A CONCATENATED CODING SCHEME FOR ERROR CONTROL

Technical Report

to

NASA
Goddard Space Flight Center
Greenbelt, Maryland

Grant Number NAG 5-407

Shu Lin
Principal Investigator
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822

July 9, 1985

A CONCATENATED CODING SCHEME FOR ERROR CONTROL*

ABSTRACT

In this paper, a concatenated coding scheme for error control in data communications is presented and analyzed. In this scheme, the inner code is used for both error correction and detection, however the outer code is used only for error detection. A retransmission is requested if either the inner code decoder fails to make a successful decoding or the outer code decoder detects the presence of errors after the inner code decoding. Probability of undetected error (or decoding error) of the proposed scheme is derived. An efficient method for computing this probability is presented. Throughput efficiency of the proposed error control scheme incorporated with a selective-repeat ARQ retransmission strategy is also analyzed. Three specific examples are presented. One of the examples has been adopted for error control in NASA telecommand system.

*This research is supported by NASA Grant No. NAG 5-407.

1. Introduction

Consider a concatenated coding scheme [1] for error control for a binary symmetric channel with bit-error-rate $\varepsilon \leq 1/2$ as shown in Figure 1. Two linear block codes, C_1 and C_2, are used. The inner code C_1 is an (n_1,k_1) code with minimum distance d_1. The inner code is designed to correct t or fewer errors and simultaneously detect $\lambda(\lambda \geq t)$ or fewer errors where $t+\lambda+1 \leq d_1$. The outer code C_2 is an (n_2,k_2) code with minimum distance d_2 and

$$n_2 = mk_1$$

where m is a positive integer. The outer code is designed for error detection only.

The encoding is done in two stages. A message of k_2 bits is first encoded into a codeword of n_2 bits in the outer code C_2. Then the n_2-bit word is divided into m k_1-bit segments. Each k_1-bit segment is then encoded into an n_1-bit codeword in the inner code C_1. This n_1-bit word is called a frame. Thus, corresponding to each k_2-bit message at the input of the outer code encoder, the output of the inner code encoder is a sequence of m frames. This sequence of m frames is called a block. A two dimensional block format is depicted in Figure 2.

The decoding consists of error correction in frames and error detection in m decoded k_1-bit segments. When a frame in a block is received, it is decoded based on the inner code C_1. If the decoding is successful, the n_1-k_1 parity bits are then removed from the decoded frame, and the k_1-bit decoded segment is stored in a buffer. If there are t or fewer transmission errors in a received frame, the errors will be corrected and the decoded segment is error free. If t+1 or more transmission errors are detected in a received frame, then the entire block which contains the

erroneous frame is discarded, and the receiver requests a retransmission of the block. If there are more than λ errors in a received frame, the errors may result in a syndrome which corresponds to a correctable error pattern with t or fewer errors. In this case, the decoding will be successful, but the decoded frame (or segment) contains undetected errors. If m frames of a received block have been successfully decoded, the receiver buffer contains m k_1-bit decoded segments. Then error detection is performed on these m decoded segments based on the outer code C_2. If no error is detected, the m decoded segments are assumed to be error free and are accepted (with n_2-k_2 parity bits removed) by the receiver. If the presence of errors is detected, then the m decoded segments are discarded and the receiver requests a retransmission of the rejected block. Retransmission and decoding process continue until the block is successfully received. Note that a successfully received block may be either error free or contains undetectable errors.

The error control scheme described above is actually a combination of forward-error-correction (FEC) and automatic-repeat-request (ARQ), called a hybrid ARQ scheme [2]. The retransmission strategy determines the system throughput, it may be one of the three basic modes namely, stop-and-wait, go-back-N or selective-repeat. In this paper, we analyze the performance of the proposed error control scheme in terms of the reliability and throughput efficiency. The reliability is measured in terms of the probability of undetected error after decoding. The probability of undetected error is derived, and an efficient method for computing this probability is presented. The throughput efficiency depends on the mode of retransmission. In this paper, we analyze the throughput efficiency of the proposed error control scheme incorporated with a selective-repeat ARQ with a finite receiver buffer.

