Data Transmission over a Self-Contained Error Detection and Retransmission Channel

By F. E. FROEHLICH and R. R. ANDERSON

(Manuscript received December 10, 1962)

Error control of the detection and retransmission type requires an internal storage buffer when the data source cannot be stopped. With finite capacity there will be occasions when this internal buffer is overfilled. This paper investigates the relationships among the error statistics of the channel, the storage capacity of the buffer, the round-trip transmission delay and the bit rate from the source. It is shown that the process can be treated as a Markov chain. The solution algorithm is programmed for machine computation, and representative cases are solved numerically. For typical values selected from the telephone plant, it is found that buffer capacities of a few hundred bits would be adequate.

The technique described should be useful for solving other problems in queueing theory.

I. INTRODUCTION

Studies during the last few years have shown that in the transmission of digital data over telephone lines, high accuracy can be achieved when the message is encoded in an error detecting code. Correction can then be accomplished by a repeat transmission of the portion of the information containing the errors. These so-called "feedback" techniques have been shown to be very effective in controlling errors.[1,2,3,4]

For some sources of data it is inconvenient or impossible to have the source wait while previous data are being retransmitted. There are also cases where it is required that the output from the receiver be at a uniform rate. This memorandum describes a self-contained error detection and retransmission channel capable of accepting data from the source at a steady rate, or at any rate less than a specified maximum, and of delivering it to the sink at this same rate. The channel is "self-contained," meaning that the channel itself provides enough storage of in-

formation to permit the detection of errors and their correction by retransmission without the data source and sink being aware that these processes are going on. The data source merely puts data into the transmission system at its own rate, and the data sink accepts highly reliable data from the system at the same rate. The relationships among system delay time, error probability, bit rate, and storage capacity are investigated.

The use of feedback error control with a data source which cannot be interrupted was briefly discussed by Reiffen, Schmidt, and Yudkin.[5] A. B. Fontaine has simulated such a system on a computer, using error data collected on private wire circuits.[6] Our analysis has indicated that shorter blocks could well have been used in the experimental simulation, which would have reduced the required storage capacity or increased the time to overflow.

II. THE DATA CHANNEL

A block diagram of the self-contained data channel is shown in Fig. 1. The transmitter consists of a buffer store, an encoder, a modulator, a reverse channel receiver and some logic. The transmission channel itself has a forward path and a reverse path, the latter carrying very little information compared to the former. The receiver consists of a detector, a decoder, a buffer store, and a reverse channel transmitter plus logic.

The forward channel carries data (plus any necessary redundancy and starting codes); the reverse channel carries information indicating whether retransmission is required. Errors in the reverse channel will not appreciably affect the operation. The small amount of information required over this channel permits a high degree of redundancy. In addition, a "fail-safe" code can be used, so that any undetected errors on the reverse channel result in unnecessary retransmissions (subsequently eliminated at the receiver) to ensure against loss of data.

To facilitate discussion, a specific model, chosen for its relative simplicity, is described. Modifications and improvements are apparent and will be briefly discussed. The method of operation is to accept data from the source continuously at a constant rate, R_S bits per second, which is less than the maximum rate, R_L, allowed by the data transmission system. The efficiency then, without considering the error-detecting code, is

$$E = R_S/R_L . \tag{1}$$

The data are transmitted at an effective rate of R_S until a retransmis-

sion is requested. After a retransmission request, data are sent at the higher rate, R_L, until the system is returned to normal.

The change in rate could be made by switching the transmitting speed of the data set. Another method to achieve the data rate change is continuous transmission at rate R_L with interspersed dummy or "fill-in" bits as needed. The two methods are mathematically equivalent, and we shall assume the latter for the discussion in this paper. Thus, in the transmitting buffer the data are organized into blocks of N bits each and sent to the encoder at a rate, R_L, faster than the maximum allowable input rate. In order to equalize the input and output rates of the buffer, "fill-in" bits containing no information are inserted between the blocks of message bits as shown in Fig. 2. The data then pass through the encoder, where additional redundancy is added to allow for error detection. At the encoder, one may either ignore the fill-in bits or encode them, but will probably use them to transmit additional useful information. It is of course possible to place the error control encoder before the buffer, but this increases the required buffer size without gaining any apparent advantages. The signal is then modulated for transmission over the forward path.

Each block of information is retained in the transmitting buffer until it is certain that there will be no retransmission request from the receiver. When a sufficient time interval has elapsed and no retransmission request is received, the block of data is erased from the transmitting buffer. This time interval is taken to be T_D, the maximum round-trip delay for which the system is designed. This includes the transmission time in both directions plus any additional time for logical operations at either end.

The system as described has a sort of natural block length, the number of bits emitted by the source at rate R_S in time T_D

$$N = R_S T_D . \tag{2a}$$

With this block size, it is known that a retransmission request must apply to the immediately preceding block of data bits.

It is shown later that shorter blocks have an advantage in reducing the required buffer size, and hence we let

$$N = R_S T_D / k \tag{2b}$$

where k is an integer. For these shorter blocks, the system must assume a maximum T_D or must include some provision for determining the actual round-trip delay time so that retransmission requests can be associated with the proper blocks of data.

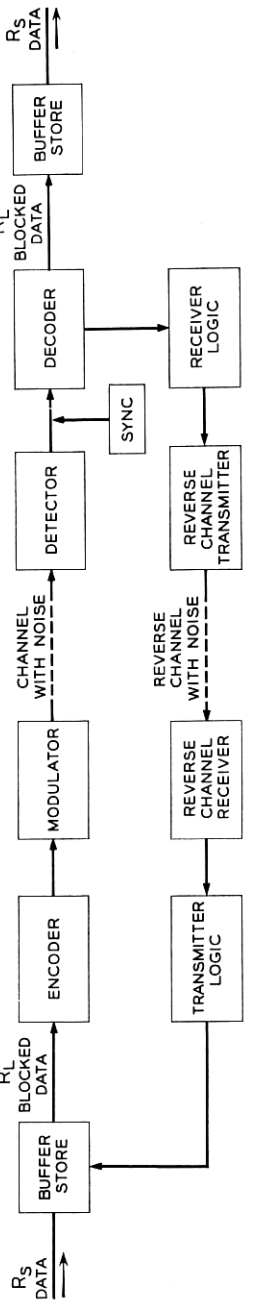

Fig. 1 — Complete self-contained error control channel.

