replacing each zero in S with 0^m and each one in S with $1^{m-1}0$, where m is an arbitrary positive integer. Then the protocol matrix $S^{(m)}$, together with interleaving $m-1$ times the code previously given to each user, yields error-free operation at the joint rate $((m-1)/m)\boldsymbol{R}$ on the M-user, unsynchronized, CCw/oFB.

For example, with $M = 3$ and taking S to be the protocol matrix (15), we would have

$$S^{(3)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

We saw in Section V-B that the protocol matrix S, together with $r = 1/2$ repeat codes for both users, yielded error-free operation at $\boldsymbol{R} = (1/4, 1/4)$ on the slot-synchronized CCw/oFB. Lemma 5 asserts that the protocol matrix $S^{(3)}$ of (20), together with two interleaved $r = 1/2$ repeat codes for each user, will yield error-free operation at $\boldsymbol{R} = (1/6, 1/6)$ on the unsynchronized CCw/oFB.

To reduce the unsynchronized case to the slot-synchronized case, we argue as follows. If user i were the only active user in the unsynchronized case, then his packets would fall into "virtual time slots" at the receiver whose edges would occur at noninteger times because of the time offset δ_i that is in general not an integer. Fig. 4(a) illustrates this situation. The packets of another user, say user j, would not be aligned with these virtual time slots for user i, as illustrated in Fig. 4(b), because $\delta_i - \delta_j$ will not in general be an integer. However, the effect of these packets from user j on the packets of user i is precisely the same with regard to idle slots and to successes for user i as if the packets from user j were advanced (by less than one slot) in time to alignment with the virtual time slots for user i and then an additional dummy packet were inserted after each run of consecutive packets; the equivalence is illustrated in Fig. 4(c). We can summarize these observations as follows. *Provided that all M users align their packet transmissions with time slots on their own local clocks, then, in the unsynchroinized case, the resulting pattern of idle slots and of successes by user i is the same as in the slot-synchronized case, provided that a dummy packet is inserted after each run of consecutive packets from every user j for which $\delta_i - \delta_j$ is not an integer.* In what follows, we shall make the pessimistic assumption that $\delta_i - \delta_j$ is not an integer for all $j \neq i$ when considering packet transmissions from user i.

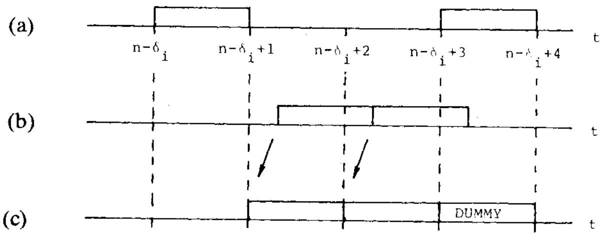

Fig. 4. (a) Packets from user i. (b) Packets from user j, as seen on receiver's clock. (c) Equivalent packets from user j in user i's virtual time slots.

As an example, when the protocol matrix $S^{(3)}$ of (20) is used, the packet transmissions from user 1 can be studied by replacing $S^{(3)}$ with the matrix

$$S^{(3,1)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

in which the runs of ones in the second row of $S^{(3)}$ have been extended by one. Note that, if we take third decimations of the *columns* of $S^{(3,1)}$, the three phases are just the following matrices:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

The first $m - 1 = 2$ of these matrices are just the original protocol matrix S of (15) that was used to construct $S^{(3)}$. By the construction of Lemma 5, each of these protocol matrices will be used by user 1 with an $r = 1/2$ repeat code, and hence each will deliver one packet error-free. User 1 sends no packets in the slots corresponding to the last of these $m = 3$ matrices. Hence user 1 sends error-free at a rate $R = 2/12 = 1/6$ packets/slot, as does user 2.

