

An Improved Selective-Repeat ARQ Strategy

E. J. WELDON, JR., MEMBER, IEEE

Abstract—A new procedure for handling retransmissions in a selective-repeat ARQ system is proposed. This procedure can operate with a receive buffer of minimal size; in addition it places little computational load on the transmit and receive processors.

The procedure is simple enough that its throughput can be calculated exactly. Analysis of this strategy shows that:

- 1) it yields higher throughput than earlier ARQ techniques;
- 2) for modest receive buffer size, its throughput differs little from channel capacity;
- 3) as buffer size increases, throughput approaches channel capacity.

The final section of the paper considers the performance of ARQ systems on channels in which errors occur in bursts. It indicates that on reasonably good channels, error burstiness has little effect on throughput.

I. INTRODUCTION

A. Background

VERY early work on ARQ systems was done by Chang [1], Harris *et al.* [2], and Metzner and Morgan [3]. These authors examined the basic nature of communications with feedback and classified the types of ARQ systems. Shannon [4] investigated the theoretical limits of ARQ systems and demonstrated that although feedback could not increase the capacity of a memoryless channel, it could improve the reliability of such channels at all rates below capacity. Weldon [5] applied these results to the binary symmetric channel and derived bounds on the reliability achievable with ARQ.

Along more practical lines, Reiffen *et al.* [6] and Cowell and Burton [7] describe early ARQ systems employing a stop-and-wait strategy. Stuart [8] describes the selective-repeat strategy and Benice and Frey [9] compare the performance, in terms of undetected error probability and throughput, of stop-and-wait, go-back-N, and selective-repeat ARQ strategies. Somewhat later, Balkovic and Muench [10] examined the performance of these ARQ strategies on the switched telephone network.

All of the early commercial systems which employed ARQ used the stop-and-wait strategy, no doubt because of its simplicity; IBM's widely used binary synchronous communication (BISYNC) [11] protocol is an example. In the early and mid-1970's, the widespread use of satellites for data communications [12], coupled with the declining cost of digital hardware, encouraged the use of continuous-ARQ strategies. Pro-

ocols like SDLC [13] and ADCCP [14] employ the go-back-N strategy; its performance is substantially better than that of stop-and-wait and it can be implemented at moderate cost. Studies by Kaul [15], Traynham and Steen [16], Doll [17], and others have determined the throughput performance of these protocols under a wide range of conditions.

Recently there has been considerable interest in the selective-repeat ARQ strategy. For example, in [18], Easton analyzes the performance of real protocols employing this technique. In [19], the authors derive an approximate expression for the throughput of a selective-repeat system; their analytical results agree closely with data measured on an experimental satellite link [20]. All of the papers show a wide range of system parameters over which the selective-repeat strategy is markedly superior to go-back-N.

In a selective-repeat ARQ system, blocks are numbered and ACKed or NACKed by number. Assuming a noiseless feedback link, only erroneous blocks are repeated provided the receiver has sufficient buffer capacity to store all correctly received blocks until earlier errored blocks can be replaced. If the receiver buffer is infinite and if the channel makes block errors with probability P , then the throughput of the link (i.e., the average number of blocks communicated¹ per transmission) is just the probability that a block will be received correctly. That is

$$T = 1 - P. \quad (1)$$

The forward data link may be regarded as an M -ary erasure channel where M is the number of possible choices for a block. The capacity of such a channel is well known to be $1 - P$ and, as proved by Shannon [4], this capacity is not increased by the use of a feedback link. Hence the selective-repeat strategy represents the optimal form of ARQ when the receive buffer size is unlimited.

Unfortunately, real receivers have finite buffers and the simple selective-repeat procedure described above is no longer optimal. Basically, when the receive buffer fills, data are discarded causing a reduction in throughput.

Recently several modifications of the basic selective-repeat strategy have been proposed. Morris [22], [23], following Sastry [21], proposes to repeat blocks S times rather than just once when a NACK is received, where S is the number of blocks stored in the data link. This virtually eliminates buffer overflow, but because blocks are repeated many times, performance suffers. For high error-rate conditions, Lin and Yu [24] improved on Morris' scheme by having the transmitter

¹ In this paper, we say a block is "communicated" if it is transmitted and received correctly.

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication without oral presentation. Manuscript received February 6, 1981; revised July 20, 1981. This work was supported in part by NATO under Grant RG 004.80.

The author is with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 and ADTECH, Inc., Honolulu, HI 96816.

repeat the requested block a number of times and also repeat subsequent NACKed blocks. The same authors analyze a variation on selective-repeat ARQ in [25] and lower-bound its throughput.

