LIDS-P-1406



# Multiple Accessing for the Collision Channel Without Feedback

JOSEPH Y. N. HUI, MEMBER, IEEE

*Abstract*—We consider asynchronous multiple accessing without feedback over the collision channel. Redundant coding is used to overcome user interference which causes erasures for collided packets. The channel has a sum capacity of $e^{-1}$ and a sum cutoff rate of 0.295. The best codes are long constraint length rate 1/3 convolutional encoders which achieve a sum throughput up to the sum cutoff rate using an easy-to-implement forward search decoding algorithm. In the presence of additive Gaussian noise, the same redundant coding can save, at hardly any extra cost, about 6 dB of signal-to-noise ratio.

## I. Introduction

MASSEY [1] has shown that the asynchronous collision channel, which is similar to the ALOHA channel [2] except without feedback, has the same $e^{-1}$ sum capacity as the ALOHA channel. This sum capacity is achievable whether the packets are slotted or not. However, the techniques used to prove the $e^{-1}$ sum capacity are difficult to use in practice because the set of simultaneous users can be large and varying. This paper proposes practical and novel signaling and decoding schemes for the nonfeedback collision channel which achieve very low bit error rates with small decoding complexity. The maximum sum throughput achieved, however, is upper bounded by the sum cutoff rate rather than the sum capacity of the channel. We believe that these schemes are significant for mobile radio applications for the following reasons. First, the transmitters are not required to transmit synchronously; thus, there is no need for scheduling. Second, there is no need for assigning different codes to different users as in spread spectrum multiple accessing [3]; each user identifies his packets by checking the preamble of the packets. Third, the channel degrades gracefully as the number of simultaneous users gets large. Fourth, any modulation technique may be used for these schemes, thus standardized modems are not required.

For the collision channel, each user sends packets which are destroyed if they overlap. We consider the use of such a channel for multiple accessing when each user redundantly encodes the information to be sent. The resulting data sequence is blocked into packets for transmission. The receiver collects all the packets from the corresponding transmitter and ignores all other packets and collisions. These packets are subsequently decoded. Section II describes the channel model and derives the capacity region. Section III analyzes the performance of block codes. Section IV proposes a convolutional coding and interleaving scheme, and computes the sum cutoff rate of the channel. Section V generalizes Sections II and IV to the case when the channel is corrupted by additive Gaussian noise; it shows that substantial power saving is gained, at hardly any extra cost, by the redundant coding that was originally intended to correct erasures due to packet collisions.

Manuscript received July 28, 1983; revised February 8, 1984. This work was supported by the National Science Foundation under Grant NSF/ECS 79-19880 and the Advanced Research Project Agency under Grant ONR/N00014-75-C1183.

The author is with Bell Communications Research, Murray Hill, NJ 07974.

## II. Modeling and Channel Capacity

The slotted (or packet synchronized) noiseless collision channel is defined as follows. The code symbol alphabet is $X_i = \{0,1,2,\cdots,2^n\}$ for user $i$ $(1 \leqslant i \leqslant M)$, in which 0 represents an idle and the numbers $1-2^n$ each represent a packet of length $n$ bits. The channel output is given by $y = x_j$ if $x_i = 0$ for all $i \neq j$, and $y =$ erasure if more than one $x_i \neq 0$. The channel transition diagram for user $i$ is shown in Fig. 1.

Without code synchronization and treating the signals of the other users as memoryless noise [4], reliable communication for each user at rate $R_i$ can be achieved if and only if for some

$$P(x_1 \cdots x_M y) = P_1(x_1) \cdots P_M(x_M) P(y/x_1 \cdots x_M)$$

we have

$$R_i \leqslant I(X_i; Y)$$

for all $i$. The capacity region is defined as the set of $(R_1, \cdots, R_M)$ that satisfies the above conditions. Due to the symmetry of the numbers $1-2^n$, it is apparent that each of the $2^n$ nonzero packets should be equiprobable for achieving capacity. Therefore, we shall abbreviate $p_{i,o}$ in Fig. 1 by $p_i$, and $p_{i,j}$ by $(1-p_i)/2^n$ for all $1 \leqslant j \leqslant 2^n$. It can be shown after some tedious evaluation that

$$I(X_i; Y) = n(1-p_i)/p_i \prod_{j=1}^{M} p_j + 0(1) \qquad \text{bits/slot}$$

$$\cong (1-p_i)/p_i \prod_{j=1}^{M} p_j \qquad n-\text{bits/slot}$$

for large $n$. We would like to have a characterization of the capacity region without the $p_i$'s. Consider the equations