Three specific example schemes are considered. The first two example schemes use the same inner code which is a distance-4 shortened Hamming code with generator polynomial [2],

$$\bar{g}_1^{(1)}(X) = \bar{g}_1^{(2)}(X) = (X+1)(X^6+X+1) = X^7+X^6+X^2+1 \ . \tag{1}$$

In the first scheme, the outer code is a distance-4 shortened Hamming code with generator polynomial,

$$\begin{aligned}\bar{g}_2^{(1)}(X) &= (X+1)(X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+X+1) \\ &= X^{16}+X^{12}+X^5+1 \end{aligned} \tag{2}$$

which is the X.25 standard for packet-switched data network [3]. This example scheme is proposed and adopted for error control on NASA telecommand links. In the second example scheme, the outer code is a shortened Reed-Solomon (RS) code [2,4,5] with symbols from the Galois field $GF(2^8)$ and generator polynomial,

$$\bar{g}_2^{(2)}(X) = (X+1)(X+\alpha) \ , \tag{3}$$

where α is a primitive element in $GF(2^8)$. In the third example scheme, the inner code is a shortened version of the extended double-error-correcting (63,51) BCH code with generator polynomial [2],

$$g_1^{(3)}(X) = X^{12}+X^{10}+X^8+X^5+X^4+X^3+1 \ , \tag{4}$$

The outer code is the same as that of the first example scheme.

The probabilities of a decoding error and throughput efficiencies for the three example schemes are computed. We show that they all provide very high reliability even for a very high bit-error rate. All three example schemes also provide high throughput efficiency.

2. Probabilities of Incorrect Decoding and Retransmission

Let V_ℓ denote the set of all binary vectors of length ℓ. Let n be a positive integer defined as follows:

$$n = mn_1 \tag{5}$$

where n_1 is the length of the inner code and m is the number of frames per block. Let

$$\bar{u} = (u_1, u_2, \ldots, u_n) \tag{6}$$

be a vector in V_n. The n_1-tuple

$$\bar{u}_h = (u_{(h-1)n_1+1}, u_{(h-1)n_1+2}, \ldots, u_{hn_1}) \tag{7}$$

is called the h-th frame of $\bar{u}$ for $1 \leq h \leq m$. Hence, we can represent the vector $\bar{u}$ by its frames as follows:

$$\bar{u} = (\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m) \ . \tag{8}$$

The first k_1 components of the h-th frame of $\bar{u}$ are said to form the h-th segment of $\bar{u}$. The mk_1-tuple obtained by concatenating the m segments of $\bar{u}$ in order is called the projection of $\bar{u}$, denoted $\rho(\bar{u})$.

Let C denote the overall code obtained by concatenating the inner code C_1 and the outer code C_2. Then C is a binary (n,k_2) linear code where $n=mn_1$. A binary vector $\bar{u}$ in V_n is a codeword in C if and only if

(1) each frame of $\bar{u}$ is a codeword in the inner code C_1, and

(2) the projection $\rho(\bar{u})$ of $\bar{u}$ is a codeword in the outer code C_2.

Let P_c be the probability of a correct decoding using the concatenated coding scheme described in Section 1. For a binary symmetric channel with bit-error-rate $\varepsilon \leq 1/2$, P_c is equal to the probability that there are t or fewer errors in each frame of the received block, and is given by

$$P_c = \left[\sum_{i=0}^{t} \binom{n_1}{i} \varepsilon^i (1-\varepsilon)^{n_1-i} \right]^m . \tag{9}$$

Let P_d be the probability of a successful decoding (either correct or incorrect) of a received block. Then P_d is the probability that, for a channel-error pattern $\bar{e}$, there is a codeword $\bar{v}$ in the overall code C such that, for $1 \leq h \leq m$, the Hamming distance between the h-th frame of $\bar{e}$ and the h-th

frame of $\bar{v}$ is t or less. If P_d can be computed, then the probability of an incorrect decoding (or a decoding error), P_e, is given by

$$P_e = P_d - P_c \ , \tag{10}$$

and the probability of a decoding failure, P_r, is given by

$$P_r = 1 - P_d \ . \tag{11}$$

Note that P_r is also the probability of a retransmission.