Practical selective-repeat ARQ procedures fall far short of channel capacity when the error probability is high. There would appear to be two reasons for this:

1) At the high data rates concordant with selective-repeat ARQ, there is no time to do the processing required.

2) The amount of storage available at the receiver is limited (usually to one round-trip delay).

In this paper, we show that the first of these factors is relatively unimportant and that it is the amount of receiver storage that controls the performance of a selective-repeat ARQ system.

B. This Work

In the present paper we present a new selective-repeat ARQ strategy and show that its throughput is superior to that of earlier techniques. In Section II we define our channel model precisely. In Section III we define the new strategy and determine its throughput. In the final section we consider briefly the performance of ARQ systems when errors occur in bursts.

II. SYSTEM MODEL

The system we are considering consists of a transmitter connected to a receiver by a data link; this is shown in Fig. 1. The transmitter sends n -bit blocks (or frames) to the receiver; these may contain errors. The receiver also sends n -bit blocks to the transmitter and these we assume are error-free.² These reverse blocks carry feedback information to the transmitter about the state of the receiver. They may also carry other information, but this is not of concern to us.

The major system components will now be described.

A. Data Link

The link is capable of simultaneous bidirectional transmission at a rate of R bits/s in each direction. It has a propagation delay of t_{prop} seconds in each direction; hence the link "stores"

$$s_{prop} = 2Rt_{prop} \quad (2)$$

data bits. In this paper, we are concerned primarily with links having s_{prop} in the range 10^3 - 10^8 bits.

The delay in transmitting a block across the link in either direction is $t_{emis} + t_{prop}$, where t_{emis} is the emission delay of an n -bit block emitted at rate R . That is, $t_{emis} = n/R$. At the receiver, there is an additional delay between the reception of a block and the transmission of a reply; similarly at the transmitter there is a delay between the receipt of the reply and the transmission of the next block. Adjusting the relative phasing of the two block streams can force one of these delays to zero, while the other will not be greater than one block

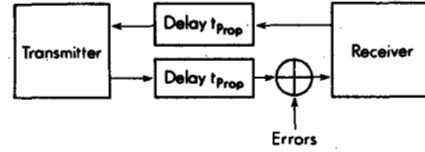


Fig. 1. Communication system model.

time. Thus, the number of blocks stored by the link is³

$$S = \left\lceil \frac{R(3t_{emis} + 2t_{prop})}{n} \right\rceil \quad (3)$$

and the total round-trip delay is nS/R seconds. We refer to the quantity S as the (block) storage of the link.

The link causes errors to occur in blocks sent in the forward direction. In calculating the throughput of our proposed strategy (Section III), we assume that these errors are independent events and occur with probability P . In Section IV, we consider the case where errors occur in bursts.

B. Transmitter

This device forms numbered frames and passes them to the link. (Each frame also contains a polynomial check sequence and, unavoidably, a number of other overhead bits; these are not considered further in this study, however.) The transmitter and receiver converse using a continuous-ARQ protocol which informs the transmitter of the error status of each block received at the receiver. The transmitter stores a copy of each sent block until it is acknowledged (ACKed) by the receiver. Since the link storage is S , the transmitter must also have the capacity to store S blocks. It is not hard to show that this transmit buffer size is sufficient to support a selective-repeat protocol regardless of the receive buffer size.

C. Receiver

This device recalculates the polynomial check sequence on each received word. If the check sum is nonzero, a block error has been detected. In this paper, we assume that all block errors are detected by the polynomial check sequence. No attempt is made to correct any detected errors; the transmitter is simply notified that an error was received via the next block on the reverse channel. Based on prior numbered ACK's, the transmitter can determine the number of the corrupted block. A copy is transmitted at the completion of the current block transmission.

Since the data link stores S blocks, any selective-repeat ARQ scheme must employ a receiver capable of storing S blocks; otherwise blocks could not be delivered continuously in correct order following an error. In this paper, we assume that the receiver has a buffer of size qS blocks, $q = 1, 2, 3, \dots$. This implies that buffer overflow will occur only after q attempts to send a copy of an errored block have failed.

It is interesting to observe that were the receive buffer size zero, the selective-repeat protocol would become a go-

² This is not as unrealistic as it seems. Although on a physical link, errors will occur in the reverse channel, their effect can be almost totally eliminated by the link protocol.

³ The notation $\lceil x \rceil$ refers to the smallest integer greater than or equal to x .

back-N protocol. That is, all blocks following an error would be rejected until a correct copy of the errored block was received.