$$R_i = (1-p_i)/p_i \prod_{j=1}^{M} p_j$$

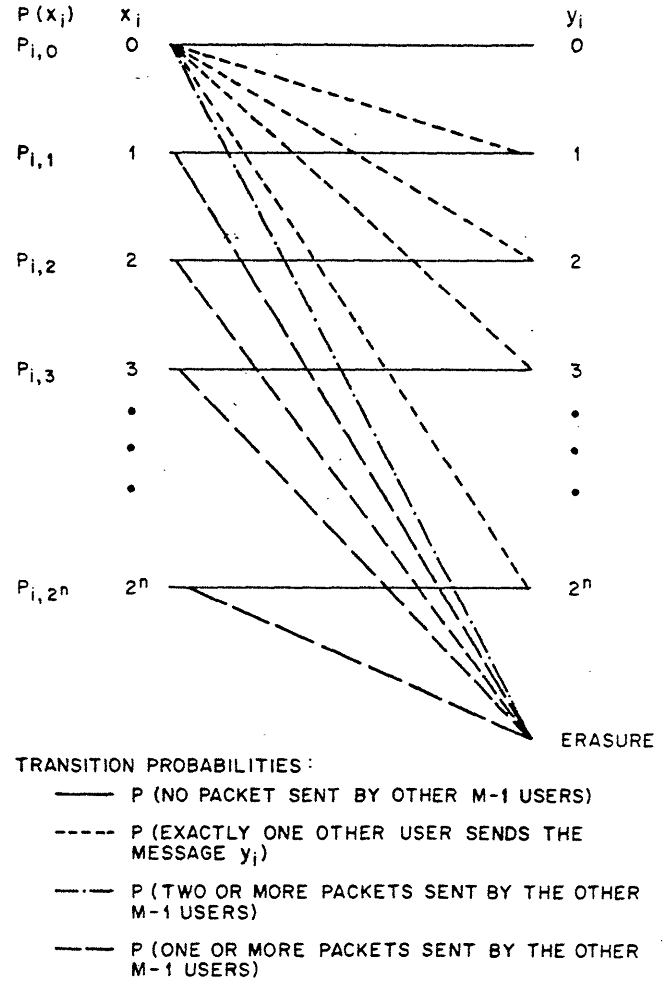


Fig. 1. Channel transition diagram for the collision channel.

for all $i$. Differentiating the $R_i$'s with respect to the $p_i$'s gives

$$dR_i = \sum_{\substack{k=1\\k\neq i}}^{M} \left\{(1-p_i)/(p_i p_k)\prod_{j=1}^{M} p_j\right\} dp_k - \left(\prod_{j=1}^{M} p_j\right) \Big/ p_i\, dp_i$$

which in matrix form can be expressed as

$$\begin{bmatrix} dR_1 \\ dR_2 \\ \cdot \\ \cdot \\ dR_M \end{bmatrix} = \begin{bmatrix} 1/p_1 & 0 & \cdots & 0 \\ 0 & 1/p_2 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 1/p_M \end{bmatrix} \cdot \begin{bmatrix} -p_1 & 1-p_1 & 1-p_1 & \cdots & 1-p_1 \\ 1-p_2 & -p_2 & 1-p_2 & \cdots & 1-p_2 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1-p_M & 1-p_M & 1-p_M & \cdots & -p_M \end{bmatrix} \cdot \begin{bmatrix} 1/p_1 & 0 & \cdots & 0 \\ 0 & 1/p_2 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 1/p_M \end{bmatrix} \begin{bmatrix} dp_1 \\ dp_2 \\ \cdot \\ \cdot \\ dp_M \end{bmatrix}.$$

At the boundary of the capacity region, it is necessary that the vector $(dR_1, dR_2, \cdots, dR_M)$ be tangential to the surface of the boundary for all vectors $(dp_1, dp_2, \cdots, dp_M)$. Therefore, it is necessary that the second matrix on the right-hand side be singular, which implies that

$$q_1 + q_2 + \cdots + q_M = 1$$

where $q_i = 1 - p_i$. Thus, the sum of the probabilities of sending packets for the $M$ users equals one at the boundary of the capacity region. Therefore, the boundary of the capacity region satisfies the following set of equations:

$$\sum_{i=1}^{M} (1-p_i) = 1$$

$$R_i = (1-p_i)/p_i \prod_{j=1}^{M} p_j, \qquad 1 \leqslant i \leqslant M.$$

For $M=2$, eliminating $p_1$ and $p_2$ from the above equations gives

$$\sqrt{R_1} + \sqrt{R_2} = 1.$$

It is very difficult to solve the above equations in useful forms for $M > 2$. Instead, we shall obtain the sum of the rates along the main diagonal (by having equal $p_i$'s). The sum has a maximum value of $(1-1/M)^{M-1}$. For large $M$, this value converges to $e^{-1}$, which is the same as the capacity of the slotted ALOHA channel.

The remainder of this section examines how unslotted packets affect the capacity of the channel. Let $1-p_i$ be the probability that user $i$ transmits a packet in a slot. The probability (conditioned on a transmission in a slot) that a packet for user $i$ does not collide with the packets of user $j$ is $p_j^2$ because the packet for user $i$ may collide with a packet in the two slots that overlap with the packet of user $i$. Hence, the throughput for user $i$ is given by

$$T_i = (1-p_i)/p_i^2 \prod_{j=1}^{M} p_j^2.$$

Assume equal rates for all users so that $p_i = p_j = p$; the sum of the rates is then

$$T = \sum_{i=1}^{M} T_i = M(1-p)p^{2(M-1)}$$

which is maximized when $p = 1 - 1/(2M-1)$, with

$$T = [M/(2M-1)](1-1/(2M-1))^{2(M-1)}.$$

For large $M$, $T$ approaches $e^{-1}/2$, which is the same as the capacity for the pure ALOHA scheme.