In the following we will derive an expression for P_d. Let $W_{j,s}^{(i)}$ denote the number of binary vectors of weight j in V_{n_1} which are at a (Hamming) distance s from a given binary vector of weight i in V_{n_1}. Let $\bar{a} = (a_1, a_2, \ldots, a_{n_1})$ and $\bar{b} = (b_1, b_2, \ldots, b_{n_1})$ be two binary vectors in V_{n_1} with weights i and j respectively. Let q be the number of suffices ℓ's such that $a_\ell = 0$ and $b_\ell = 1$. Then $\bar{b}$ is at a distance s from $\bar{a}$ if and only if

$$q = (j+s-i)/2 \ . \tag{12}$$

Therefore we have that

$$W_{j,s}^{(i)} = \binom{n_1-i}{q}\binom{i}{j-q} = \binom{n_1-i}{(j+s-i)/2}\binom{i}{(j+i-s)/2} \ . \tag{13}$$

The generating function for $W_{j,s}^{(i)}$ is

$$\sum_{j=0}^{n_1}\sum_{s=0}^{n_1} W_{j,s}^{(i)} X^j Y^s = (1+XY)^{n_1-i}(X+Y)^i \ , \tag{14}$$

which was derived by MacWilliams in 1963 [6].

Let $\bar{e} = (\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_m)$ be a channel error pattern, and let j_h be the weight of the h-th frame $\bar{e}_h$ for $1 \le h \le m$. The occurrence probability of $\bar{e}$ is

$$P(\bar{e}) = \prod_{h=1}^{m} \varepsilon^{j_h}(1-\varepsilon)^{n_1-j_h} \tag{15}$$

Suppose that there is a codeword $\bar{v} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_m)$ in C such that $\bar{e}_h$ is at a distance $s_h \le t$ from $\bar{v}_h$ for $1 \le h \le m$. Since $\bar{v}_h$ is a codeword in the inner code C_1 for $1 \le h \le m$ and the minimum distance of C_1 is assumed to be greater than 2t, such a codeword $\bar{v}$ in C is uniquely determined. Conversely, for a codeword

$\bar{v} = (\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_m)$ in C whose weight in the h-th frame is i_h for $1 \leq h \leq m$, there are

$$\prod_{h=1}^{m} W_{j_h, s_h}^{(i_h)} \tag{16}$$

error patterns $(\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_m)$'s such that the weight of $\bar{e}_h$ is j_h and $\bar{e}_h$ is at a distance $s_h \leq t$ from $\bar{v}_h$ for $1 \leq h \leq m$. Let $A_{i_1, i_2, \ldots, i_m}$ denote the number of codewords in C whose weight in the h-th frame is i_h for $1 \leq h \leq m$. Then, the probability of a successful decoding is given below:

$$P_d = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} A_{i_1, i_2, \ldots, i_m} \left\{ \prod_{h=1}^{m} \left[\sum_{j_h=0}^{n_1} \sum_{s_h=0}^{t} W_{j_h, s_h}^{(i_h)} \varepsilon^{j_h} (1-\varepsilon)^{n_1 - j_h} \right] \right\}. \tag{17}$$

Hence, if we know the detail weight distribution $\{A_{i_1, i_2, \ldots, i_m} : 0 \leq i_h \leq n_1$ for $1 \leq h \leq m\}$ of C, we can compute the probability P_d of a successful decoding from (17). Then, from (9) and (10) we can compute error probability P_e. From (11), we can compute the retransmission probability P_r, from which we can determine the system throughput for a given retransmission strategy (ARQ scheme).

The dimension of C is k_2. In general, k_2 is large and it is not feasible to compute the detail weight distribution, $A_{i_1, i_2, \ldots, i_m}$, directly from C by generating all the codewords of C. In the next section, we will express P_d in terms of the detail weight distribution of the dual code of C by using the generalized MacWilliams's identity [7, p. 147].

3. Evaluation of the Probability of Decoding

In this section, we will derive a number of results which will facilitate the computation of the probability P_d of a successful decoding. Let $C^{\perp}$ denote the dual code of the overall code C. Let $B_{i_1, i_2, \ldots, i_m}$ be the number

of codewords in $C^{\perp}$ for which the weight of h-th frame is i_h for $0 \le i_h \le n_1$ and $1 \le h \le m$. Then

$$\{B_{i_1,i_2,\ldots,i_m} : \ 0 \le i_h \le n_1 \ \text{ with } 1 \le h \le m\} \tag{18}$$

represents the detail weight distribution of $C^{\perp}$. In the following, the first lemma gives the decoding probability P_d in terms of the detail weight structure of $C^{\perp}$.