III. THE NEW STRATEGY

We define throughput T as the average number of blocks communicated per transmitted block. It is more convenient to calculate the average number of transmissions necessary to communicate a block, a quantity we denote by β , and then to determine T from the equation

$$T = 1/\beta. \tag{4}$$

A. Description of the Strategy

The basic idea of the strategy is to repeat NACKed blocks multiple times with the number of repeats increasing as the receive buffer approaches overflow. The strategy operates as follows:

- Level 0 If an ACK is received at the transmitter, send a new block.
- Level 1 If a NACK is received after the transmission of a new block, the block is repeated n_1 times.
- Level 2 If all n_1 copies are received in error (corresponding to n_1 NACK's) the block is repeated n_2 times.
- ⋮
- Level q If all n_{q-1} copies are received in error, buffer overflow occurs. The block is repeated n_q times.
- Level $q + 1$ If all n_q copies are received in error, buffer overflow occurs. The block is repeated n_q times. The strategy remains at this level until the block is communicated.

At each level, the transmitter sends new blocks whenever it does not have repeats to send. It should be noted that the "classical" selective-repeat strategy results when $q = 1$ and $n_1 = 1$.

As a result of multiple errors the transmitter may be required, by the foregoing, to send two or more blocks at the same time. In this case, the lowered numbered (older) block is sent first.

After $q + 1$ attempts to send a block have failed and level $q + 1$ has been reached, the receive buffer is considered to be full. The next $S - 1$ received blocks are discarded and, of course, subsequently repeated by the transmitter. It is recognized that the receive buffer may not really be full (as a result of numerous errors in blocks which were to be stored in the buffer). Also it is recognized that discarding blocks which may be the first correct copy of a particular block reduces throughput. Nevertheless the strategy operates in this manner so that the expected number of transmissions needed to communicate block i is independent of errors in block j and its repetitions for all $j \neq i$. Hence, the expected number of transmissions is the same for all blocks.

B. Calculation of Throughput

We calculate throughput by first determining β , the average number of block transmissions necessary to communicate a block. Throughput is then given by (4).

Three types of blocks may be involved in communicating a block; the original block, copies sent in response to NACK's and blocks lost because of receive buffer overflow. Blocks discarded because of errors in block i or its copies are attributed to the transmission of block i . Thus, for example, in a system with $q = 1$ and $n_1 = 1$, if block 0 is received in error and its repeat (block S) is also received in error, then blocks $S + 1$ through $2S - 1$ will be discarded because of buffer overflow. If block $2S$ (the second repeat) is received correctly, then by our reckoning $S + 1$ blocks were transmitted to communicate block 0.

With probability $1 - P$ the first transmission of a block is successful. If the first transmission is received in error (probability P) then exactly $1 + n_1$ transmissions will be needed unless all n_1 repeats are in error. Hence, we have

$$\Pr(1) = 1 - P \tag{5a}$$

$$\Pr(1 + n_1) = P(1 - P^{n_1}). \tag{5b}$$

If all n_1 repeats are in error, then exactly $1 + n_1 + n_2$ transmissions will be required, so

$$\Pr(1 + n_1 + n_2) = P^{1+n_1}(1 - P^{n_2}) \tag{5c}$$

and for $i \leq q$

$$\begin{aligned} \Pr(1 + n_1 + n_2 + \dots + n_i) \\ = P^{1+n_1+\dots+n_{i-1}}(1 - P^{n_i}). \end{aligned} \tag{5d}$$

If all $(n_1 + n_2 + \dots + n_q)$ repeats are in error, the receive buffer overflows, $S - 1$ blocks are discarded and n_q additional repeats are sent. Hence,

$$\begin{aligned} \Pr(1 + n_1 + \dots + n_q + n_q + S - 1) \\ = P^{1+n_1+\dots+n_{q-1}+n_q}(1 - P^{n_q}). \end{aligned} \tag{5e}$$

If all n_q additional copies are received in error, $S - 1$ blocks are discarded and n_q additional copies are sent. The probability of this and subsequent cycles can be written directly. Equations (5a)-(5e) can be used to compute the average number of transmissions needed to communicate a block, that is,

$$\begin{aligned} \beta = \sum_{i=0}^q \left(\sum_{j=0}^i n_j \right) P^{\binom{i-1}{\sum_{j=0}^{i-1} n_j}} (1 - P^{n_i}) \\ + \sum_{k=1}^{\infty} \left[k(n_q + S - 1) + \sum_{j=0}^q n_j \right] \\ \cdot (1 - P^{n_q}) P^{\left[\sum_{j=1}^q n_j + (k-1)n_q \right]}. \end{aligned} \tag{6}$$