In practice, we may be able to recover the front part of a packet up to the point when it starts to collide with another packet, as shown in Fig. 2. No portion of a packet which starts at a time when another packet is being transmitted may be recovered since the preamble (for identification and receiver synchronization), which is placed at the beginning of the packet, is lost in the collision. We assume

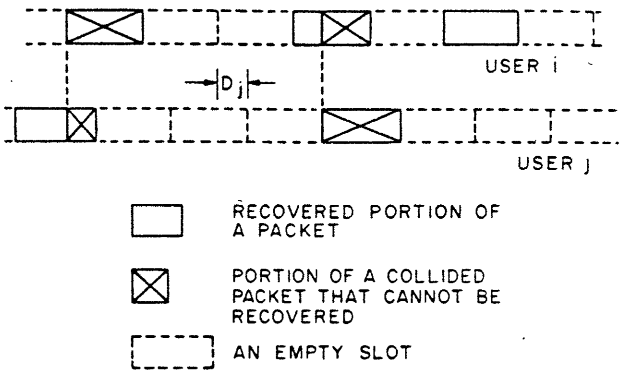


Fig. 2. Partially recoverable packets.

the length of the preamble to be small compared to the length of the packet. Appendix A shows that the maximum sum of the throughput equals 1/4.

Massey [1] has shown that the capacity of the unslotted channel is the same as that of the slotted channel. Consider the grouping of $u$ packets into a superpacket as shown in Fig. 3. The key idea is that even though a packet is totally lost through partial overlapping with other packets, part of a superpacket can be retrieved when it overlaps partially with other superpackets, as shown in Fig. 3. The proof of the $e^{-1}$ sum throughput is shown in Appendix B.

## III. Block Coding Scheme

This section will examine the use of block codes for the slotted collision channel. At the end of the section we show how superpacketing may be used for the unslotted channel.

Reed–Solomon codes are especially effective for the correction of erasures. Each packet, which consists of $n$ bits, may be treated as an element of the Galois field $GF(2^n)$. A Reed–Solomon code, defined on $GF(2^n)$, has codewords $f=(f_1,\cdots,f_k)$ satisfying the equations [for some $m$, $d$, and distinct $z_1, z_2, \cdots, z_k \in GF(2^n)$]

$$\begin{bmatrix} z_1^m & z_2^m & \cdots & z_k^m \\ z_1^{m+1} & z_2^{m+1} & \cdots & z_k^{m+1} \\ z_1^{m+2} & z_2^{m+2} & \cdots & z_k^{m+2} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ z_1^{m+d-2} & z_2^{m+d-2} & \cdots & z_k^{m+d-2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \cdot \\ \cdot \\ f_k \end{bmatrix} = \underline{0}_{d-1}.$$

Consequently, the Reed–Solomon code has code rate $(n-d+1)/n$. Define the distance between two codewords $f=(f_1,\cdots,f_k)$ and $f'=(f_k',\cdots,f_k')$ as the total number of places where $f_i \neq f_i'$. It is well known that this Reed–Solomon code has a minimum distance of at least $d$.

Massey [1] used the Reed–Solomon code to prove that zero error probability can be achieved with each user transmitting at an information rate of $(1-1/M)^{M-1}$ $n$-bit/slot. The scheme involves a clever way for each user to put $M^{M-1}$ packets in a time frame of $M^M$ slots, such that exactly $(M-1)^{M-1}$ packets would be collision free for each user, no matter how the frames of the users are relatively shifted (slotwise). Therefore, a Reed–Solomon

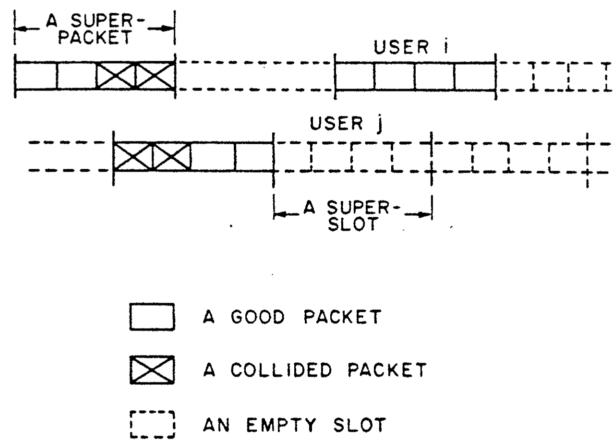


Fig. 3. Superpackets for achieving the $e^{-1}$ capacity.

code of rate $(M-1)^{M-1}/M^{M-1}$ would suffice to correct all the erasures. The throughput for each user is then $(M-1)^{M-1}/M^M=(1-1/M)^{M-1}/M$. The sum throughput is $(1-1/M)^{M-1}$, which is the same as the capacity derived in Section II.

The scheme of Massey is inadequate for implementation if the number of users ($M$) is large or variable. However, Reed–Solomon codes are still effective for such a channel. Assume that a user puts a packet in a slot with probability $q=1-p$, and the $k$ consecutive packets he puts on the channel comprise a codeword for a rate $r=h/k$ Reed–Solomon code. Consequently, the information rate for the user is $q\cdot h/k$ $n$-bits/slot. The sum throughput is $T=Mqr$ $n$-bits/slot. The probability of erasure for a packet is

$$\epsilon = 1-(1-q)^{M-1} = 1-(1-T/Mr)^{M-1} \approx 1-e^{-T/r}$$

for large $M$. The probability of block error is upper bounded by

$$P_b < \sum_{i=h}^{k} {}_kC_i \epsilon^i (1-\epsilon)^{k-i}.$$

Using an upper bound [5] on the sum of the tail of a binominal distribution, we have

$$\begin{aligned} P_b &\leqslant h(1-\epsilon)/\big(h(1-\epsilon)-(k-h)\epsilon\big)\, {}_kC_h \epsilon^h (1-\epsilon)^{k-h} \\ &\leqslant r(1-\epsilon)/\big(r(1-\epsilon)-(1-r)\epsilon\big)\, e^{kH_e(r)}\epsilon^h(1-\epsilon)^{k-h} \\ &= r(1-e^{-T/r})/\big(re^{-T/r}-(1-r)(1-e^{-T/r})\big) \\ &\quad \cdot \Big[r^{-r}(1-r)^{-(1-r)}(1-e^{-T/r})^r e^{-(T/r)(1-r)}\Big]^k. \end{aligned}$$

This upper bound $P_b(T,r,k)$ is plotted as a function of $T$ for several codes in Fig. 4.