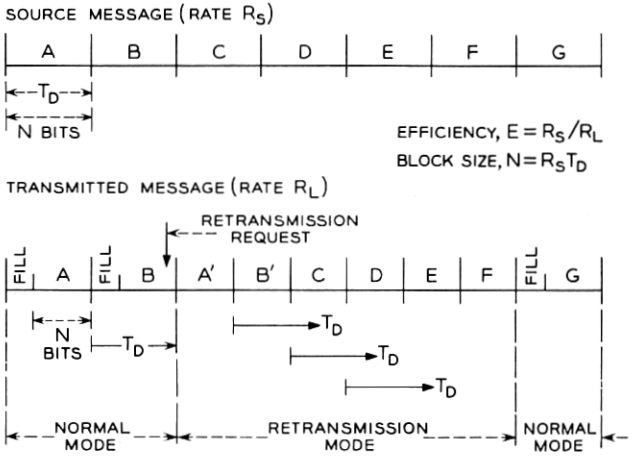

Fig. 2 — Example of time sequence at transmitter.

The number of bits, including both data and fill-in, from the buffer in the same time T_D/k is

$$N + M = R_L T_D/k. \tag{3}$$

In the receiver the demodulated signal is decoded and checked for errors. If no errors are found, the data block, with all redundancy removed, is placed in the receiving buffer. In case an error is detected in the received block of data, a retransmission request is sent to the transmitter via the reverse data channel, and no data are sent to the receiving buffer.

In the transmitter we impose the operating rule: in case a retransmission request is observed, the transmitter will complete the transmission of the current block of $N + M$ bits and then revert to the beginning of the block detected to be in error.† The transmitter then enters the retransmission mode and retransmits information starting with the block in error. During this period, the transmitting buffer continues to receive and store data from the source, thus increasing the quantity of information stored. In order to return the transmitting buffer to its normal state, the fill-in bits are now omitted between the transmitted blocks of data, so that bits will be removed faster than they arrive. This reduces the

† Another way to say this is that the transmitter takes no action on a retransmission request until the end of a full round-trip delay time, T_D, after sending the last bit of the block to be retransmitted. In this form the statement is also true when the transmitter is already in the retransmission mode. Note that in the latter case the time of decision is not necessarily at the end of a block.

information stored in the transmitting buffer and at the same time tends to refill the receiving buffer. The fill-in bits are omitted until both buffers have returned to their normal state.

The above sequence is illustrated in Fig. 2. Block A has been received in error. The retransmission request is noted by the transmitter before the completion of block B. At the conclusion of block B transmission, both A and B are retransmitted. Fill-in bits are now omitted until such time as the transmitting buffer returns to its normal state. This occurs after transmission of block F, if there are no additional retransmission requests.

We note immediately that, in case a number of nearby data blocks are found to be in error, the transmitting buffer may overflow. Similarly, the receiving buffer may empty out, so that for some time no information will be available to the data sink. The frequency of occurrence of these events depends, of course, on the error statistics of the channel, the storage capacity of the buffers, the round-trip transmission delay, the number of fill-in bits allowed between data blocks, and the size of the data block.

Questions to be answered about the self-contained data channel are: How often does the transmitting buffer store overflow and the receiving buffer empty completely? What delay is encountered by the information prior to delivery to the sink? What efficiency can this system achieve? What buffer store capacity is needed? In general, what are the relationships between buffer store size, block length, transmission efficiency, transmission delay, and average time between overflows, in any given message?

III. THE MARKOV PROCESS

In the following development, it will be assumed that retransmission requests are independent with probability P_r. For digital data transmission over telephone lines, individual bit errors are known to be not independent; however, for blocks which are long with respect to the bit error dependence, the retransmission requests will be nearly independent. There is some evidence that over voice telephone circuits at 1000–2000 bits per second the correlation among bit errors becomes so small after 10–15 bit intervals that the assumption of block error independence is acceptable.[1] An estimate of the probability of a retransmission is available, since the block error rate cannot be greater than the bit error rate times the block length.†

† Let λ be the bit error rate in B bits. Then λB is the number of bits in error. The number of blocks in error cannot be greater than λB. The total number of blocks is B/N so an upper limit of probability of block error is

We shall now devote ourselves to the question of the relationship between the storage capacity of the buffer and the average time between overflow of the buffer. It is evident that, since the number of data bits transmitted per unit time is not constant, an actual time calculation is inconvenient. We therefore quantize time into unequal units, such that the number of data bits transmitted per quantum is always the same.

The possible number of bits stored in the buffer form the states of a stochastic process. It will now be shown that, if these are considered only at certain moments of decision, the buffer states, y, form a finite Markov chain.

The only time a decision is made is exactly T_D seconds after the last bit of a block has been transmitted, and the decision consists of three parts:

(a) Which block shall be transmitted?

(b) Shall fill-in bits be transmitted following the data block?

(c) May the transmitting buffer erase a block of data?

The decision depends only on the state of the buffer and on whether a retransmission is requested; there are four cases:

(*i*) *Normal — The system is not in the retransmission mode, and retransmission is not required.* The buffer erases one block; the transmitter sends fill-in bits and then the next block in sequence from the source. By the time of the next decision, the buffer will have replaced the erased block with one block from the source. Thus, at the moment of the next decision, the total change in the buffer storage is zero. The time to the next decision is T_D/k.

(*ii*) *The system is not in the retransmission mode, but a retransmission is requested.* The buffer does not drop any bits. The transmitter backs up to the block at the beginning of the buffer in order to retransmit the block received in error. The transmitter shifts its mode and no fill-in bits are sent. The next decision will be made after one block has been completely transmitted plus T_D seconds, to allow time for another retransmission request to be received. During the retransmission time, EN bits come

---

$$\frac{\lambda B}{B/N} = \lambda N.$$

There may be multiple bit errors in a block, and some of the block errors may not be detected, so

$$P_r \leqq \lambda N. \tag{4a}$$

For the special case where bit errors are independent

$$P_r = 1 - (1 - \lambda)^N \doteqdot \lambda N. \tag{4b}$$

for λ much smaller than 1.

from the source, and during T_D, $R_S T_D$ bits. The total increase in storage due to one retransmission is thus

$$I = R_S T_D + EN = R_S T_D(1 + E/k). \tag{5}$$

The time to the next decision is $T_D(1 + E/k)$.

(*iii*) *Off-Normal — The system has previously entered the retransmission mode and no additional retransmission is requested.* The buffer can drop the block which was received correctly. The transmitter continues with the block following the one just sent, without fill-in bits. The next decision will take place after the time required to transmit one block, in which time EN bits are added to the buffer. Since the buffer has dropped a full block, the amount of data in the buffer has decreased by

$$D = N(1 - E). \tag{6}$$

The time to the next decision is $T_D E/k$.