To simplify the notation in this expression, we have used the symbol n_0 to represent the first term (i.e., 1) in each j -sum. The k -sum can be rearranged to give

$$\sum_{j=0}^q n_j P^{j=0} \sum_{k=1}^{\infty} (P^{n_j})^{k-1} + (n_{q+1} + S - 1) P^{j=0} \sum_{k=1}^{\infty} k (P^{n_j})^{k-1}.$$

Then use of the identities $\sum_{i=0}^{\infty} x^i = (1-x)^{-1}$ and $\sum_{i=1}^{\infty} ix^i = (1-x)^{-2}$ allows us to write this as

$$\sum_{j=0}^q n_j P^{j=0} + \frac{(n_q + S - 1) P^{j=0}}{1 - P^{n_q}}. \tag{7}$$

Throughput can be determined by substituting this expression into (6) and the result into (4). This gives

$$T = 1 / \left[\sum_{i=0}^q \left(\sum_{j=0}^i n_j \right) P^{\binom{i-1}{\sum_{j=0} n_j}} (1 - P^{n_i}) + \sum_{j=0}^q n_j P^{j=0} + \frac{(n_q + S - 1) P^{j=0}}{1 - P^{n_q}} \right]. \tag{8}$$

This result needs some explaining. In practice, q usually equals unity. Here (8) reduces to

$$T = 1 / \left[1 + n_1 P + \frac{(n_1 + S - 1) P^{(1+n_1)}}{1 - P^{n_1}} \right]. \tag{9}$$

For a given channel defined by P and S , the number of repeats n_1 should be selected to maximize T . Although it is not feasible to give a complete solution for n_1 in (9), the following may be useful guidelines for practical ARQ systems. Throughput is maximized for the following choices of n_1 :

$$\begin{aligned} n_1 &= 1, & 0 < SP < 1 \\ n_1 &= 2, & 1 \leq SP \text{ and } 0 \leq SP^2 < 1 \\ n_1 &\geq 3, & 1 \leq SP^2. \end{aligned} \tag{10}$$

The quantity SP is the expected number of errored blocks in the data link, i.e., the number of errors "in flight." For most links, SP is less than one and NACKed blocks should be repeated once. For long-haul links with a high data rate, SP can exceed unity; here blocks should be repeated twice. Only for channels with extremely large storage S or very high error probability will more than two repetitions be necessary.

Equation (9) is plotted in Figs. 2 and 3 for several values of n_1 . In Fig. 2, throughput is plotted versus error probability for a channel with storage $S = 1000$ blocks. In Fig. 3, it is

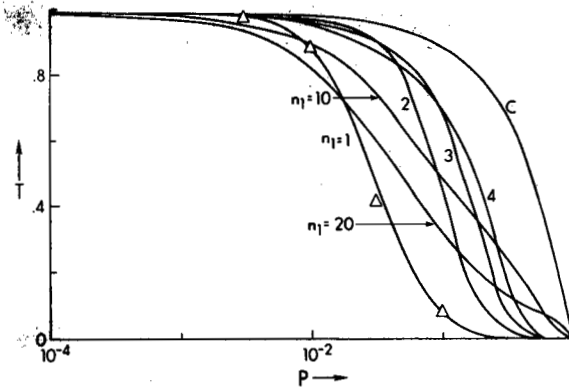


Fig. 2. Plot of throughput (T) versus block error probability (P) for the new strategy with $q = 1$ and $S = 1000$ for several values of n_1 . Results of simulation for $n_1 = 1$ denoted by Δ .

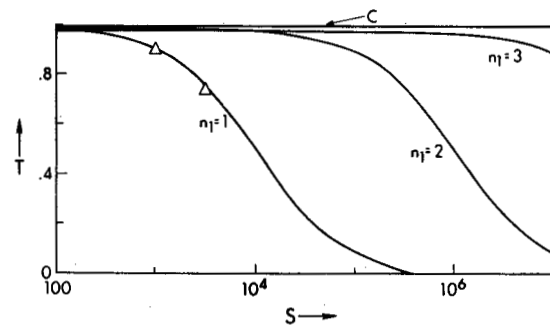


Fig. 3. Plot of throughput (T) versus round-trip channel delay (S) for several continuous-ARQ strategies (independent errors, $P = 10^{-2}$, $q = 1$). Results of simulation for $n_1 = 1$ denoted by Δ .

plotted versus S for $P = 10^{-2}$. Channel capacity (1) is also plotted in both figures.

The curves show several things. First of all, they show that if the number of in-flight errors (SP) is greater than three or so, multiple repeats should be used. They also show that for channels with very high error probability, the number of repetitions which maximizes throughput is significantly greater than one. On the other hand, Fig. 2 shows that regardless of the value of n_1 , throughput is significantly less than channel capacity for $P > 0.1$.