If the channel is unslotted, $u$ Reed–Solomon code symbols may be grouped together to form a superpacket. For large $u$, the probability of erasure for a particular packet is the same as that of the unslotted channel. Thus, the throughput of the unslotted channel is the same as the $e^{-1}$ throughput of the slotted channel.

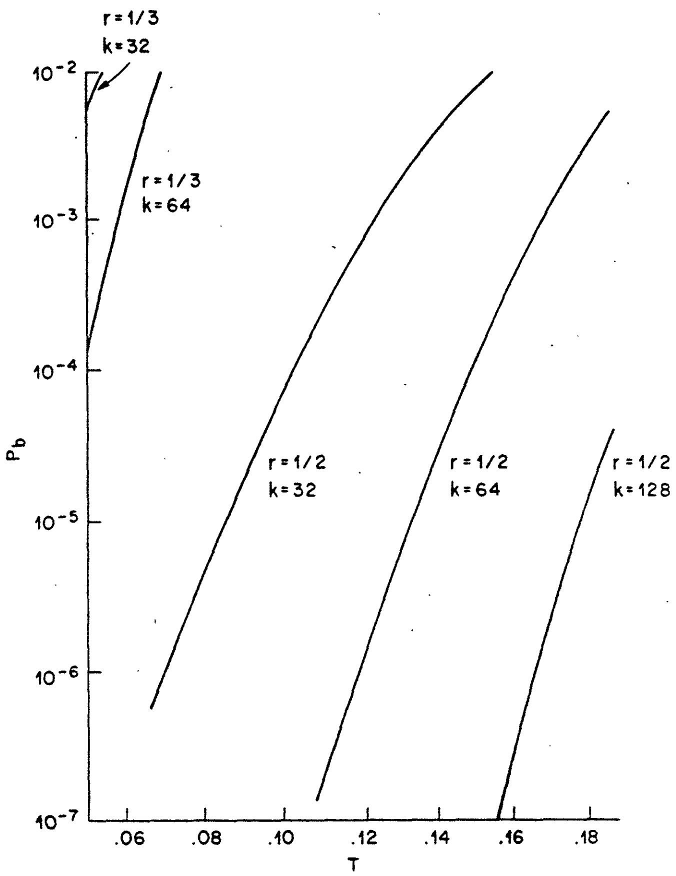


Fig. 4. Upper bound on block error probability versus throughput.

## IV. Convolutional Coding Scheme

We now turn to the use of convolutional codes. The encoding scheme is shown in Fig. 5. The outputs of the rate $r = 1/v$ constraint length $k$ code ($v = 2$ in the figure) is fetched alternately by the switch $S_1$. Switch $S_2$ puts successive bits into successive packets in a stack of $m$ packets. When all $m$ packets are filled, each packet would be transmitted after a random delay.

For the unslotted channel, the data sequence is fed in parallel (Fig. 6) to $u$ encoder-interleavers (the square box in Fig. 5), and the $u$ packets from the encoder-interleavers are grouped together to form a superpacket. The superpacket is transmitted after a random delay.

Decoding delay results from the fact that the data bits have to be deinterleaved before decoding. This delay is of the order $mn$. To minimize delay, we may use a smaller stack or shorten the length of a packet. Shortening the length of a packet, however, would incur a higher fraction of preamble overhead in the message body. The number of packets in the stack should be at least $kv$ so that each bit in a packet is generated from a different set of information bits. If $m < kv$, error probability would increase because erasures are clustered together. Choosing $m$ slightly larger than $kv$ would suffice since the likelihood of long error paths in the trellis is small.

Decoding delay due to deinterleaving can be reduced by the use of convolutional interleaving as shown in Fig. 7.

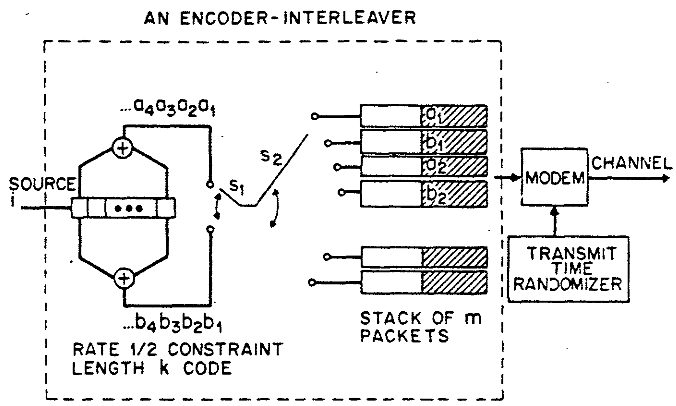


Fig. 5. Multiple access scheme for convolutional codes.

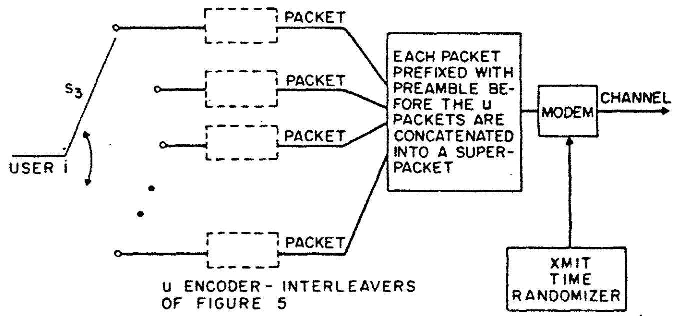


Fig. 6. Formation of superpackets.