(*iv*) *The system is in the retransmission mode and another retransmission is required.* This is similar to case (*ii*), except that the transmitter shift is not required since it is already in the retransmission mode. The same number of bits will be discarded at the receiver, but, being already in the retransmission mode, none of these are fill-in bits, so the number of blocks to be retransmitted is greater by the ratio $(N + M)/N$. The transmitter remains in the retransmission mode and fill-in bits will not be sent. The increase in storage is given by (5), and the time to the next decision is $T_D(1 + E/k)$.

Let C be the total storage capacity of the transmitting buffer. When the source rate is constant, the transmitter can send the block as it is received. In this case, the smallest useful capacity, $C_{\min}$, includes the one block to which the retransmission request applies, if received, plus the data which arrive from the source during the round-trip delay preceding the request

$$C_{\min} = N + R_S T_D . \tag{7}$$

If the source rate may fluctuate and the start of transmission must be delayed, C must be larger. The worst case is that in which the source may intermittently stop so the transmitter must wait until the full block is received, in which case the minimum C is one block more. This additional block of storage to compensate for an intermittent source should probably not be charged to the error control system. The ability to provide this feature in a simple manner is, however, an advantage of the system.

There is another meaning for $C_{\min}$. In the normal mode of operation

there must be just this many bits in storage at each time of decision. In setting up the Markov states below, we do not count this irreducible storage, but it is included in the final results for total storage capacity.

We have defined the state of the buffer, y, as the number of bits stored at any instant of decision. With a capacity of C bits, the range of this variable is

$$0 \leqq y \leqq C + 1. \tag{8}$$

The normal state is $y = 0$; overflow is $y = C + 1$.

We can now write down the transition probabilities, p_{ij}, of going from buffer state y_i to state y_j. Starting in the zero or normal state, the buffer stays in the normal state with probability $1 - P_r$ and increases by I with probability P_r

$$p_{0,0} = 1 - P_r, \qquad p_{0,I} = P_r. \tag{9a}$$

If the buffer is within D states of normal, at the next decision it will either return to normal or will increase by I

$$p_{y,0} = 1 - P_r, \qquad p_{y,y+I} = P_r \quad \text{for} \quad 0 \leqq y \leqq D \tag{9b}$$

If the buffer is more than D states from normal and more than I states from overflow, it will decrease by D or increase by I, but can neither return to normal nor overflow

$$p_{y,y-D} = 1 - P_r, \quad p_{y,y+I} = P_r \quad \text{for} \quad D < y \leqq C - I. \tag{9c}$$

If the buffer is within I states of overflow, the buffer will either decrease by D or go to overflow

$$p_{y,y-D} = 1 - P_r, \quad p_{y,C+1} = P_r \quad \text{for} \quad C - I < y < C + 1. \tag{9d}$$

In order to calculate the time to overflow, we force the buffer to stay in the overflow condition once it enters this state; i.e., the overflow state is made "absorbing"

$$p_{C+1,\,C+1} = 1. \tag{9e}$$

For all other transitions $p_{ij} = 0$. The transition matrix is

$$T = \{p_{ij}\}. \tag{9f}$$

In addition, we let the process start in the normal state with probability 1. The buffer state, in response to the retransmission signal, depends only on the buffer state at the previous moment of decision. This is the fundamental property for a process to be a Markov chain.

A schematic representation of the Markov chain described by equa-

tions (9) is given in Fig. 3. The over-all operation of the transmitter may be seen in Fig. 4, which shows the internal state diagram of a sequential machine which might be used to implement the transmitter. The states of the sequential machine are the same as the states of the Markov process, except that several of the latter may map into a single one of the former.

The arrow labels — A,B/C,D — are identified as follows. In all cases, a dash means the item is immaterial.

Transmitter inputs:

A — Has a retransmission request been received?

0 — no 1 — yes

B — What is the state of the buffer?

0 — empty (except for $C_{\min}$)

I — partially filled

1 — over-filled

Transmitter outputs:

C — May a block be dropped from storage?

0 — no 1 — yes

D — Which block shall be sent next?

D_{n-1} was the block which was just sent. D_n is the next block in sequence, and D_{n-m} is the mth block before.

F_1 and F_2 are fill-in bits. Note that if $F_1 = F_2$, two states may be combined.

Fig. 4 also applies to the receiver, except for reinterpretation of the labels.

Receiver inputs:

A — Has an error been detected?

0 — no 1 — yes

B — State of receiving buffer

0 — full

I — intermediate

1 — empty

Receiver outputs:

C — Shall this block be sent to output store?

0 — no 1 — yes

D — Shall a retransmission request be sent?

These will all be 0 except the two labelled D_{n-m} and D_{n-m-1}, which will be 1.

For certain relations among the quantities involved, the matrix can be partitioned into several closed sets[7] of states, such that it is not possible to make the transition from a state in any one closed set to a state in

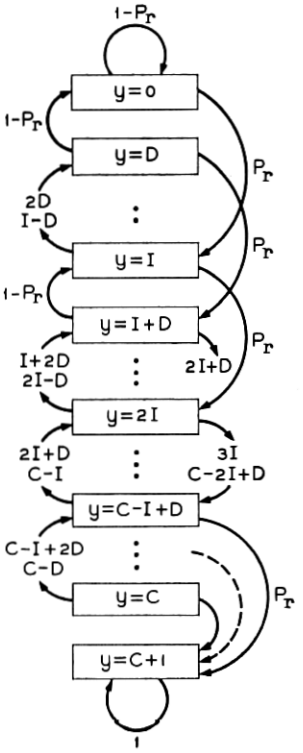

Fig. 3 — Markov state diagram.

any other such set. The states which cannot be entered from the normal state by any path may be removed from the matrix, thus reducing its size. This can be done by dividing out the greatest common factor in D, I, N, and C. A large number of the cases of interest are still included when this "normalizing factor" is made equal to D.