For a given value of q , we will denote by T_q the value of T obtained by maximizing over the set of possible choices for $n = (n_1, n_2, \dots, n_q)$. That is

$$T_q = \max_{\forall n} T. \tag{11}$$

Thus, T_1 is represented by the upper envelope of the curves of T versus P in Fig. 2. Table I shows the values of n_1, n_2 , and n_3 which maximize throughput for $q = 3$ for several large values of P .

The quantity T_1 is less than channel capacity for two reasons, the receiver's fixed buffer size and its limited processing power. It is interesting to ask if removing the first of these constraints, i.e., enlarging the receive buffer, will permit channel capacity to be achieved.

In Fig. 4, we plot T_q versus P for several values of q . These curves indicate that increasing the size of the receive buffer permits channel capacity to be achieved. This can be seen

TABLE I
VALUES OF $n_1, n_2,$ AND n_3 WHICH MAXIMIZE THROUGHPUT FOR $q = 3$ FOR SEVERAL VALUES OF p

p	n_1	n_2	n_3	T
10-0.1	5	8	24	0.123
10-0.2	2	4	13	0.257
10-0.3	2	3	10	0.391
10-0.4	1	2	7	0.517
10-0.5	1	2	6	0.630
10-0.6	1	2	5	0.713
10-0.7	1	1	5	0.780
10-0.8	1	1	4	0.832
10-0.9	1	1	4	0.869
10-1	1	1	3	0.897

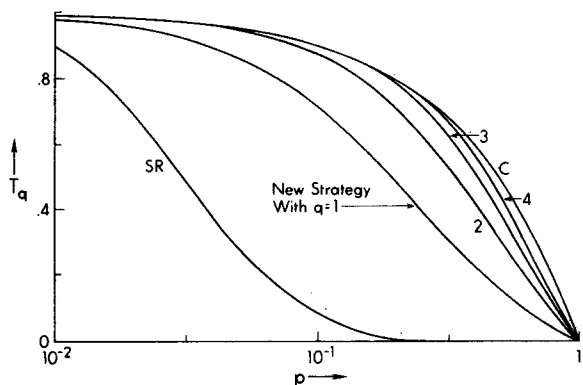


Fig. 4. Plot of throughput (T_q) versus block error probability (P) for several values of buffer size (q).

analytically as well. Setting all $n_i = 1$ in (8) gives

$$T = 1 / \left[\sum_{i=0}^q (i+1)p^i(1-P) + \frac{p^{q+1}(S+(q+1)(1-P))}{1-P} \right] \tag{12}$$

Since the first and second terms in the brackets approach $1 - P$ and 0 , respectively, with increasing q , it follows that

$$\lim_{q \rightarrow \infty} T_q = 1 - P = \text{capacity.} \tag{13}$$

Fig. 4 also shows the rather limited benefits achieved by using a receive buffer with $q > 3$. If a system must operate with $P > 0.1$, it would seem that some form of error correction should be used. For example, the simple FEC procedure considered by Sindhu [26] yields significant increases in throughput with a modest amount of receiver processing.

The new strategy compares favorably with older continuous-ARQ procedures even with the receive buffer size limited to $q = 1$. Fig. 5 shows a plot of throughput versus error probability for this strategy with $q = 1$ and several other strategies. Among these are the following.

Go-back-N: This strategy is a special case of our strategy with $q = 0$ and $n_1 = 1$. For these values (8) yields

$$T_{GBN} = \frac{1-P}{1+(S-1)P} \tag{14}$$

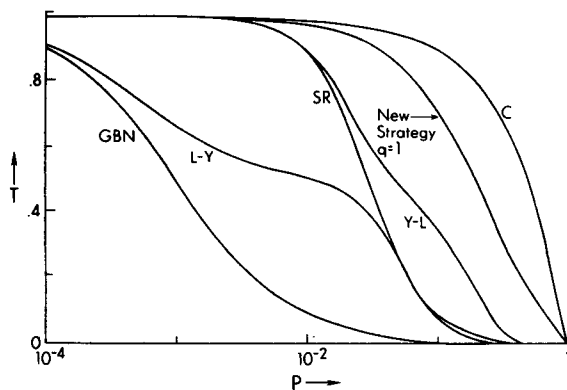


Fig. 5. Plot of throughput (T) versus block-error probability (P) for several continuous-ARQ strategies ($S = 1000, q = 1$ independent errors).

a result previously derived by Sastry [27]. As Fig. 5 shows, this strategy is effective only when $SP \ll 1$, as is usually the case on low-speed and terrestrial data links. Morris' procedure [22] yields performance which is virtually identical to that of the go-back- N strategy.

