TIMESHARING WITHOUT SYNCHRONIZATION*

ROBERT J. McELIECE and ARTHUR L. RUBIN
Caltech's Jet Propulsion Laboratory

Summary. The capacity region of a multiple-access channel has recently been identified as the convex hull $\bar{K}$ of a certain set K of points in the first quadrant of the (R_1,R_2) plane. For a pair of rates in K, a more or less standard random coding argument can be used to show the existence of a good pair of codes. But for points in $\bar{K}$-K, it is apparently necessary for the two senders to use some form of timesharing to achieve the desired rates. However, in order to timeshare, at least one of the senders must have knowledge of the other's phase; and in many practical situations this knowledge does not exist. In this paper we investigate the problems which arise in coding for multiple access channels when the senders cannot synchronize with each other.

Introduction. We begin with a brief summary of the main results about (discrete, memoryless) multiple-access channels. A multiple access channel has two inputs $x^{(1)} \epsilon A_1$, $x^{(2)} \epsilon A_2$, and one output $y \epsilon B$, where A_1, A_2, and B are finite sets. The transition probability $p(y|x^{(1)}x^{(2)})$represents the probability that y will be the output, given that $x^{(1)}$ and $x^{(2)}$ are the two inputs. The appropriate block diagram for using this channel:

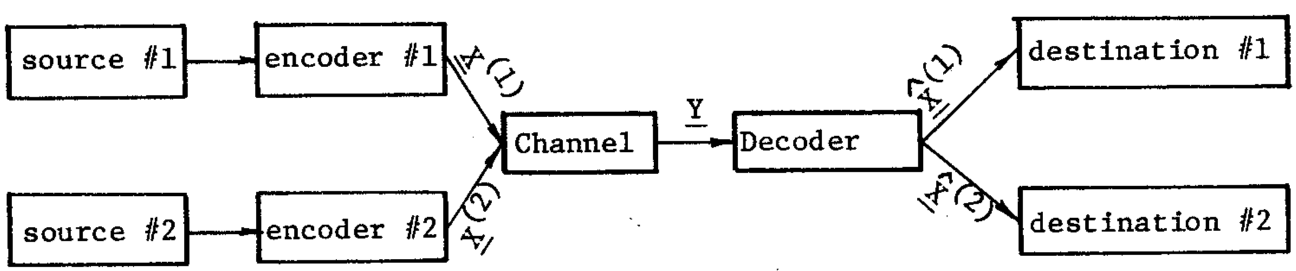

Figure 1. Multiple-access communication system

The sources are assumed to be independent; note that since the channel has only one output, there is no loss of generality in postulating a single decoder, although there may actually be two receivers located in physically different places. Given a pair of codes for this situation , $C_1 = \{\underline{x}_1^{(1)}, \ldots, \underline{x}_M^{(1)}\} \subseteq A_1^n$, $C_2 = \{\underline{x}_1^{(2)}, \ldots, \underline{x}_{M_2}^{(2)}\} \subseteq A_2^n$ and a decoding rule $\Delta: B^n \to C_1 x C_2$, the rates are $R_i = \frac{1}{n}\log_2 M_i$, and the error probability is $P_E = P\{\hat{\underline{X}}^{(1)} \neq \underline{X}^{(1)}$ or $\hat{\underline{X}}^{(2)} \neq \underline{X}^{(2)}\}$ (this assumes each of the codewords is equally

*This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract N. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

likely to be sent.) A pair of rates (R_1,R_2) is said to be achievable, if for each $\epsilon > 0$ there exist a pair of codes of rates $\geq R_1,-\epsilon, R_2-\epsilon$ with $P_E < \epsilon$. The set of achievable pairs is called the capacity region and it is described as follows.