The truth of Lemma 5 should now be evident. If we restrict ourselves to consideration of packets from user i, the effect of other users on these packet transmissions is equivalent to that in the slot-synchronized case if each occurrence in $S^{(m)}$ of $1^{m-1}0$ in row j, all $j \neq i$, is replaced by 1^m. But the first $m - 1$ phases of the mth decimation of the columns of this resulting matrix, $S^{(m,i)}$, are all just the original matrix S, while the last phase is a matrix whose ith row is all zeroes. In each of these first $m - 1$ phases, user i, by hypothesis, uses a code that guarantees error-free coding at a rate R_i packets/slot. User i is silent in the last phase. Hence, user i sends information error-free precisely at the rate $((m-1)/m)R_i$ packets/slot, as claimed in Lemma 5.

A remark on the decoding process is in order. Upon seeing an uncollided packet, the receiver will extrapolate virtual time slots to align with its edges, then de-interleave the received symbols (packets, idles or collisions) in these virtual time slots into m streams. The receiver then applies the usual decoding procedure (decimation decoding followed by decoding of the block code of the user whose packets are found) to each of these streams. This will succeed in general only in decoding the packets of that user i who sent the original uncollided packet, as packets from other users will fall across time slots and must be treated as collisions. Thus, a similar de-interleaving and decoding process must be carried out for each of the M users.

VII. Prior Work and Some Remarks

We have already mentioned in Section IV that the region $\mathscr{C}$ of Theorem 1 coincides with the "achievable throughput region" determined by Abramson [8] (cf. [9, pp. 365–369]) for an M-user slotted ALOHA system. Abramson considered the situation where user i sends a packet in each slot with probability p_i, independent of previous transmissions. This corresponds in the language of this paper to using stochastic protocol sequences, each of which is an independent identically distributed (IID) sequence. In a certain sense, the "time statistics" of the protocol sequences constructed in Section V are the same as the ensemble statistics of Abramson's stochastic protocol sequence, but the former have additional structure that permits the receiver to identify the sender of each successfully received packet and that guarantees that the number of successes is independent of the time offsets. Reliable communication with stochastic protocol sequences seems to necessitate header information in packets to identify their sender and also seems to rule out error-free operation when there is no feedback link.

Tsybakov and Mikhailov [11] showed that Abramson's achievable throughput region $\mathscr{C}$ coincides with the "ergodicity region" of a slotted ALOHA system with feedback, i.e., with the region of joint user rates for which the retransmission processes can be stabilized. It seems somewhat surprising that precisely the same set of rates can be achieved error-free without feedback, as we have shown.

The only explicit prior work on random accessing without feedback, of which we are aware, is that of Huber and Shah [12], who were interested in applications to alarm systems. They considered the fully unsynchronized case with equal user rates. They used IID protocol sequences and achieved a symmetric throughput approaching $1/(2e)$ packets/slot as the number of users approached infinity. Lemma 5 of our paper suggests that, to approach a throughput of $1/e$ packets/slot with stochastic protocol sequences, one must have statistical dependence of successive digits in the protocol sequences.

All of the results and constructions in this paper were orally presented without detailed proofs by the first author on two occasions in 1982 [2], [13]. However, the abstract of [13] gives only the symmetric capacity results—the extension to the full capacity region was done jointly by the two authors in the time between submission of the abstract and summary of [13] and its oral presentation. Armed only with this abstract and summary, Tsybakov and Likhanov [14] independently derived the capacity region $\mathscr{C}$ for the CCw/oFB on the assumption that the packet size Q was equal to the order of a finite field and was sufficiently large to permit use of a maximum distance separable code for the packets of each user. Their work also has several interesting differences from that in this paper, such as a more thorough examination of the nature of the protocol matrices that can be used. For the equal-rate case, the abstract and summary of [13] used protocol sequences different from those now used in this paper; Cohn [15] independently suggested the same protocol sequences, for the equal-rate case, that are now used here.