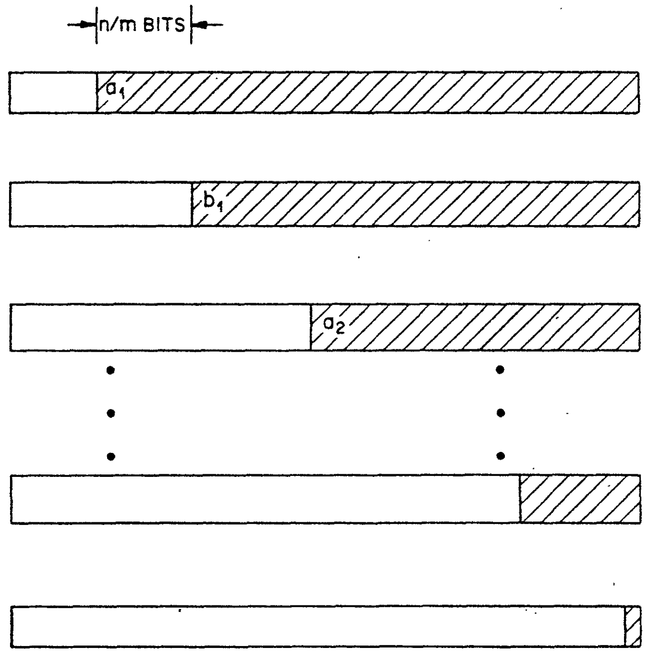


Fig. 7. Convolutional interleaving.

Each packet is filled with $n/m$ more bits than the next packet in the stack. The packet at the top of the stack is popped off the stack when the packet is filled. An empty packet is added to the bottom of the stack when the packet at the top is popped. The popped packet is transmitted after a random delay, which is smaller than the packet interarrival time. Using this scheme, the delay incurred and the buffer required are about half that of the previous interleaving scheme. The convolutional interleaving scheme,

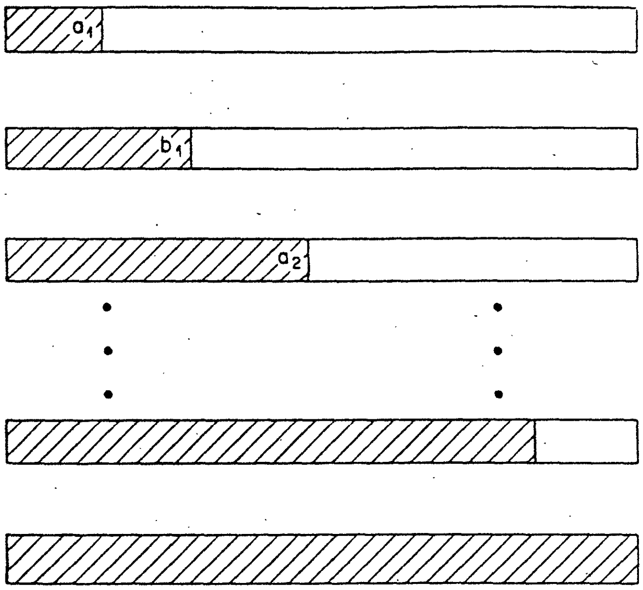


Fig. 8. Length of the first few packets of a message.

however, has a problem with the first few packets of a message since the packet at the top of the stack is empty at the beginning. It seems that the first few packets would have to be filled in the manner shown in Fig. 8, subsequently wasting $m/2$ packets per message. This waste is tolerable only if the message is long. The choice between these two schemes depends on the length of the message and the amount of decoding delay that can be tolerated.

We now give a random coding bound, based on the cutoff rate, for the error probability. Consider the deinterleaved data sequence at the receiver. The occurrence of erasures in the deinterleaved sequence is not memoryless in the sense that erasures, if they occur, are periodic. If we use a decoding metric that is time invariant and a convolutional encoder of long constraint length, these erasures may be viewed as being memoryless [4]. With lots of interleaving, we may treat the channel as a binary erasure channel shown in Fig. 9 if we impose the time-invariant structure on the decoder. The capacity of the binary erasure channel is equal to $C = 1 - \epsilon$. Consequently, the sum capacity is

$$C_T = \max_q Mq(1-\epsilon) = \max_q Mq(1-q)^{M-1} = (1-1/M)^{M-1}$$

which approaches $e^{-1}$ for large $M$. The cutoff rate for the binary erasure channel is

$$R_o = -\log_2\left(\sum_y\left(\sum_x P(x)\sqrt{P(y/x)}\right)^2\right) = -\log_2((1+\epsilon)/2) = -\log_2(1-e^{-Tv}/2)$$

since $\epsilon = 1 - e^{-Tv}$ for large $M$.