IV. CALCULATIONS

Following the method outlined in Kemeny and Snell,[8] we let Q be the transition matrix of all the transient states, i.e., matrix T, excluding the overflow state. Let J be the identity matrix. Then

$$G = (J - Q)^{-1} \tag{10}$$

exists and is called the fundamental matrix of the Markov process, with the following interpretation. Each element n_{ij} of G is the mean number

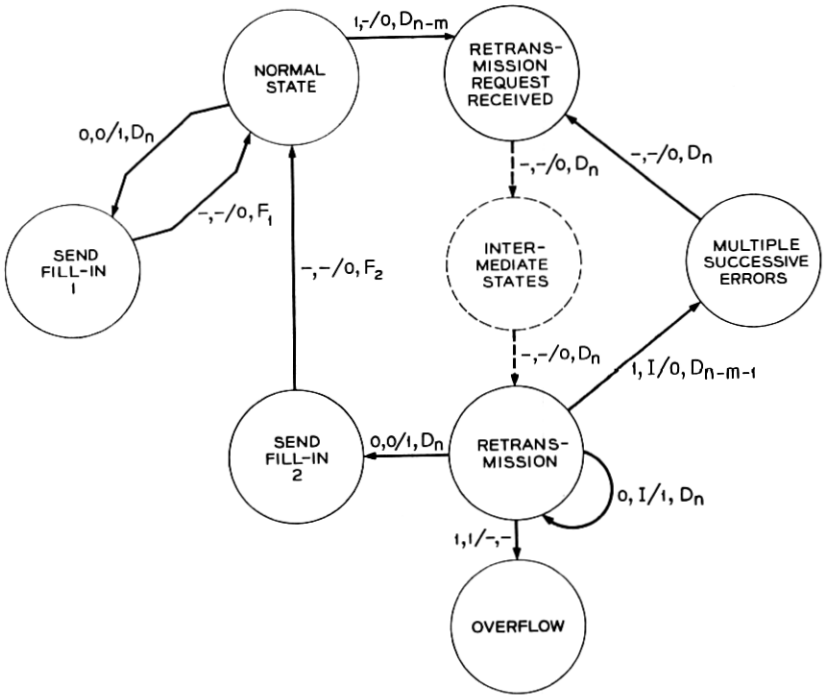

Fig. 4 — Diagram of internal states for transmitter.

of times the process is in state j, given that it started in state i. With $i = 0$ for starting in the normal state, the row sum over j is the mean number of times the process is in any of the transient states, from which we can calculate the mean time to the first overflow. Thus, the average number of decisions before overflow is

$$\langle n \rangle = \sum_{j=0}^{\sigma} n_{0j} . \tag{11}$$

Higher moments, in particular the second, can be found by additional operations on the fundamental matrix.[7]

A computer program was written to do the matrix arithmetic, and a few representative cases were solved numerically. The program computes the average number of blocks transmitted before overflow and the variance about this mean. The standard deviation is usually large, nearly equal to the mean. Typical examples are: when mean number of blocks before overflow was 23, standard deviation was 19; when mean was 949, standard deviation was 943; and when mean was 4795, standard

deviation was 4792. Thus the mean is a poor estimate of the actual time to overflow for any specific message, but is meaningful when a large number of transmissions are considered.

The calculations to this point have been in terms of the number of blocks, and we now convert back to actual time. Instead of a straight sum on n_{0j}, we multiply each term by the actual time taken.

There are four terms corresponding to the four cases described under the Markov process. The average time for each of the four cases is

$$(i)\ n_{00}(1 - P_r)T_D/k$$

$$(ii)\ n_{00}P_r(1 + E/k)T_D$$

$$(iii)\ \sum_{j=1}^{C} n_{0j}(1 - P_r)ET_D/k$$

$$(iv)\ \left[\sum_{j=1}^{C} n_{0j} - 1\right] P_r(1 + E/k)T_D .$$

The average time to overflow is the sum of these four:

$$\frac{t_{\text{ave}}}{T_D} = n_{00}\left(P_r + \frac{1 - P_r}{k} + \frac{EP_r}{k}\right) + \sum_{j=1}^{C} n_{0j}(P_r + E/k) - P_r(1 + E/k). \tag{12}$$

V. RESULTS

As expected, the average time before the buffer overflows will increase when the buffer capacity is increased, and when the following variables are decreased: the bit rate, the round-trip delay, the probability of retransmission, the efficiency, and the block size. The number of variables can be reduced by measuring time in units of T_D, the round-trip delay, and bits in units of R_LT_D, the number of bits from the buffer in time T_D. Since the block error rate depends on the length of the block, the probability of retransmission is modified by the block length. The variables of the system, all of which are now dimensionless, become

$$C^* = C/R_LT_D$$

$$N^* = N/R_LT_D$$

$$E$$

$$P^* = P_rR_LT_D/N$$

$$t^* = t/T_D$$

A number of curves are plotted to show the expected time between overflows as a function of the probability of retransmission. For each curve, the size of the buffer, the block size, and the efficiency are held constant. When the expected time, t/T_D, is greater than about 100 (corresponding to several seconds of transmission for reasonable values of T_D), the curves are nearly linear on log-log paper, and only this portion is plotted.

To use the curves, it is assumed that the transmission line parameters, R_L and T_D, are known. In order to facilitate interpretation of the curves, some reasonable specific values have been assigned to these parameters and the corresponding values of time, buffer size, and block length have been calculated. The assignments are as follows: Let R_L be 2000 bits per second; this could be a 2400 bps data set with an $83\frac{1}{3}$ per cent efficient error-detecting code. Let T_D be 120 ms. Then $R_L T_D = 240$ bits, the total number of bits sent in one round-trip delay time. Some other parameters are given in Table I.

Fig. 5 shows the time gained by increasing the capacity of the buffer store. For this set of curves the efficiency is 0.5 and the block length is $0.5\ R_L T_D$; that is, the block is as long as the maximum round-trip delay. When the efficiency is increased to 0.75 and 0.9, with the same block length ($0.5\ R_L T_D$), the results are as shown in Figs. 6 and 7, respectively. The storage capacity required to provide a specified time to overflow at a given probability of retransmission increases markedly with efficiency. The same effect is shown in Fig. 8, where the capacity is held constant for several efficiencies. The source bit rate at $E = 0.75$ is 50 per cent greater than at $E = 0.5$, and at $E = 0.9$ the bit rate is up by 80 per cent. The cost of this increased bit rate is either the extra buffer storage or the reduced time between overflows. Some of the data from Figs. 5–8 are

TABLE I — OPERATING PARAMETERS
(Given that $R_L = 2000$ b/s and $T_D = 120$ msec)

E	R_S (bits/sec)	N (bits)	I (bits)	C_{min} (bits)
0.5	1000	20	130	140
		120	180	240
0.75	1500	20	195	200
		120	270	300
		180	315	360
0.9	1800	20	234	236
		120	324	336
		216	410	432