Appendix A
Proof of Lemma 1

The claim of Lemma 1 is trivially true if $\boldsymbol{p}' = 0$, if $p_i' = 1$ for some i, or if $\boldsymbol{p}'$ has only one nonzero component. Thus, we restrict our attention to the case where $\mathbf{0} \le \boldsymbol{p}' < \mathbf{1}$ and where $\boldsymbol{p}'$ has at least two nonzero components. For each β, $0 < \beta < \infty$, we define a vector $\boldsymbol{p}$ by

$$p_i = p_i'/\left[p_i' + \beta(1-p_i')\right], \qquad 1 \le i \le M, \tag{A1}$$

and we note that, if $p_i' > 0, p_i$ decreases monotonically from one to zero as β increases from $0+$ to ∞. Hence, there is a unique value of β, say β_0, such that $\Sigma_i p_i = 1$, i.e., such that $\boldsymbol{p}$ is a probability vector. We also note that $\boldsymbol{p} = \boldsymbol{p}'$ if and only if $\beta = 1$.

From (A1), we obtain

$$p_i \prod_{j \ne i} (1-p_j) = f(\beta) p_i' \prod_{j \ne i} (1-p_j') \tag{A2}$$

for all i, where

$$f(\beta) = \beta^{M-1} \Big/ \prod_i \left[p_i' + \beta(1-p_i')\right]. \tag{A3}$$

We note that $f(0+) = f(\infty) = 0$ and $f(1) = 1$, and that the derivative $f'(\beta)$ is continuous on $0 < \beta < \infty$. A straightforward differentiation gives

$$f'(\beta) = \left[\sum_i p_i - 1\right] g(\beta)$$

where $g(\beta) > 0$ for $0 < \beta < \infty$. Thus, $f'(\beta) = 0$ if and only if $\beta = \beta_0$. It follows that $\beta = \beta_0$ uniquely maximizes $f(\beta)$ and this, because of (A2), proves Lemma 1.

We have now shown that $\boldsymbol{C}$ as defined by (3) can be on the outer boundary of $\mathscr{C}$ only if $\boldsymbol{p}$ is a probability vector. Suppose then that $\boldsymbol{p}$ is a probability vector but the corresponding $\boldsymbol{C}$ is not on the outer boundary of $\mathscr{C}$. Then, there is a $\boldsymbol{p}^*$, $0 \le \boldsymbol{p}^* \le 1$, such that

$$p_i \prod_{j \ne i} (1-p_j) \le p_i^* \prod_{j \ne i} (1-p_i^*), \qquad 1 \le i \le M, \tag{A4}$$

with strict inequality for at least one i, say $i = 1$. By decreasing only the first component of $\boldsymbol{p}^*$, which increases the right side of (A4) for $j > 1$, we can obtain a new $\boldsymbol{p}^*$ such that strict inequality holds in (A4) for all i. We can then appropriately decrease the components of $\boldsymbol{p}^*$ to obtain a $\boldsymbol{p}'$, $\mathbf{0} \le \boldsymbol{p}' \le \mathbf{1}$, such that

$$p_i \prod_{j \ne i} (1-p_j) = \alpha p_i' \prod_{j \ne i} (1-p_j'), \qquad 1 \le i \le M, \tag{A5}$$

where $\alpha < 1$. But (A5) implies that $\boldsymbol{p}$ and $\boldsymbol{p}'$ satisfy (A1) for some β, $0 < \beta < \infty$ and hence that $\alpha = f(\beta)$. This, together with the fact that $\boldsymbol{p}$ is a probability vector, implies the contradiction $\alpha = f(\beta_0) \ge 1$. We conclude that $\boldsymbol{C}$ as defined by (3) is on the outer boundary of $\mathscr{C}$ if and only if $\boldsymbol{p}$ is a probability vector. Moreover, distinct probability vectors $\boldsymbol{p}$ and $\boldsymbol{p}'$ must give distinct corresponding points $\boldsymbol{C}$ and $\boldsymbol{C}'$, respectively, for otherwise (A5) would be satisfied with $\alpha = 1$ and this would again imply that $\boldsymbol{p}$ and $\boldsymbol{p}'$ satisfy (A1) for some β and thus that $\boldsymbol{p} = \boldsymbol{p}'$.