For each pair of independent probability assignments p_1 and P_2 on A_1 and A_2, define the following figure:

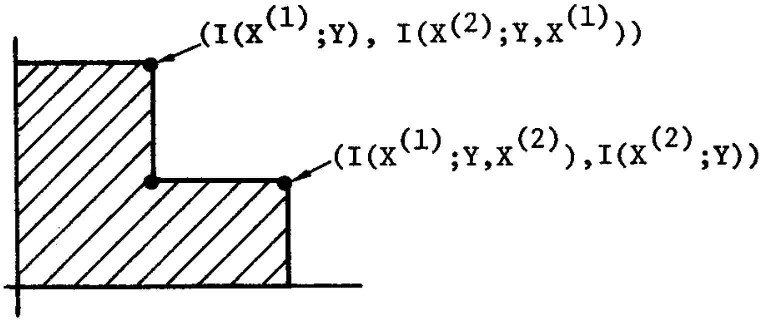

Figure 2.
The Region $K(p_1,p_2)$.

Here the random variables $X^{(1)},X^{(2)}$, and Y represent the two inputs and the output when the input distribution is $P\{X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}\} = p_1(x^{(1)})p_2(x^{(2)})$. The capacity region is the convex hull $\overline{K}$ of $K = \cup K(p_1,p_2)$ where the union is taken over all pairs p_1, p_2. (This result is attributed by Wyner [1] to Liao; Ah1swede; and Slepian,and Wolf.)

It is possible to show that every point in K is achievable by random coding, i.e. by computing the average of P_E over all pairs of codes chosen according to the probability assignments p_1 and P_2, etc. For points in $\overline{K}$- K, i.e. points that are convex combinations of points in K but not in K, it is apparently necessary to use a timesharing argument. By this we mean dividing the transmission time between a strategy which achieves (R_1,R_2) and one that achieves(R_1',R_2') in the ratio λ to $1-\lambda$ in order to achieve the rates $(\lambda R_1+(1-\lambda)R_1'$, λR72$+(1-\lambda)R_2')$. However, in order to timeshare, at least one of the senders must know when the other is using the first strategy and when he is using the second, i.e. same kind of synchronization between the senders must be possible. It is this assumption of synchronization we wish to abandon.

To be precise, for the rest of the paper we will investigate the multiple-access channel under the following assumption: The senders may agree ahead of time on the strategies they will use, but neither can know when the other will begin transmission. The receiver, however, can determine when each sender has begun transmission. (We assume the receiver gets his synchronization information either through separate sync channels, or by detecting synchronization prefixes used by the senders. The design of such prefixes presents some interesting problems we do not propose to address in this paper.)

2. Noiseless binary channels. In this section we shall illustrate the problems that arise in the simple case of noiseless channels, i.e. channels for which the output is uniquely determined by the inputs. If we disregard channels for which y depends only on $x^{(1)}$ or $x^{(2)}$

but not on both, there are up to obvious equivalence, only <u>five</u> such channels with $A_1 = A_2 = \{0,1\}$. Their behavior is described by the following table.

$x^{(1)}$	$x^{(2)}$	y: A	B	C	D	E
0	0	0	0	0	0	0
0	1	1	1	2	1	1
1	0	2	2	2	1	1
1	1	3	2	1	0	1

Table 1.
Five Noiseless Channels

The capacity regions (assuming synchronization; cf. section 1) are given below:

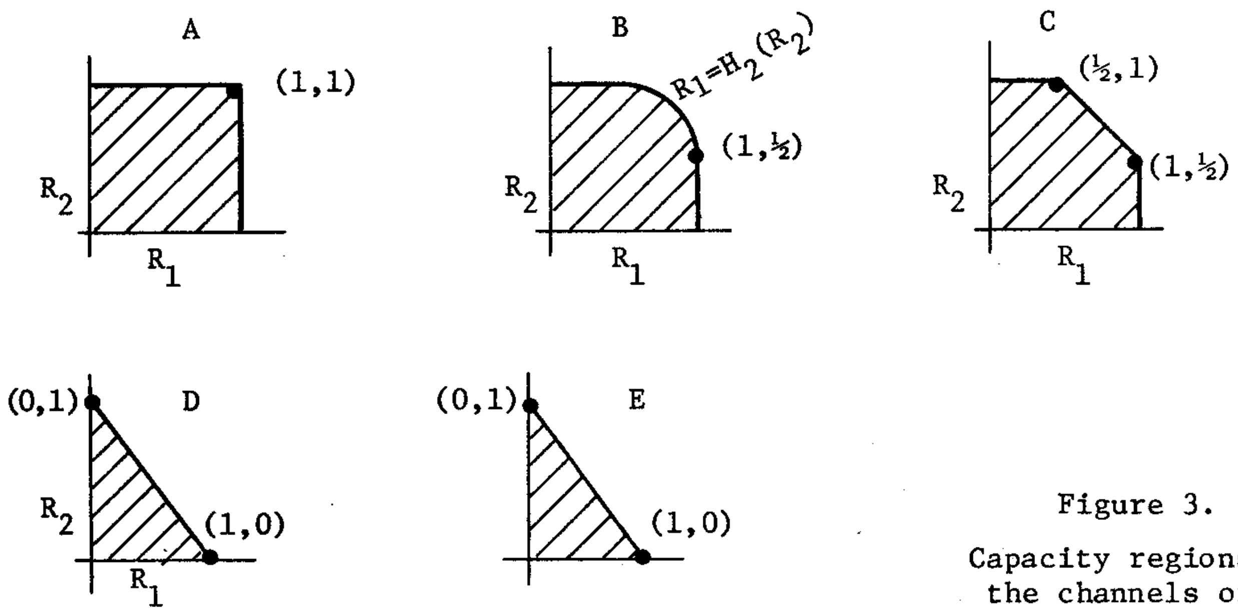

Figure 3.
Capacity regions of the channels of Table 1.

Channel A presents no difficulty, since the inputs do not interfere with each other; each sender can achieve rate 1 and $P_E = 0$ without coding, and this does not require synchronization.

Skipping channels B and C temporarily, we consider D. Here $Y \equiv x^{(1)}+x^{(2)}$ (mod 2). With synchronization, a pair of rates $R_1 = k/n$ and $R_2 = (n-k)/n$ can be achieved as follows:

$$\#1: \left[x_1^{(1)}, x_2^{(2)}, \ldots, x_k^{(1)}, 0, \quad 0, \ \ldots, \quad 0\right]$$
$$\#2: \left[0, \quad 0, \ \ldots, \quad 0, \quad x_1^{(2)}, x_2^{(2)}, \ldots, x_{n-k}^{(2)}\right].$$

Without synchronization, this won't work. However, if we assume that there exists an (n,k) binary cyclic code, we can still achieve rates R_1, and R_2: #1 encodes its bits into a codeword from the cyclic code, and #2 uses the same strategy as above.

$$\#1: \left[c_1, c_2, \ldots, c_k, c_{k+1}, \ldots, c_n\right]$$
$$\#2: \left[0,\ 0,\ \ldots, 0\ , x_1^{(2)}, \ldots, x_{n-k}^{(2)}\right].$$

Then whatever the relative phases of the two senders, out of every #1 codeword at least a block of k (cyclically) consecutive symbols will be received accurately, and so since the code is cyclic, it will be possible to reconstruct the entire codeword, and so also to reconstruct the k information bits $x_1^{(1)}, \ldots, x_k^{(1)}$. Once this is done the information bits $x_1^{(2)}, \ldots, x_{n-k}^{(2)}$ can be recovered. Finally, although (n,k) cyclic codes do not exist for all values of (n,k), the set {k/n: there exists an (n,k) cyclic code} is dense in [0,1], so that every point in the capacity region of figure 3 is achievable, without synchronization.

Channel E is more difficult. Here $y = x^{(1)}$ OR $x^{(2)}$, so if either input is 1, the other is completely lost. In this case we do not know how to achieve explicitly every point in the capacity region, but we can achieve every point that satisfies $\sqrt{R_1} + \sqrt{R_2} \leq 1$, as follows. Let A be an (n,k) cyclic code of rate $R_A = k/n$, and let B be a (m,j) cyclic code of rate $R_B = j/m$. Then #1 sends blocks of m interleaved codewords from A, the last j of which are all zeroes.