Lemma 1: Let $P_s(\cdot,\cdot)$ be a Krawtchouk polynomial [7, p. 129]. Then

$$P_d = 2^{-(n-k_2)} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m} \left\{ \prod_{h=1}^{m} \left[(1-2\varepsilon)^{i_h} \sum_{s=0}^{t} P_s(i_h,n_1) \right] \right\}. \tag{19}$$

Proof: See Appendix A. ΔΔ

It follows from (19) that, for the worst channel condition $\varepsilon = 1/2$,

$$P_d = 2^{-(n-k_2)}(n_1+1)^m . \tag{20}$$

Let r_1 and r_2 denote the numbers of parity check bits of the inner and outer codes respectively. Then $r_1 = n_1 - k_1$ and $r_2 = n_2 - k_2$. The dimension of $C^{\perp}$ is thus $mr_1 + r_2$. In general $mr_1 + r_2$ is also large, and it is still not feasible to compute $B_{i_1,i_2,\ldots,i_m}$ by generating all the codewords of $C^{\perp}$. However, the computing time can be reduced considerably by using Lemma 2 given below.

For a vector $\bar{v} = (v_1, v_2, \ldots, v_{mk_1})$ in V_{mk_1}, the k_1-tuple $(v_{(h-1)k_1+1}, v_{(h-1)k_1+2}, \ldots, v_{hk_1})$ is called the h-th segment of $\bar{v}$ for $1 \le h \le m$. Let $\bar{v}^{(0)}$ (all-zero vector), $\bar{v}^{(1)}, \ldots, \bar{v}^{(2^{r_2}-1)}$ denote all the codewords of $C_2^{\perp}$, the dual code of the outer code C_2. For $0 \le j < 2^{r_2}$ and $1 \le h \le m$, let $\bar{v}_h^{(j)}$ denote the h-th segment of $\bar{v}^{(j)}$. Let $C_1^{\perp}$ be the dual code of the inner code C_1. Let $\bar{u} = (u_1, u_2, \ldots, u_{k_1})$ be a binary vector in V_{k_1}. Let

$$\bar{u}_0 = (u_1, u_2, \ldots, u_{k_1}, 0, 0, \ldots, 0)$$

be a vector obtained by appending n_1-k_1 zeros to $\bar{u}$. Hence $\bar{u}_0$ is a vector in V_{n_1}. For $0\le i\le n_1$, let $B_i^{(1)}(\bar{u})$ denote the number of vectors of weight i in the coset of $C_1^{\perp}$ which contains the vector $\bar{u}_0$. Lemma 2 gives $B_{i_1,i_2,\ldots,i_m}$ in terms of $B_i^{(1)}(\bar{u})$.

Lemma 2:

$$B_{i_1,i_2,\ldots,i_m} = \sum_{j=0}^{2^{r_2}-1} \left\{ \prod_{h=1}^{m} B_{i_h}^{(1)}(\bar{v}_h^{(j)}) \right\}. \tag{21}$$

Proof: See Appendix B. ΔΔ

Since the dimensions r_1 and r_2 of $C_1^{\perp}$ and $C_2^{\perp}$ are much smaller than the dimension mr_1+r_2 of $C^{\perp}$, it is much easier to compute $B_{i_1,i_2,\ldots,i_m}$ from (21) than from $C^{\perp}$ directly by generating all the codewords of $C^{\perp}$. This will be shown in the next section.

It follows from Lemmas 1 and 2 that we have Theorem 1.

Theorem 1:

$$P_d = 2^{-(n-k_2)} \sum_{j=0}^{2^{r_2}-1} \prod_{h=1}^{m} \left[\sum_{i=0}^{n_1} B_i^{(1)}(\bar{v}_h^{(j)}) (1-2\varepsilon)^i \sum_{s=0}^{t} P_s(i,n_1) \right]. \tag{22}$$

Proof: (See Appendix C).

It follows from (9), (10), and (22) that we can obtain the probability P_e of a decoding error for the concatenated coding scheme proposed in Section 1.