Appendix B
Proof of Lemma 4

A systematic generator matrix for an (n, k) linear code over Z_Q, the ring of integers modulo Q, is a matrix G of the form $G = [I_k : P]$ where P is some $k \times (n-k)$ matrix over Z_Q. Such a G defines a systematic encoding rule in which the information vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_k]$ is mapped to the codeword $\boldsymbol{y} = [y_1, y_2, \cdots, y_n]$ in the manner $\boldsymbol{y} = \boldsymbol{x}G$ so that $y_i = x_i$ for $i = 1, 2, \cdots, k$. We have said that such a G defines a maximum-erasure-burst-correcting (MEBC) code if $\boldsymbol{x}$ can still be determined when any $n-k$ consecutive components of $\boldsymbol{y}$ are erased (where position 1 is considered to follow position n). Equivalently, G specifies an MEBC code if and only if each set of k consecutive columns of G forms an invertible matrix over Z_Q (where column 1 is considered to follow column n). We now write $G_{(i)}$ to denote the $k \times k$ submatrix formed by columns, $i, i+1, \cdots i+k-1$ of G, where by column j of G we understand column $j-n$ when $j > n$. Note that $G_{(1)} = I_k$. With this notation, the $k \times n$ matrix $G = [I_k : P]$ is the systematic generator matrix of a Q-ary MEBC code if and only if the matrix $G_{(i)}$ is invertible for $1 \le i \le n$. [A square matrix over Z_Q is invertible if and only if its determinant, computed over the integers, is an integer relatively prime to Q.] Hereafter, all matrices are assumed to be over Z_Q.

Proposition 1: The matrix $G_{(i)}$ corresponding to the $k \times n$ matrix $G = [I_k \ : \ P]$ is invertible for all i with $n - k < i \leq n$ if and only if the $m \times m$ submatrix $L_m(G)$ found in the last m rows and last m columns of G is invertible for all $m, 1 \leq m \leq k$.

Proof: This proposition follows directly from the fact that, for $n - k < i \leq n$,

$$G_{(i)} = \left[\begin{array}{c|c} A & I_{k+i-n-1} \\ \hline L_{n+1-i}(G) & 0 \end{array}\right]$$

where, here and hereafter, A denotes a matrix of appropriate dimension whose entries are of no interest.

Proposition 2: If G is the systematic generator matrix of an (N, K) MEBC code, then

$$G' = \left[I_N \ : \ G^T\right]$$

(where the superscript T denotes transpose) is the systematic generator matrix of an $(n = N + K,\ k = N)$ MEBC code.

Proof: We must show that $G'_{(i)}$ is invertible for $1 \leq i \leq n$. This is trivial for $i = 1$; for $1 < i \leq n$, we distinguish three cases.

Case 1: $1 < i \leq K + 1$. This gives

$$G'_{(i)} = \left[\begin{array}{c|c} 0 & I_{i-1} \\ \hline I_{k-i+1} & A \end{array}\right],$$

which is clearly invertible.

Case 2: $k < i \leq n$. Because $n + 1 - i \leq n - k = K$, this gives

$$G'_{(i)} = \left[\begin{array}{c|c} A & I_{i-K-1} \\ \hline L_{n+1-i}(G)^T & 0 \end{array}\right].$$

But G specifies an (N, K) MEBC code so Proposition 1 ensures that $L_{n+1-i}(G)$ is invertible and thus also its transpose is invertible. Hence, $G'_{(i)}$ is also invertible.

Case 3: $K + 1 < i \leq k$. This gives

$$G'_{(i)} = \left[\begin{array}{c|c|c} \begin{array}{c} 0 \\ \hline I_{k-i+1} \end{array} & G^T & \begin{array}{c} I_{i-K-1} \\ \hline 0 \end{array} \end{array}\right].$$

It follows that $G'_{(i)}$ is invertible if and only if the $K \times K$ submatrix consisting of rows $i - K, i - K + 1, \cdots, i - 1$ of G^T is invertible. But this submatrix is the transpose of G_{i-K}, which, because G specifies an MEBC code, is invertible.