The sum of the cutoff rates for the $M$ users is

$$R_T = MqR_o = -Tv\log_2(1-e^{-Tv}/2)$$

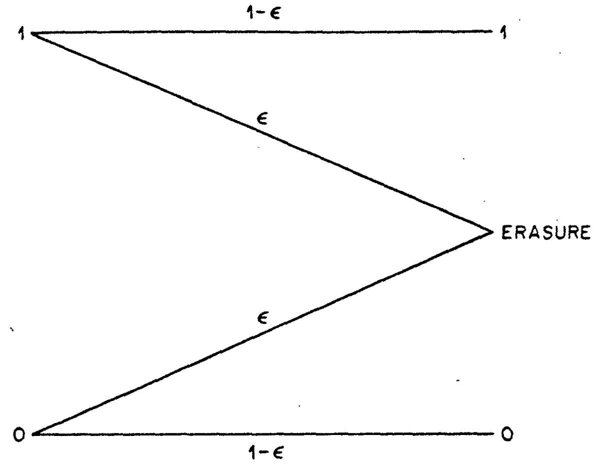


Fig. 9. The binary erasure channel.

since $Mq/v = T$. Reliable communication using sequential decoding is achievable for $T$ up to $R_T$. Consequently, the maximum sum throughput satisfies

$$T = -Tv\log_2(1-e^{-Tv}/2).$$

Solving for $T$, we have

$$T = -1/v\ \ln 2(1-2^{-1/v}). \tag{1}$$

$T$ has a minimum value of 0.295096 when $v = 3.014$. For practical purposes, $v$ is an integer so the best code rate is 1/3, with $T = 0.295093$. For $v = 2$, the maximum $T$ is 0.2674.

Errors in decoding can arise when an incorrect path with a higher metric merges into the correct path. The ensemble average of the expected number of bit errors for such an error event starting at a given time is upper bounded by [6]

$$P_b < 2^{-vR_ok}/[1-2^{-(vR_o-1)}]^2 = (1-e^{-Tv}/2)^{vk}/[1-2(1-e^{-Tv}/2)^v]^2.$$

This bound is plotted in Fig. 10 for several code rates and constraint lengths. The plot shows that a long constraint length should be used to combat the severe user interference. Such long constraint lengths make Viterbi decoding impractical. Thus, the decoder should track only a subset of states, the size of which is independent of the constraint length.

One decoding scheme works as follows. The decoder keeps a list of the plausible states of the trellis. A state is plausible if there is a path leading up to the state such that the code sequence of the path is agreeable with the deinterleaved channel sequence. The code symbol zero (or one) is disagreeable with the channel symbol one (or zero) while the channel symbol erasure is agreeable with the code symbols zero or one. The complexity of decoding is measured by the random variable $n$, the number of plausible states in steady state, with probability distribution $P(n)$. There is some preliminary evidence that this complexity is small [4].

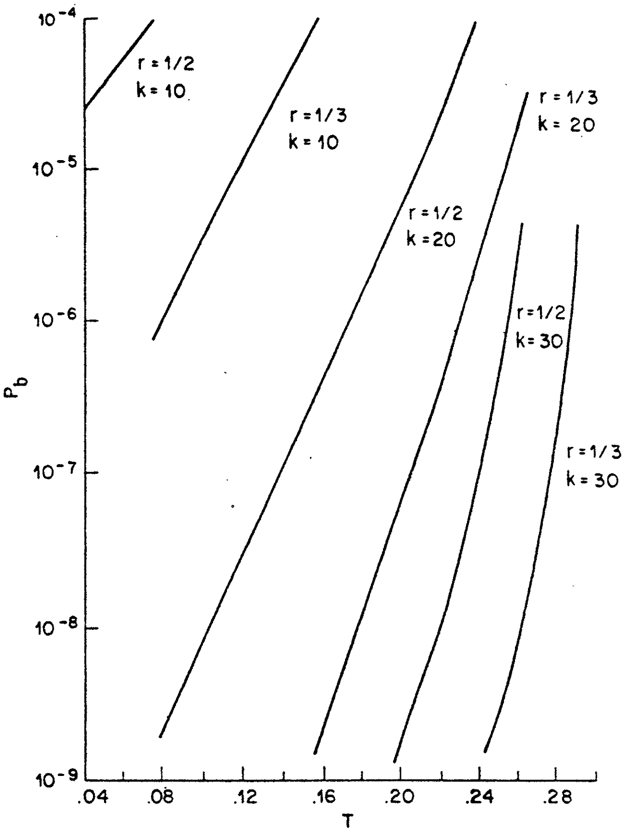


Fig. 10. Upper bound on bit error probability for convolutional codes.

## V. The Collision Channel with Additive Gaussian Noise

This section shows that the coding that was originally intended for correcting erasures is effective for combating additive Gaussian channel noise. Therefore, signaling power can be reduced compared to the case of no coding, at hardly any extra cost.

We shall assume that antipodal signaling is used. The results obtained in this section can be extended easily to other modulation techniques. The channel can be modeled as the memoryless binary input, Gaussian or erasure output channel shown in Fig. 11(a). The capacity of this channel is

$$C = (1-\epsilon) \quad \text{of the capacity of the channel in Fig. 11(b)}$$

$$= (1-\epsilon)\left\{-1/2 \log_2(2\pi e) - \int_{-\infty}^{\infty} P(y) \log_2 P(y)\, dy\right\}$$

where

$$P(y) = (P_1(y) + P_{-1}(y))/2$$

and

$$P_l(y) = \exp\left[-\left(y - l(2E_s/N_o)^{1/2}\right)^2/2\right]/\sqrt{2\pi}.$$

The cutoff rate can be shown to be

$$R_o = -\log_2\left[\epsilon + (1-\epsilon)(1+e^{-E_s/N_o})/2\right]. \quad (2)$$

Since sequential decoders are used for decoding, we concentrate on the study of the cutoff rate. Fig. 12 gives a plot of $R_o$ as a function of $E_s/N_o$ for several values of $\epsilon$. The value of $R_o$ is seen to saturate at