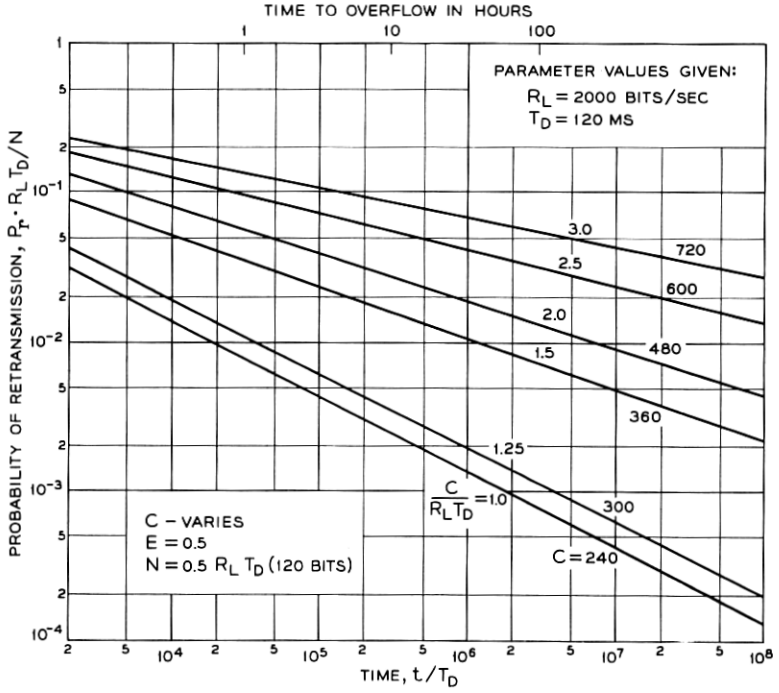

Fig. 5 — Effect of buffer size: $E = 0.5$.

shown in Table II, using the arbitrary assignments $R_L = 2000$ b/s, $T_D = 120$ ms, and $N = 120$ bits.

In all the above cases, the block lengths have been the same, $0.5\ R_L T_D$ (120 bits). Only when the efficiency is 0.5 does this represent the so-called "natural" block, i.e., the number of bits from the source in one round-trip delay time; at the increased bit rates of the higher efficiencies, the natural block length is also increased. The effect of increasing the block length in one case is shown in Fig. 9, which can be compared to Fig. 6. The required capacity for a given time to overflow has increased markedly. We therefore investigate the effects of shorter blocks.

Fig. 10 illustrates the case where each natural block is divided into three shorter blocks. A decision is made at the end of each arrow, and the fourth block back is either dropped from the buffer or is retransmitted. For example, when a retransmission is received while sending B_3, both A_1 and A_2 have been dropped and A_3 is the next block to be sent. With sufficiently inexpensive logic in the terminals, improved per-

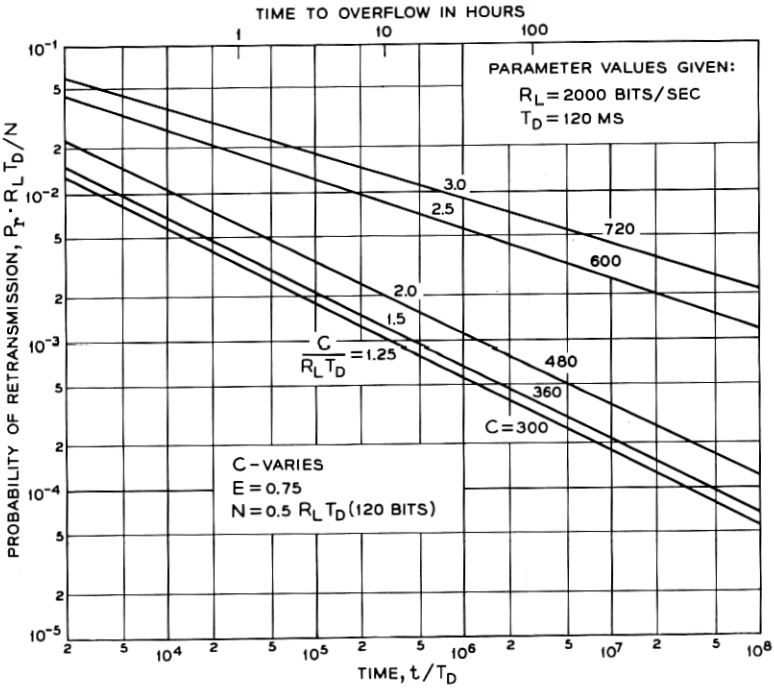

Fig. 6 — Effect of buffer size: $E = 0.75$.

formance is possible on short loops by using the actual value of T_D. In the example, we might have already dropped A_3 and therefore start the retransmission with B_1.

In Fig. 11 we show the effect of decreasing the block size, at constant capacity and efficiency. Similar results for a larger capacity and efficiencies of 0.5 and 0.75 are shown in Fig. 12.

It is somewhat difficult to visualize all of these effects when plotted separately. We attempt to summarize some of the results in Fig. 13. For these curves the normalized retransmission probability, P^*, is held constant, and buffer storage capacity is held to the minimum usable value, as given by (7); that is, the capacity is the natural block length plus the actual block length, and therefore decreases with either the block size or the efficiency. Both the latter are allowed to vary and we show the effect on the time to overflow.

There is little effect from changing the block size — except on the buffer capacity. One would therefore choose the smallest practical block.

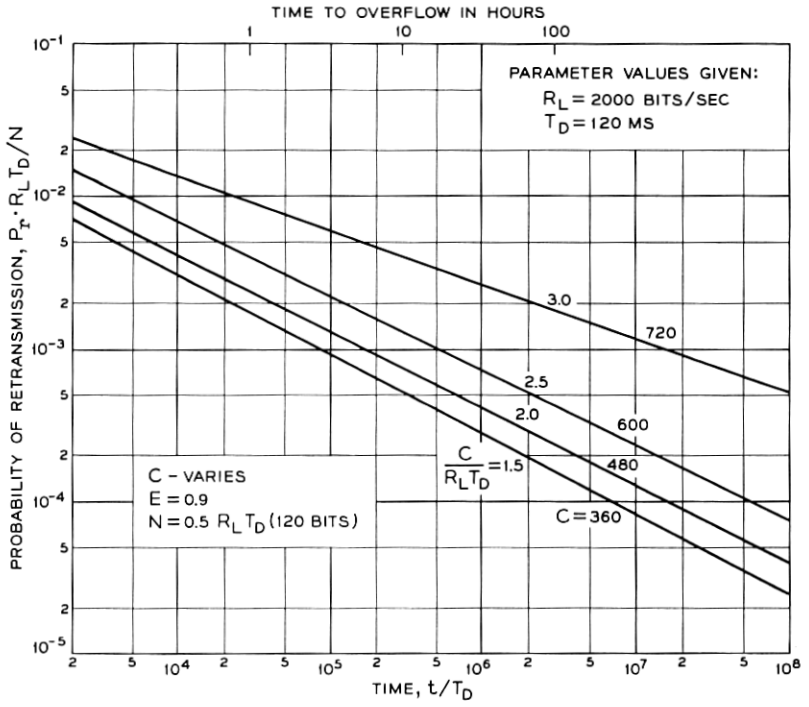

Fig. 7 — Effect of buffer size: $E = 0.90$.