The *Lin-Yu procedure* [24] (SETRAN) behaves like the go-back- N procedure for small P even though it employs a buffer with $q = 1$. Its throughput is lower bounded by the expression [24]

$$T_{LY} \geq \frac{1 - (1-P)(1-P^2)^{S-1}}{1 - (1-P)(1-P^2)^{S-1} + P^2(S+X)} \tag{15}$$

where

$$X = \frac{P}{(1-P)^2} \left[\frac{1 - (1 - (1-P)^2)^S}{1 - P(1 - (1-P)^2)^{S-1}} \right]$$

As Fig. 5 shows, the throughput of this procedure is significantly inferior to that of even the selective-repeat strategy.

Selective-repeat: This is the new strategy with $q = 1$ and $n_1 = 1$. Here (8) yields

$$T_{SR} = \frac{1-P}{1+(S-1)P^2} \tag{16}$$

The plot in Fig. 5 shows that the procedure is effective provided the number of in-flight errors is less than three or so.

The procedure recently proposed by Yu and Lin [25] is identical to the selective-repeat strategy but allows some processing at the receiver. In particular, it allows copies of blocks which are correctly received when the buffer has overflowed to replace incorrect blocks in the buffer. This additional processing buys some improvement over selective-repeat when the throughput of the latter is small; it does little to extend the useful range of the selective-repeat strategy, however. The throughput of this procedure is lower bounded as follows [25]:

$$T_{YL} \geq \frac{E}{E+F+GS} \tag{17}$$

where

$$E = \frac{B}{1-C} (1 - BC^{S-1})$$

$$F = (1-P)^2 [A^{S-2} + PB^{S-2} + P^2 C^{S-2}]$$

$$G = 3 - (1-P)^2 A^{S-2} - A^2 B^{S-2} - B^2 C^{S-2}$$

and where $A = 1 - P^2$, $B = 1 - P^3$ and $C = 1 - P^4$.

Equation (17) is plotted in Fig. 5 as curve $Y - L$.

The final curve (C) plotted in Fig. 5 is the channel capacity given by (1). The curve shows that for interesting values of throughput the selective-repeat strategy is inferior to capacity by about a factor of ten in error probability. (This factor varies with channel storage S , of course.) The new strategy with $q = 1$ recovers most of this loss and is inferior to capacity by less than a factor of three.

IV. PERFORMANCE OF ARQ STRATEGIES WHEN ERRORS OCCUR IN BURSTS

The effect of channel memory (i.e., error burstiness) on the performance of ARQ strategies has been considered by a number of authors. Blank and Trafton [28] define a burst-error model and analyze the performance of a simple delayless ARQ system. Kanal and Sastry [29] present a number of different channel models which can be used for analyzing ARQ systems subject to burst errors. In this section, we use one of these models, the two-state Gilbert model [30], to derive an approximate expression for the throughput of our selective-repeat strategy when the channel is subject to long-term error dependence or burstiness.

Fig. 6 depicts the Gilbert model. In the good state, errors do not occur; the link spends a fraction δ of the time in this state. In the bad state, errors occur independently with probability $P_B = P/(1 - \delta)$; the link is in this state $1 - \delta$ of the time. With this definition the average error probability remains P . We call the parameter δ the channel *burstiness*. When $\delta = 0$, we have the case considered previously; as δ increases toward its upper limit of $1 - P$, errors become increasingly clustered.

Calculating throughput exactly under these assumptions is difficult; the fact that the link can change state during error recovery complicates things. To avoid this complexity, we will assume that the average state occupancy time is considerably larger than the round-trip delay. Thus we do not take into account the "fine structure" of the bursty channel. In this case, throughput can be approximated by the weighted average of the throughputs of the two states separately. That is,

$$T \cong \delta T_G + (1 - \delta) T_B \tag{18}$$

where T_G and T_B are the values of throughput achieved by the strategy in question in the good and bad states, respectively. By hypothesis, the good state is error-free; hence $T_G = 1$ for any continuous ARQ scheme considered here.

Equation [18] can be used to approximate throughput for any of the strategies considered in Section III. For example, for the new strategy with $q = 1$, T_B is given by (9) with P re-

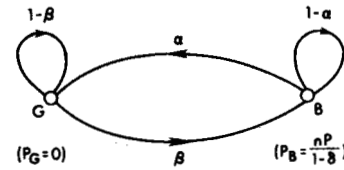


Fig. 6. Gilbert model of data link.