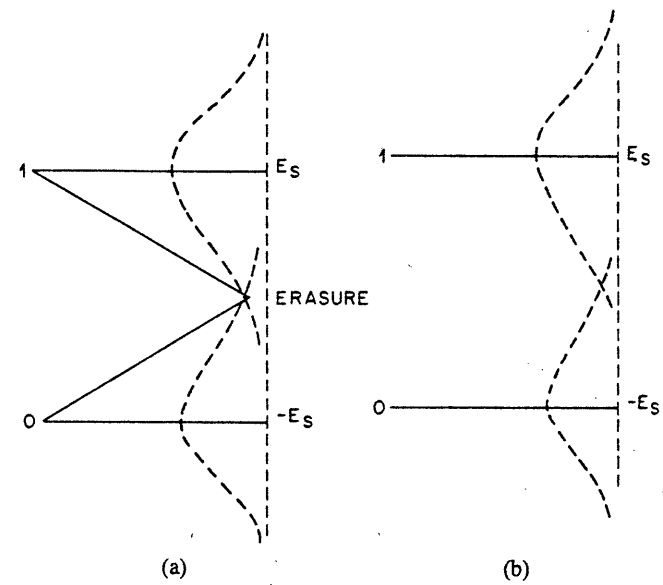


Fig. 11. (a) The binary input, erasure or Gaussian output channel. (b) The binary input, Gaussian output channel.

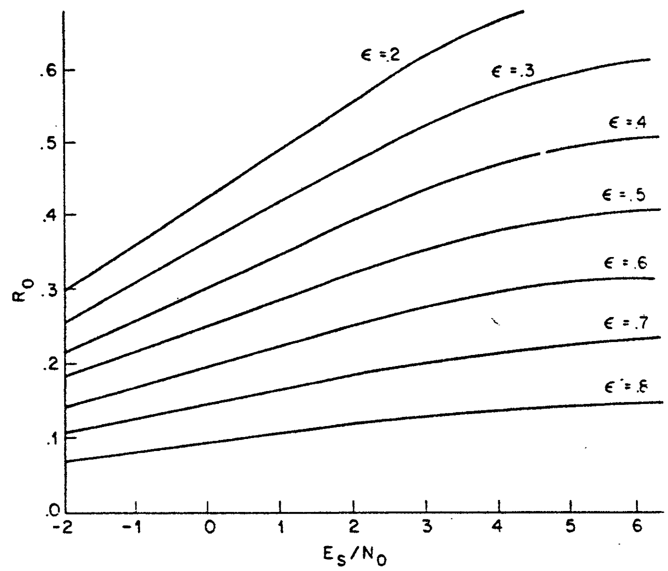


Fig. 12. Cutoff rate versus signal-to-noise ratio.

$$R_o^* = -\log_2((1+\epsilon)/2)$$

for large values of $E_s/N_o$. This occurs when the occurrence of erasures dominates the effect of the Gaussian noise of the channel.

Suppose we allow Gaussian noise to decrease $R_o$ from $R_o^*$ by a fraction $f$, such that

$$R_o = fR_o^* = -f \log_2(1 - e^{-Tv}/2). \quad (3)$$

Consequently, the maximum sum throughput achievable by sequential decoding is [similar to the derivation of $T$ in (1)]

$$T = -1/v \ln 2(1 - 2^{-1/vf}). \quad (4)$$

This throughput as a function of $f$ is shown in Fig. 13 for $v = 2,3$.

What remains to be found is the value of the minimum $E_s/N_o$ that is required in order to achieve the maximum value of $T$ for fixed $v$ and $f$. Eliminating $R_o$ and $\epsilon$ in (2) by the substitutions

$$R_o = -f \log_2 (1 - e^{-Tv}/2)$$
$$\epsilon = 1 - e^{-Tv}$$

with $T$ given in (4), we have after some simplification

$$E_s/N_o = -\ln\left[(2^{-1/v} - 2^{-1/vf})/(1 - 2^{-1/vf})\right]$$

which is shown in Fig. 14 for values of $v$ and $f$. The figure shows that 4 or 5 dB of signal-to-noise ratio is sufficient for operation. This compares favorably to the typical 10.5 dB required for uncoded antipodal signaling for a bit error rate of $10^{-6}$.

## VI. Conclusion

We have demonstrated the viability of the collision channel for mobile radio application where there is no feedback and scheduling, and transmitting power is limited. Total throughput approaches 30 percent. Further work is required to characterize the decoding complexity, delay, and performance in a cellular environment.

## Appendix A

This Appendix derives the sum throughput of the unslotted collision channel for which we may recover the front part of a packet up to the point where it starts to collide with another packet. Let $1 - p$ be the probability that a user puts a packet in a slot. The probability that a packet is collision free is $p^{2(M-1)}$. The probability that the packet is totally lost is $1 - p^{M-1}$. Let $D_j$ be the difference in the starting time between user $i$ and user $j$, as shown in Fig. 2. The probability that at least a fraction $f$ of a packet of user $i$ is recovered is

$$P_f = \prod_{\substack{j=1 \\ j \neq 1}}^{M} \Big[ P(D_j \leq f) P(\text{no packet for user } j \text{ in the two slots that overlap with the packet of user } i) + P(D_j > f) P(\text{no packet for user } j \text{ in the first slot of the two slots that overlap with the packet of user } i) \Big]$$
$$= \left[ fp^2 + (1-f)p \right]^{M-1}.$$

Hence

$$P(f) = -dP_f/df$$
$$= (M-1)(1-p)p^{M-1}(1 - f(1-p))^{M-2}.$$

| f | v = 2 | v = 3 |
|---|---|---|
| 0.40 | -- | .044 |
| 0.45 | -- | .073 |
| 0.50 | -- | .100 |
| 0.55 | .034 | .126 |
| 0.60 | .065 | .149 |
| 0.65 | .095 | .171 |
| 0.70 | .124 | .192 |
| 0.75 | .150 | .211 |
| 0.80 | .176 | .230 |
| 0.85 | .200 | .247 |
| 0.90 | .224 | .264 |
| 0.95 | .246 | .280 |
| 1.00 | .267 | .295 |

Fig. 13. Maximum throughput $T$ for given $f$ and $v$.