#2 sends n concatenated codewords from B, the last k of which are all zeroes. We illustrate for the simple case k = j = 1, m = n = 2, where both codes are the (2,1) repetition code:

$$\#1: \left[x^{(1)}\ 0 \quad x^{(1)}\ 0\right]$$
$$\#2: \left[x^{(2)}\ x^{(1)}\ 0 \quad 0\right]$$

It is easy to verify that whatever the relative phases of #1 and #2, the locations of the zeroes in one encoded stream allow the recovery of the other. Furthermore the rates are $R_1 = R_A(1-R_B)$, $R_2 = R_B(1-R_A)$. It is easy to verify that this implies $\sqrt{R_1} + \sqrt{R_2} \leq 1$, with equality iff $R_A + R_B = 1$. Since as we pointed out above the possible rates R_A and R_B are dense in [0,1], it follows that every pair (R_1,R_2) with $\sqrt{R_1} + \sqrt{R_2} \leq 1$, is achievable in this way. In particular, the example above yields the point (¼,¼)

We conclude this section with some brief remarks about channels B and C. In channel B, note that if #1 sends his data at $R_1 = 1$ uncoded, #2 sees a binary erasure channel with erasure probability ½. The capacity of this channel is ½, so #2 can achieve any rate

< ½ with $P_E < \epsilon$. Hence the point (1,½) is achievable without synchronization. Similarly the points (1,½) and (½,1) are achievable for channel C. In the next section we will see how to achieve some, but not all, of the remaining points in the capacity region.

3. A General Result. In this section we shall sketch a proof of the fact that even without synchronization every point in K is achievable. We do not know whether points in $\overline{K}$- K are or not.

The main idea is to let the senders use block codes whose lengths are relatively prime, for example n and n+1; then, after a short initial period, a kind of quasi-synchronization will be achieved between blocks of length n(n+1):

Once this quasi-synchronization is achieved, the decoder will base its estimate of $\underline{X}^{(1)}$ and $\underline{X}^{(2)}$ on these long blocks, so we have, in effect,a pair of synchronized codes of length n(n+1). If the desired rates (R_1,R_2) are in the region K, then for some particular choice of input distributions (p_1,p_2), $(R_1,R_2)\epsilon K(p_1,p_2)$, and the random coding argument which is used to prove the achievability of (R_1,R_2) in these synchronized case can be modified to handle the quasi-synchronized case as well. (The two codes are chosen "at random" according to the probability distributions p_1,p_2,etc.)

In the noiseless channels of section 2, it is relatively easy to verify that $\overline{K}$ = K except for channel C. In particular, the entire capacity region for channel E is achievable without synchronization, though we know of no explicit scheme for doing this outside the region $\sqrt{R_1} + \sqrt{R_2} \leq 1$. Thus what is possible trivially with synchronization--clearly any point on the line $R_1 + R_2 = 1$ can be achieved on channel E with timesharing and no coding--is also possible without.

However, for channel C the situation is different. Here $\overline{K} \neq K$, and the two regions K,$\overline{K}$ look like this:

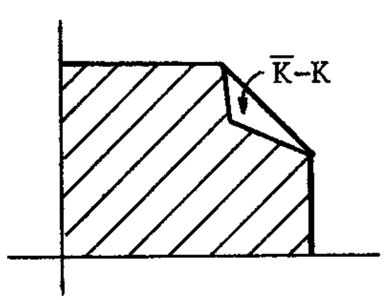

Figure 4.
The ambiguous region
$\overline{K}$-K for channel C.

(The exact description of $\overline{K}$-K is complicated, but it is well approximated by the triangle with vertices (1,½), (½,l), and (1/3,1/3).) We conjecture, but cannot yet prove, that every point in $\overline{K}$-K is still achievable.

Reference

[1] Wyner, A. D.; Recent Results in the Shannon Theory, IEEE Transactions on Information Theory IT-20 (1974), pp. 2-9).