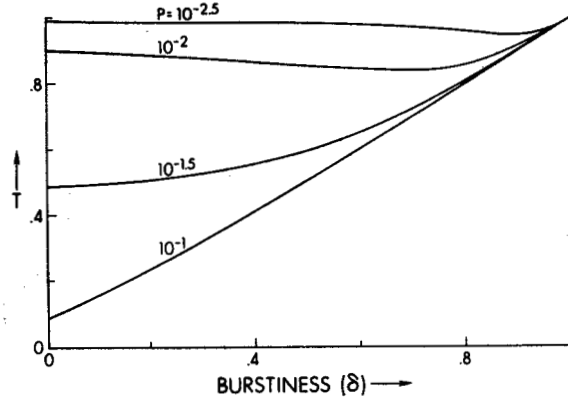


Fig. 7. Plot of throughput (T) versus burstiness (δ) for the new strategy with $q = 1$ and $n_1 = 1$. Channel has $S = 1000$.

placed by $P/(1 - \delta)$. That is,

$$T \cong \delta + (1 - \delta) \left[\frac{1 + n_1 \left(\frac{P}{1 - \delta} \right) + \frac{(n_1 + S - 1) \left(\frac{P}{1 - \delta} \right)^{1+n_1}}{1 - \left(\frac{P}{1 - \delta} \right)^{n_1}}}{1 - \left(\frac{P}{1 - \delta} \right)^{n_1}} \right] \tag{19}$$

This result is plotted in Fig. 7 for $n_1 = 1$, $S = 1000$, and $P = 10^{-1}$, $10^{-1.5}$, 10^{-2} , and $10^{-2.5}$. The curves show that for links in the normal operating range (T slightly less than unity) error burstiness has little or no effect. For poor channels ($T < 0.8$ or so) burstiness improves performance while for marginal channels ($0.8 \leq T \leq 0.9$) burstiness lowers throughput.

Generally speaking, error burstiness increases channel capacity. However, as pointed out by Kanal and Sastry [29], throughput does not necessarily increase with capacity; rather, it reflects a system's ability to exploit capacity. Thus, in Fig. 7 for the channel with $P = 10^{-2}$, increasing burstiness up to $\delta = 0.7$ or so causes more buffer overflows (high overhead) and fewer normal retransmissions (low overhead). The net effect is a decrease in throughput.

The primary conclusion that can be drawn from this simple analysis is that, provided error burstiness is a long-term phenomenon relative to system response time, it does not have a significant impact on channels with reasonably high throughput (≥ 0.8 or so). Under such conditions, the independent-error model can be used with some confidence in predicting

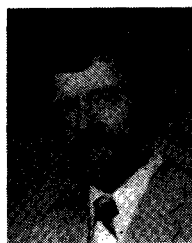
the performance of data links regardless of their error burstiness.

ACKNOWLEDGMENT

The author would like to thank Dr. W. Yumin of the Northwest Telecommunication Engineering Institute of the People's Republic of China for several helpful discussions.