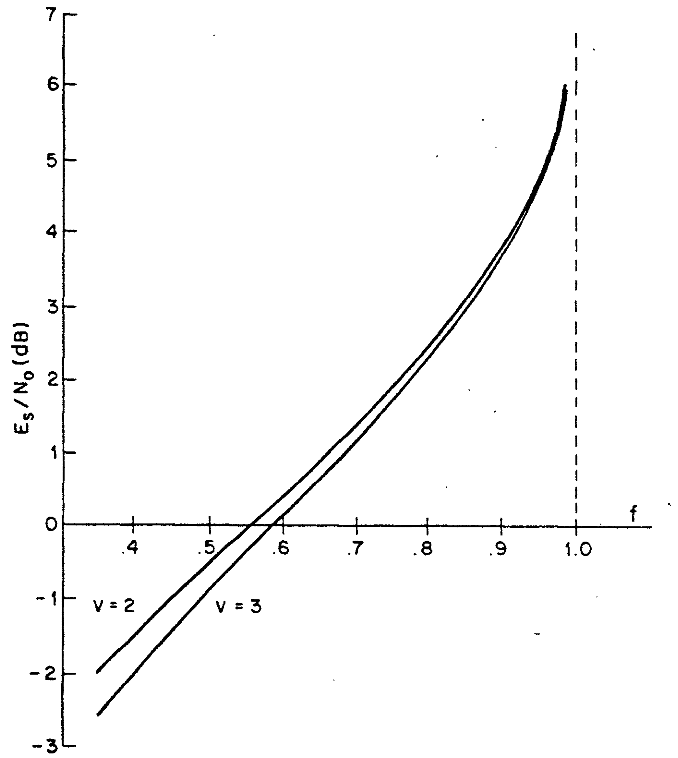


Fig. 14. Power required to achieve maximum $T$ for given $f$ and $v$.

Therefore, the average fraction of a packet that is recovered is

$$\int_0^1 P(f) f df + 1.p^{2(M-1)}$$
$$= (1 - p^M)p^{M-1}/[(M-1)(1-p)].$$

The sum of the throughput for the $M$ users is

$$T = M(1 - p^M)p^{M-1}/(M-1)$$

which has a maximum value of

$$[M/(M-1)][M/(2M-1)][(M-1)/(2M-1)]^{(M-1)/M}$$

which occurs at

$$p = [(M-1)/(2M-1)]^{1/M}.$$

For large $M$, this throughout approaches 1/4.

## Appendix B

This Appendix derives the throughput of the unslotted collision channel with superpacketing. Let $q = 1 - p$ be the probability that each user puts a superpacket in a superslot. For user $i$

$P$(a packet is collision free with user $j$)

= $P$(a superslot for user $j$ does not start within the duration of the packet).

$P$(no superpacket in the superslot of user $j$ that overlaps with the packet)

+ $P$(a superslot for user $j$ starts within the duration of the packet).

$P$(no superpacket in either superslots of user $j$ that overlap with the packet)

$= [(u-1)/u]\, p + [1/u]\, p^2.$

Therefore, a packet is successfully transmitted with probability

$$\{[(u-1)/u]\, p + [1/u]\, p^2\}^{M-1}$$

and the sum throughput is

$$T = M(1-p)\{[(u-1)/u]\, p + [1/u]\, p^2\}^{M-1}.$$

Letting $q = f/M$ and assuming a large value of $M$ give the value $T$ of

$$\begin{aligned} &f\{[(u-1)/u](1-f/M)+[1/u](1-f/M)^2\}^{M-1} \\ &\quad \cong f\{1-1/M\,([(u-1)/u]\,f+[2/u]\,f)\}^{M-1} \\ &\quad = f\{1-1/M\,([(u+1)/u]\,f)\}^{M-1} \\ &\quad = fe^{-f(u+1)/u} \end{aligned}$$

which has a maximum value of $e^{-1}u/(u+1)$ when $f = u/(u+1)$. Thus, the capacity approaches $e^{-1}$ for large $u$, which is the same as the capacity for the slotted channel.

## Acknowledgment

The author would like to thank Prof. P. A. Humblet for his supervision of the work for the two papers in this issue.

## References

[1] J. L. Massey, "The capacity of the collision channel without feedback," private communication.
[2] N. Abramson, "The Aloha system—Another alternative for computer communications," in *Fall Joint Comput. Conf. AFIPS Conf. Proc., vol. 37*. Montvale, NJ: AFIPS Press, 1970.
[3] J. Y. N. Hui, "Throughput analysis for code division multiple accessing of the spread spectrum channel," this issue, pp. 482–486.
[4] —, "Fundamental issues of multiple accessing," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1983.
[5] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[6] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

**Joseph Y. N. Hui** (S'82–M'83), for a photograph and biography, see this issue, p. 486.