However, as the efficiency is increased, the required buffer capacity is increased, although not rapidly, and the time between overflows decreases. As shown earlier (Figs. 5–7, 9) it is possible to regain this loss in time to overflow by modest increases in buffer capacity over the minimum used here. Since the increased efficiency increases the maximum source rate, this is certainly the direction to go, up to the point where the increased rate is worth less than the cost of the additional storage required.

VI. DELAY

For smooth flow to the sink the receiving buffer must have the same capacity as the transmitting buffer, and will normally be kept full. Thus the receiving buffer will introduce a delay in the message of

$$\tau = C/R_s \,. \tag{14}$$

This is in addition to the delay of $T_D/2$ from the transmission line.

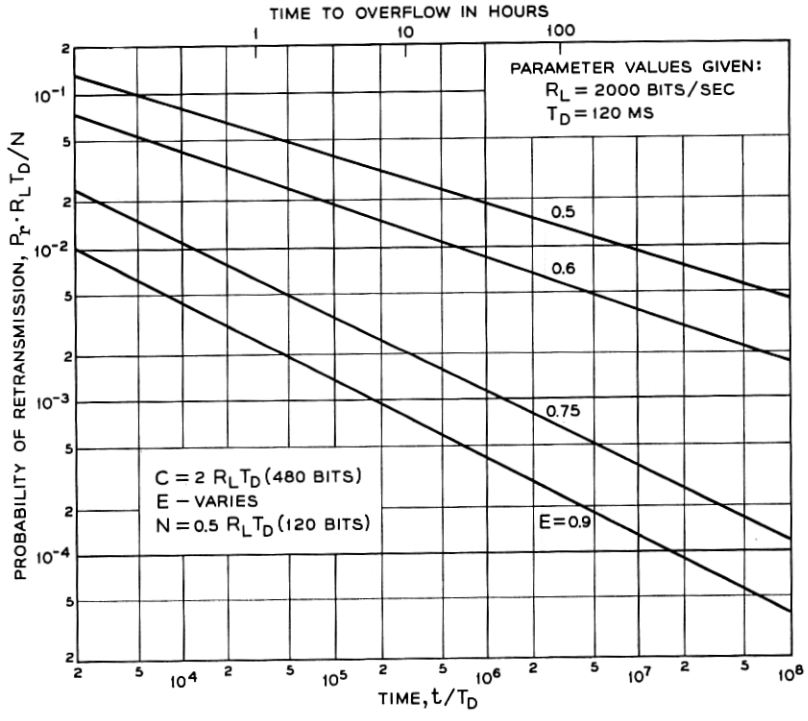

Fig. 8 — Effect of efficiency: buffer capacity fixed.

TABLE II — MEAN TIME TO OVERFLOW

(Given that R_L = 2000 b/s, T_D = 120 msec, N = 120 bits)

$E = R_S/R_L$	C (bits)	Ave. Time to Overflow (Hours)	
		P^* = 0.01	P^* = 0.001
0.5	C_{min} (240)	0.67	66.6
0.75	C_{min} (300)	0.12	11.2
0.9	C_{min} (336)	0.03	2.90
0.5	360	44.4	>1 year
0.75	360	0.15	14.9
0.9	360	0.04	3.14
0.5	480	245.3	>1 year
0.75	480	0.42	44.1
0.9	480	0.06	5.32
0.5	600	>1 year	>1 year
0.75	600	6.29	>1 year
0.9	600	0.15	17.93

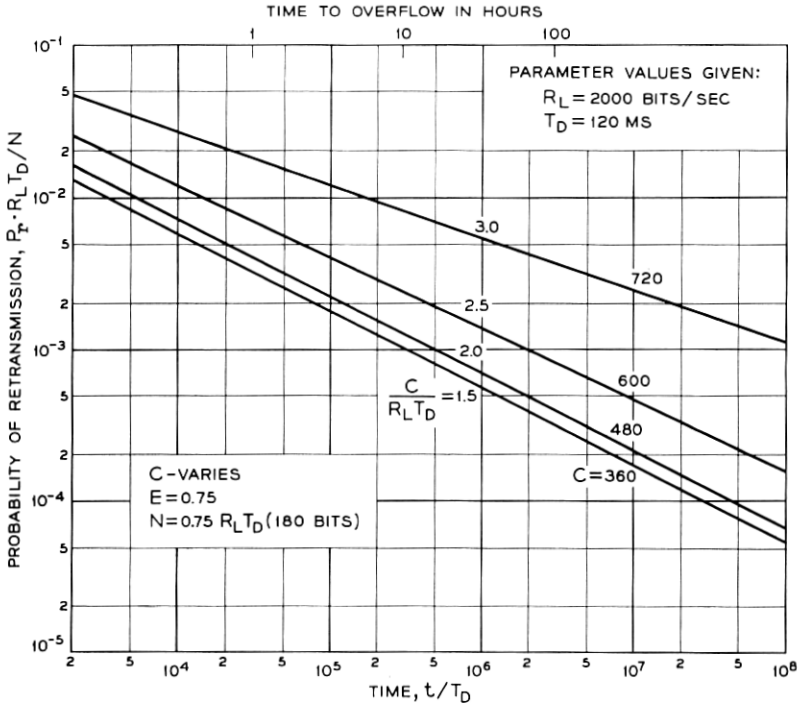

Fig. 9 — Effect of buffer size: longer block, $E = 0.75$.