If $G = [I_K \ : \ P]$ specifies an (N, K) MEBC code, then obviously the matrix $G' = [I_N \ : \ G^T]$ specifies an $(n = N + K,\ k = N)$ code. This fact, together with Proposition 2, implies the following key result.

Proposition 3: If G is the systematic generator matrix of an (N, K) MEBC code and q is any positive integer, then

$$G' = \left[I_N \ : \ I_N \ : \ \cdots \ : \ I_N \ : \ G^T\right]$$

(where there are q occurrences of I_N in G') is the systematic generator matrix of an $(n = qN + K,\ k = N)$ MEBC code.

Proposition 3 implies the validity of the MEBC code construction described in Subsection V-F. If k is not a divisor of n, one first uses Proposition 3 to reduce the problem of constructing an (n, k) MEBC code to that of constructing an (N, K) MEBC code, where $N = k$ and where K is the remainder when n is divided by k. One interates this procedure until K is a divisor of N. For $N = qK$, the $K \times qK$ matrix $G = [I_K \ : \ I_K \ : \ \cdots \ : \ I_K]$ is trivially the systematic generator matrix of an (N, K) MEBC code.

Remark: Propositions 1 through 3 hold also for matrices over any field, in particular over the finite field GF(Q) when Q is a prime power. However, every cyclic code over GF(Q) is automatically an MEBC code. Thus, the construction of MEBC codes given here would appear to be of interest, in the case were Q is a prime power, only when the parameters n and k are such that no cyclic code with these parameters exists.

References

[1] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Mathematics, Statistics, and Probability*, vol. 1, pp. 611–644, 1961.

[2] J. L. Massey, "Capacity and coding for the collision channel without feedback," presented at the Information Theory Conference, Oberwolfach, Germany, Apr. 4–10, 1982.

[3] K. A. Post, "Convexity of the non-achievable rate region for the collision channel without feedback," this issue.

[4] A. D. Wyner, "The wire-tap channel," *Bell Sys. Tech. J.*, vol. 54, pp. 1355–1387, Oct. 1975.

[5] T. M. Cover, "Broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 2–14, Jan. 1972.

[6] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservations," in *Proc. Nat. Computer Conf.*, pp. 711–716, 1973.

[7] N. Abramson, "The ALOHA system—Another alternative for computer communications," in *AFIPS Conf. Proc.*, Fall Joint Computer Conf., vol. 37, pp. 281–285, 1970.

[8] N. Abramson, "Packet switching with satellites," in *AFIPS Conf. Proc.*, National Computer Conf., vol. 42, pp. 695–702, 1973.

[9] L. Kleinrock, *Queueing Systems, Vol. II: Computer Applications.* New York: Wiley, 1976.

[10] J. K. Wolf, "Adding two information symbols to certain nonbinary BCH codes and some applications," *Bell Sys. Tech. J.*, vol. 48, pp. 2405–2424, 1969.

[11] B. S. Tsybakov and V. A. Mikhailov, "Ergodicity of a synchronous ALOHA system," *Probl. Peredachi Inform.*, vol. 15, no. 4, pp. 73–87, Oct.–Dec., 1979.

[12] J. Huber and A. Shah, "Simple asynchronous multiplex system for unidirectional low-data-rate transmission," *IEEE Trans. Commun.*, vol. COM-23, pp. 675–679, June 1975.

[13] J. L. Massey, "The capacity of the collision channel without feedback," in *Abstracts of Papers, IEEE Int. Symp. Information Theory*, Les Arcs, France, June 21–25, 1982, p. 101.

[14] B. S. Tsybakov and N. B. Likhanov, "Packet switching in a channel without feedback," *Probl. Peredachi Inform.*, vol. 19, no. 2, pp. 69–84, Apr.–June, 1983.

[15] D. L. Cohn, private communication, Apr. 1982.