4. Time Complexity for Computing the Probability of Decoding

In this section, we will evaluate the time complexity for computing the decoding probability P_d given by (22). The number of different $\bar{v}_h^{(j)}$'s with $1\le h\le m$ and $0\le j<2^{r_2}$ is at most $m2^{r_2}$. Hence the computing time for finding

$$\{B_i^{(1)}(\bar{v}_h^{(j)}) : 0\le i\le n_1,\ 0\le j<2^{r_2} \text{ and } 1\le h\le m\}$$

is roughly proportional to $mn_1 2^{r_1+r_2}$. Furthermore we need multiplications and additions of order $(n_1+m)2^{r_2}$ for computing P_d. The dominant order of computation for P_d is $mn_1 2^{r_1+r_2}$.

Next we assume that the outer code C_2 is a shortened (or the full) code of a quasi-cyclic code $C_{2,f}$ of length n_f [5] which is invariant under every cyclic shift by s places, where s divides k_1 (note that, if s=1, $C_{2,f}^{\perp}$ is a cyclic code). Then the dual code of $C_{2,f}$ denoted $C_{2,f}^{\perp}$, is also quasi-cyclic for every s-place shift. The codewords in $C_{2,f}$ can be partitioned into equivalent classes, each equivalent class consists of a codeword $\bar{v}$ and the codewords obtained by cyclically shifting $\bar{v}$ s places at a time. We may choose a codeword in each equivalent class as its representative. A vector $(u_1,u_2,\ldots,u_{n_2})$ in V_{n_2} is a codeword in $C_2^{\perp}$ if and only if there is a representative codeword $(v_1,v_2,\ldots,v_{n_f})$ for an equivalent class of $C_{2,f}^{\perp}$, for which there exists a positive integer j such that $u_i = v_{js+i}$ for $1 \le i \le n_2$ where js+i is taken modulo n_f. Hence the number of different $\bar{v}_h^{(j)}$'s with $1 \le h \le m$ and $0 \le j < 2^{r_2}$ is at most 2^{r_2}. Since $n_1 < 2^{r_1}$, the dominate order of computation for P_d is

$$\max\{n_1 2^{r_1+r_2},\ m2^{r_2}\} .$$

However, if we compute

$$\{B_{i_1,i_2,\ldots,i_m}:\ 0 \le i_h \le n_1 \text{ and } 1 \le h \le m\}$$

directly from $C^{\perp}$, the computation time is proportional to $mn_1 2^{mr_1+r_2}$ which is much greater than $n_1 2^{r_1+r_2}$ and $m2^{r_2}$. Hence, using expression (22) for computing P_d reduces the computing time considerably.

Suppose that $C_1^{\perp}$ contains the all-one vector, (1,1,...,1). Let $C_{1,0}^{\perp}$ denote the (n_1,r_1-1) linear subcode of $C_1^{\perp}$ which does not contain the all-one vector. For a vector $\bar{u} = (u_1,u_2,\ldots,u_{k_1})$ in V_{k_1}, let $B_i^{(1),0}(\bar{u})$ be the number of vectors of weight i in the coset of $C_{1,0}^{\perp}$ which contains the vector $\bar{u}_0 = (u_1,u_2,\ldots,u_{k_1},0,0,\ldots,0)$ in V_{n_1}. Then

$$B_i^{(1)}(\bar{u}) = B_i^{(1),0}(\bar{u}) + B_{n_1-i}^{(1),0}(\bar{u}) . \tag{23}$$

The above relation reduces the computing time for $B_i^{(1)}(\bar{u})$.

If $C_2^{\perp}$ is generated by an (n_2,r_2-1) linear subcode $C_{2,0}^{\perp}$ and the all-one vector $(1,1,\ldots,1)$ in V_{n_2}, the computing time for $B_i^{(1)}(\bar{u})$ can be further reduced by using the following relation:

$$B_i^{(1)}(\bar{u}+(1,1,\ldots,1)) = \sum_{j=0}^{i} [B_{k_1-j,i-j}^{(1),0}(\bar{u}) + B_{j,r_1-i+j}^{(1),0}(\bar{u})] \tag{24}$$

where $B_{i,j}^{(1),0}(\bar{u})$ denote the number of vectors with weight i in the first k_1 bit positions and weight j in the last r_1 bit positions, which are in the coset of $C_{1,0}^{\perp}$ containing the vector $(\bar{u},0,0,\ldots,0)$ in V_{n_1}.