REFERENCES

- [1] S. S. L. Chang, "Theory of information feedback systems," *IRE Trans. Inform. Theory*, vol. IT-2, no. 2, 1956.
- [2] B. A. Harris, A. Hauptschein, K. C. Morgan, and L. S. Schwartz, "Binary communications feedback systems," *AIEE Trans. Commun. Electron.*, vol. 77, pp. 960-969, 1959.
- [3] J. J. Metzner and K. C. Morgan, "Coded feedback communication systems," presented at the Nat. Electron. Conf., Chicago, IL, 1960.
- [4] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Prob. and Stat.*, vol. 1, pp. 611-644, Univ. California Press, Berkeley, CA, 1961.
- [5] E. J. Weldon, Jr., "Asymptotic error coding bounds for the binary symmetric channel with feedback," Ph.D. dissertation, Univ. Florida, Gainesville, 1963.
- [6] B. Reiffen, W. G. Schmidt, and H. L. Yudkin, "The design of an error-free data transmission system for telephone circuits," *AIEE Trans. Commun. Electron.*, vol. 80, pp. 224-231, 1961.
- [7] W. R. Cowell and H. O. Burton, "Computer simulation of the use of a group code with retransmission on a Gilbert burst channel," *AIEE Trans. Commun. Electron.*, pt. 1, pp. 577-580, Jan. 1962.
- [8] R. D. Stuart, "An insert system for use with feedback communication links," *IEEE Trans. Commun. Syst.*, vol. CS-11, pp. 142-143, 1963.
- [9] R. J. Benice and A. H. Frey, Jr., "An analysis of retransmission systems," *IEEE Trans. Commun. Syst.*, vol. CS-12, pp. 135-144, 1964.
- [10] M. D. Balkovic and P. E. Muench, "Effect of propagation delay, caused by satellite circuits, on data communication systems that use block retransmission for error correction," in *Conf. Rec. 1969 Int. Conf. Commun.*, Boulder, CO, 1969, pp. 29-31-29-36.
- [11] "General information: Binary synchronous communication," IBM Publ. GA27-3004, IBM Corp., 1969.
- [12] E. R. Cacciamani and K. Kim, "Circumventing the problem of propagation delay on satellite data channels," *Data Commun.*, July/Aug. 1975.
- [13] J. R. Kersey, "Synchronous data link control," *Data Commun.*, May/June 1974.
- [14] "Advanced data communication control procedures (ADCCP)," Amer. Nat. Standards Inst., Washington, DC, 1977.
- [15] A. K. Kaul, "Performance of high-level data link control in satellite communications," *Comsat Tech. Rev.*, vol. 8, pp. 41-87, 1978.
- [16] K. C. Traynham and R. F. Steen, "Interpreting SDLC throughput efficiency," *Data Commun.*, Oct./Nov. 1977.
- [17] D. R. Doll, "Calculating throughput of full-duplex data link controls," *Data Commun.*, Jan., Feb. 1976.
- [18] M. C. Easton, "Batch throughput efficiency of ADCCP/HDLC/SDLC selective reject protocols," *IEEE Trans. Commun.*, vol. COM-28, pp. 187-195, Feb. 1980.
- [19] J. A. Lockitt, A. G. Gatfield, and T. R. Dobyns, "A selective-repeat ARQ system," presented at the 3rd Int. Conf. Digital Satellite Commun., Kyoto, Japan, 1975.
- [20] A. Gatfield, "Error-control on satellite channels using ARQ techniques," *Comsat Tech. Rev.*, vol. 6, pp. 179-188, Jan. 1976.
- [21] A. R. K. Sastry, "Improving repeat-request (ARQ) performance on satellite channels under high error rate conditions," *IEEE Trans. Commun.*, vol. COM-23, pp. 436-439, Apr. 1975.
- [22] J. M. Morris, "On another go-back-N ARQ technique for high error rate conditions," *IEEE Trans. Commun.*, vol. COM-26, pp. 187-189, Jan. 1978.
- [23] —, "Optimal block lengths for ARQ error control schemes," *IEEE Trans. Commun.*, vol. COM-27, pp. 488-493, Feb. 1979.
- [24] S. Lin and P. S. Yu, "An effective error control scheme for satellite communications," *IEEE Trans. Commun.*, vol. COM-28, pp. 395-401, Mar. 1980.
- [25] P. S. Yu and S. Lin, "An efficient selective repeat ARQ scheme for satellite channels and its throughput analysis," *IEEE Trans. Commun.*, vol. COM-29, pp. 353-363, Mar. 1981.
- [26] P. S. Sindhu, "Retransmission error control with memory," *IEEE Trans. Commun.*, vol. COM-25, pp. 473-479, May 1977.
- [27] A. R. K. Sastry, "Retransmission error control options for dedicated access and packet switched random access satellite channels," in *Conf. Rec. 4th Int. Conf. Digital Satellite Commun.*, Montreal, P.Q., Canada, Oct. 1978, pp. 275-281.
- [28] H. A. Blank and P. J. Trafton, "ARQ analysis in channels with memory," in *Conf. Rec. Nat. Telecommun. Conf.*, Dec. 1975, pp. 39-7-39-12.
- [29] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their application to error control," *Proc. IEEE*, pp. 726-746, July 1978.
- [30] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, p. 1253, 1960.



E. J. Weldon, Jr. (S'61-M'64) was born in New York, NY, on April 8, 1938. He received the B.E.E. degree from Manhattan College, Bronx, NY, in 1958, and the M.S. and Ph.D. degrees in electrical engineering from the University of Florida, Gainesville, in 1960 and 1963, respectively.

From 1963 to 1966, he was employed by Bell Laboratories, Holmdel, NJ, where he was engaged in the analysis and design of data communication systems. In 1966, he joined the faculty of the University of Hawaii, Honolulu, where he is currently Professor of Electrical Engineering. During the summers of 1977 and 1978, he was again employed by Bell Laboratories where he worked on the design of ACS, a nationwide packet switching network. He teaches a popular three-day short course, "Error Correcting Codes with Applications to Data Storage and Communication Systems." He consults on the design and implementation of systems using error-correcting codes and holds a number of patents. He is the founder of ADTECH, Inc., a Honolulu firm specializing in the design and development of advanced electronic systems. His research interests include error-correcting codes and computer-communication networks. He has coauthored two books: *Principles of Data Communication* (New York: McGraw-Hill, 1968, with R. W. Lucky and J. Salz) and *Error-Correcting Codes*, 2nd ed. (Cambridge, MA: MIT Press, 1972, with W. W. Peterson).