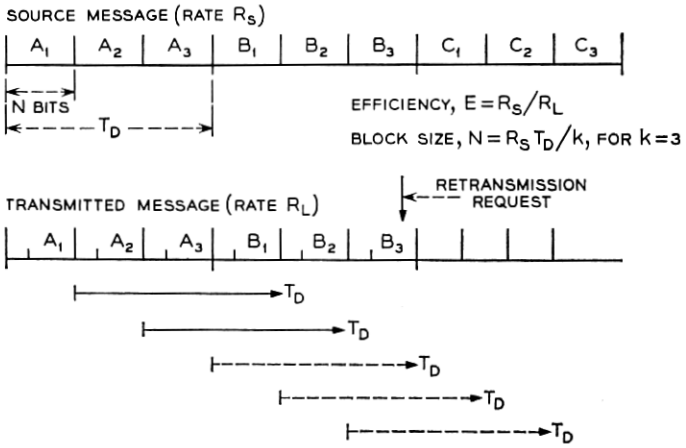

Fig. 10 — Example of time sequence with shorter blocks.

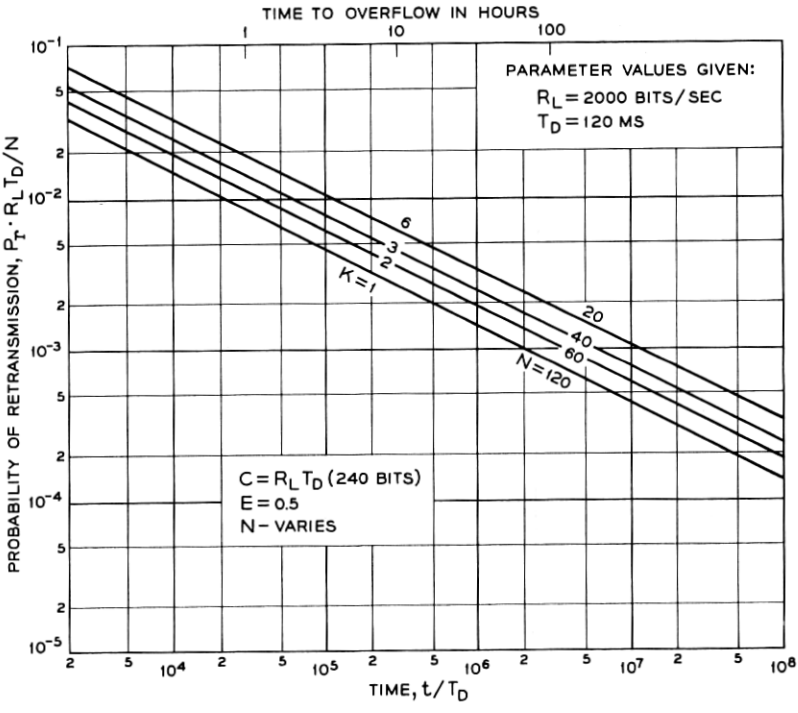

Fig. 11 — Effect of block size: fixed buffer capacity, $E = 0.5$.

There are other choices for operating the receiving buffer which will decrease the delay at the expense of irregularity of flow to the sink, which may be tolerable in many cases. If there were no receiving buffer at all, the delay would be zero except when retransmissions were required. When retransmissions are required, however, there would be additional delay until the block is received correctly, up to a maximum given by (14). The flow to the sink would not be smooth; each block would be delivered at rate R_L, followed by an interval when no data are being delivered. Various compromises between these extremes are possible. For example, buffer capacity of a single block would permit data to be delivered to the sink at the source rate with no interruptions until a retransmission is requested. Then the sink must alternately wait and accept data at the higher line rate until the process returns to normal. The delay is variable, the minimum being

$$\tau = N/R_s \tag{15}$$

with the maximum again given by (14).

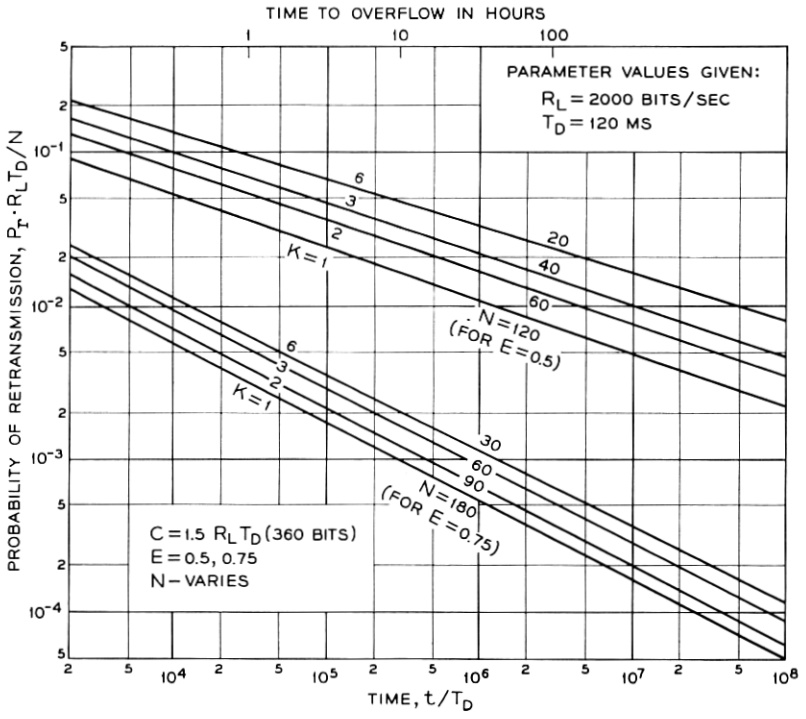

Fig. 12 — Effect of block size: fixed buffer capacity, $E = 0.5$ and 0.75.

This is one case where it has been possible to develop a calculable relationship between the message delay involved in error control and the resulting error rate.

VII. OTHER MODIFICATIONS

The system may be designed to take any of several actions when an overflow of the buffer occurs. The source and sink may be stopped, requiring manual resetting; they may be temporarily halted for a time sufficient for the system to clear; or, without stopping the source, the uncorrected data block may be delivered to the data sink, with or without an indication that the particular block contains errors.