5. Example Schemes

In this section we consider three examples of the concatenated coding scheme described in Section 1. For each of the example schemes, the probability of a decoding error, P_e, and the probability of a retransmission, P_r, are computed for various bit-error rates.

Example Scheme I

In this example scheme, the inner code C_1 is a shortened distance-4 Hamming code with generator polynomial,

$$\bar{g}_1^{(1)}(X) = (X+1)(X^6+X+1) = X^7+X^6+X^2+1 \tag{25}$$

where X^6+X+1 is a primitive polynomial of degree 6. The full length code generated by $\bar{g}_1(X)$ of (25) is a (63,56) cyclic Hamming code. The 56 information bits form 7 8-bit information bytes. If ℓ information bits are deleted from the full length code, then the inner code C_1 becomes a $(63-\ell,56-\ell)$ shortened cyclic code [2,4,5]. In practical applications, ℓ is generally chosen to make $k_1=56-\ell$ as a multiple of 8-bit byte. The inner code C_1 is used for single error correction (i.e., t=1). It is also capable of detecting all the error patterns of double and odd number errors [2,4,5,7].

The outer code is also a shortened distance-4 Hamming code with generator polynomial,

$$\bar{g}_2^{(1)}(X) = (X+1)(X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+X+1) \quad . \tag{26}$$
$$= X^{16}+X^{12}+X^5+1 \ .$$

where $X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+X+1$ is a primitive polynomial of degree 15. This code is the X.25 standard for packet-switched data networks [3]. The natural length of this code is $2^{15}-1 = 32,767$. But the maximum length of C_2 being considered is 3,584 bits. We assume that the number m of frames in a block varies from 3 to a maximum 64. We also assume that the number of information bytes contained in a frame varies 1 to 7, i.e. $k_1 = 8 \sim 56$ bits. Hence the length n_2 of the outer code varies from 3 to 448 bytes or from 24 to 3,584 bits. The 16 parity bits of the outer code is used for error detection only. The error detection performance of this outer code for various lengths has been investigated recently by Fujiwara, et al. [8].

Example scheme 1 has been adopted by NASA for error control on telecommand links. The probability of a decoding error, P_e, for this scheme is shown in figures 3, 5, 7 and 9 for various k_1, m and bit-error-rate ε. We see from Figure 9 that, as the bit-error-rate ε increases, P_e increases to a peak value and then decreases to the value $P_e^* = 1-P_d^*$ where P_d^* is given by (20). We see that the scheme provides very high reliability even for very high bit-error-rate. The probability of a decoding failure (or retransmission), P_r is shown in Figures 4, 6, 8 and 10 for various k_1, m and bit-error-rate ε.

Example Scheme II

In this example scheme, the inner code C_1 is the same as that of example scheme I. The outer code is a shortened Reed-Solomon (RS) code with symbols from the Galois field $GF(2^8)$ and generator polynomial

$$\bar{g}^{(2)}(X) = (X+1)(X+\alpha) \ , \tag{27}$$

where α is a primitive element of $Gf(2^8)$ and a root of $X^8+X^4+X^3+X^2+1$. This code is used as a binary code with each code symbol represented by a 8-bit

byte. This binary RS is quasi-cyclic and has a minimum distance 4. The natural length of the code is 255 bytes or 2040 bits.

The probability P_e of a decoding error is shown in Figures 3, 5, 7 and 9. The performance of this example scheme is inferior to example scheme 1, however still provides very high reliability. The probability P_r of a decoding failure is shown in Figures 4, 6, 8 and 10. Since P_e is very small, it follows from (10) and (11) that $P_r \approx 1-P_c$. Hence example schemes 1 and 2 have almost the same P_r.

Example Scheme III

In this example scheme, the outer code C_2 is the same as that of the example scheme I, the X.25 standard code. However, the inner code is a shortened version of the extended (with an overall parity bit appended) double-error-correcting (63,51) BCH code with generator polynomial [2],

$$\bar{g}_1^{(3)}(X) = X^{12}+X^{10}+X^8+X^5+X^4+X^3+1 \ . \qquad (28)$$

The inner code C_1 generated by $g_1^{(3)}(X)$ has minimum distance 6 and is used for correcting all the double and single errors (t=2). The code is capable of detecting all the triple errors and many other errors.