One desirable modification would be to act sooner on receipt of the retransmission request. The transmitter would not continue to the end of the current block, but would immediately back up to the beginning of the block in error. This procedure could be quantized by using blocks a fraction of N in length. As indicated above, this procedure would require

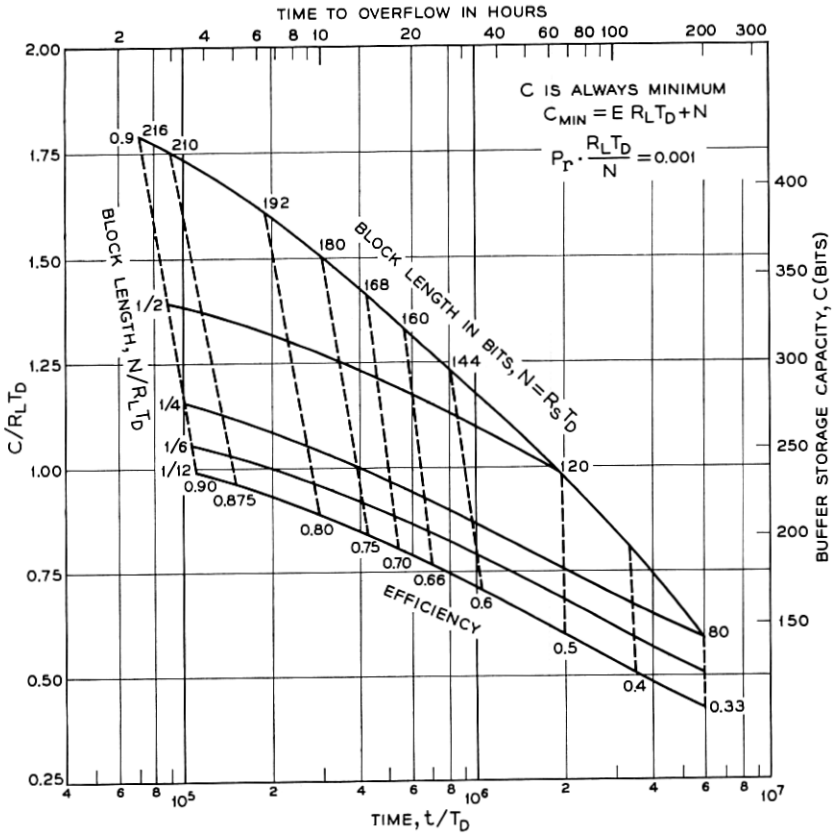

Fig. 13 — Capacity and time to overflow as functions of efficiency and block length.

either a knowledge of the actual round-trip delay or inclusion, in the retransmission requests and the retransmission, of an indication of the exact block (or fraction) involved. Another modification which would improve performance on shorter loops would be to make a preliminary measurement of the round-trip delay and adjust the operation accordingly. This could be done automatically.

Earlier, we mentioned the problem of an irregular input sequence and indicated that one additional block of storage was necessary. If this block is not counted, the performance level will be as given for a regular source, except for the possible gain arising from the probability of the intermittent source being stopped during the time when retransmissions are re-

quired. The output will be delayed an additional time corresponding to one block of data, but will be smoothed considerably — the rate will be constant except when waiting for the source.

It has been assumed that, once an error is detected, all subsequent received data are ignored until that block has been retransmitted and received correctly. With more complicated bookkeeping it would be possible to save some of these blocks, reducing the amount of retransmission required. On the other hand, since errors do occur in bursts on many transmission channels, the immediately succeeding block would have a higher-than-average error rate, and so might not be worth saving.

VIII. CONCLUSIONS

It has been shown that it is possible to calculate the performance of a self-contained error-control system by treating the system as a Markov process when the system consists of (*a*) an error-detection code, (*b*) provision for requesting and accomplishing retransmissions as necessary, and (*c*) buffer storage to allow smooth, uninterrupted flow from the source to the sink. Failure occurs when a sufficient number of retransmissions are requested in a short enough time that the total information to be stored exceeds the capacity of the buffer.

Whenever an overflow is about to occur, we could ignore the retransmission request and deliver the block as-is, in which case it appears to the sink as an error. It seems reasonable to require that this type of error should have about the same frequency of occurrence as undetected errors. For voice channels using reasonably simple codes, we might assume an undetected error rate of 10^{-8} or about one error per day.[1,3] We might also require the efficiency to be about that of the error detecting code.

With these criteria, it appears clear that one should not try to work with minimal storage, because of the relatively short time to overflow. Neither should one try to push the efficiency very high, or the required capacity grows out of bounds. A reasonable compromise for voice channels would be a buffer capacity somewhat less than 1000 bits.

We get a slightly different answer if we consider instrumentation. It is likely to be economically infeasible to build a buffer of this size with individual bit storage devices, especially since serial access is adequate. However, with bulk storage such as a circulating delay line or a magnetic tape loop, moderate increase in buffer size is not costly, and several thousand bits would be available about as cheaply as a few hundred. This would permit buffer efficiencies close to unity.

Results for any other specific cases can be easily calculated with this

computer program. It is apparent that a number of modifications in the model are possible and would serve to reduce the required storage. The transition matrix would merely have to be changed to correspond to the new model; the matrix arithmetic would be the same.

The details of the chosen model and the examples were taken from a specific data transmission problem. The techniques, both the model and the method of solution, are applicable to a wider variety of problems where buffering is a consideration.

We should like to acknowledge the assistance of H. O. Burton in consultation on the mathematics of the Markov process. We appreciate the continued encouragement of G. W. Gilman, who suggested the use of feedback error control with a data source which cannot be interrupted.

REFERENCES

1. Bennett, W. R., and Froehlich, F. E., Some Results on the Effectiveness of Error-Control Procedures in Digital Data Transmission, I.R.E., Trans. Comm. Syst., **CS-9,** March, 1961, pp. 58–65.
2. Schwartz, L. S., Some Recent Developments in Digital Feedback Communication Systems, I.R.E. Trans. Comm. Syst., **CS-9,** March, 1961, pp. 51–57.
3. Cowell, W. R., and Burton, H. O., Computer Simulation of the Use of Group Codes with Retransmission on a Gilbert Burst Channel, Trans. A.I.E.E. (Comm. & Elect.), No. 58, January, 1962, pp. 577–585.
4. Brown, A. B., and Meyers, S. T., Evaluation of Some Error Correcting Methods Applicable to Digital Data Transmission, 1958 I.R.E. Natl. Conv. Record, Pt. 4, March, 1958, pp. 37–55.
5. Reiffen, B., Schmidt, W. G., and Yudkin, H. L., The Design of an Error-Free Data Transmission System for Telephone Circuits, Trans. A.I.E.E. (Comm. & Elect.), No. 55, July, 1961, pp. 224–231.
6. Fontaine, A. B., Queueing Characteristics of a Telephone Data Transmission System with Feedback, A.I.E.E. Conference Paper 62-1441, Fall General Meeting, October 9, 1962.
7. Feller, W., *An Introduction to Probability Theory and its Applications*, Vol. I, 2nd ed., John Wiley and Sons, New York, 1957.
8. Kemeny, J. G., and Snell, J. L., *Finite Markov Chains*, D. Van Nostrand Co., New York, 1960.