The probability of a decoding error, P_e is shown in Figures 3, 5, 7, and 9. Since the inner code is designed for double error correction, the performance of this example scheme is superior to the example schemes 1 and 2. The probability of a decoding failure is shown in Figures 4, 6, 8 and 10.

Now we consider the accuracy of computation for probabilities P_c, P_d, and P_e of the above example schemes. If the wordlength of the computer under consideration is at least $r_1 = n_1-k_1$, then the exact value of $B_i^{(1)}(\bar{v}_h^{(j)})$ can be computed. Let w be the number of bits in the mantissa of the floating point number of the computer. Then the number of significant bits of the result computed for P_c by using (9) is no less than

$$\lfloor w - \log_2(t+1)mn_1 \rfloor .$$

The number of significant bits of the result computed for P_d by using (22) is no less than

$$\lambda = \lfloor w - \log_2(2^{r_2}mn_1^2) \rfloor .$$

If the computational result for P_e by using (10) is greater than

$$2^{-(\lambda-\delta)}$$

for a positive integer δ, then the number of significant bits of the results is greater than δ. In our computation, we used FORTRAN 77 on ACOS-1000 whose number of bits in the mantissa of the quadruple precision floating point number is 124. For m=4 and n_1=31, if the computational result for P_e is greater than 10^{-24} or for m=64 and n =63, if the computational result for P_e is greater than 10^{-22}, then the number of significant (decimal) digits is greater than 3.

6. Throughput Efficiency

The error control scheme presented in this paper is actually a hybrid ARQ scheme [2], which is a combination of forward-error-correction (FEC) and automatic-repeat-request (ARQ). The throughput efficiency of the scheme depends on the mode of retransmission. There are three basic modes of retransmission: namely stop-and-wait ARQ, go-back-N ARQ and selective-repeat ARQ [2]. In a stop-and-wait ARQ system, the transmitter sends a block to the receiver and waits for an acknowledgment. A positive acknowledgement (ACK) from the receiver signals that the block has been successfully decoded and accepted, and the transmitter then sends the next block. A negative acknowledgment (NAK) (or no acknowledgment) from the receiver indicates that the received block has been detected in error; the transmitter resends the block. Stop-and-wait ARQ is very simple to implement, however it is inherently inefficient because of the idle time spent waiting for an acknowledgment.

In a go-back-N ARQ, the transmitter continuously transmits blocks in order and then stores them pending receipt of an ACK/NAK for each. The acknowledgment of a block arrives after a round-trip (propagation) delay, defined as the time interval between the transmission of a block and the receipt of an acknowledgment for the block. During this interval, N-1 other blocks are also sent. Whenever the transmitter receives a NAK for a particular block, say block i, it stops transmitting new blocks. Then it goes back to block i and proceeds to retransmit block i and the N-1 succeeding blocks which were transmitted during one round-trip delay. At the receiving end, the receiver discards the erroneously received block i and all N-1 subsequently received blocks whether they are error-free or not. Retranmission continues until block i is successfully decoded. In each retransmission for block i, the transmitter resends the same sequence of blocks. As soon as block i is positively acknowledged, the transmitter proceeds to send new blocks. Clearly, go-back-N ARQ is more efficient than the stop-and-wait ARQ. However it still has a severe drawback. When the receiver fails to decode a block, it also rejects the next N-1 received blocks, even though many of them may be error free.

In a selective-repeat ARQ system, blocks are also transmitted continuously. However, the transmitter only resends those blocks that are negatively acknowledged (NAK'ed). After resending a NAK'ed block, the transmitter continues sending new blocks. Clearly, selective-repeat ARQ is superior to the other two ARQ schemes. However, with selective-repeat ARQ, a buffer must be provided at the receiver to store the successfully decoded blocks following a decoding failure, because ordinarily, blocks must be delivered to the destination in correct order, e.g. in point-to-point communications. Sufficient buffer (theoretically infinite buffer) must be provided at the receiver, otherwise, buffer overflow may occur and blocks may be lost.

Stop-and-wait and go-back-N ARQ's provide satisfactory throughput efficiency for data communication systems with low or moderate data rate and short round-trip delay. For systems with high data rate and long round-trip delay, e.g. satellite links, the throughput efficiency of stop-and-wait and go-back-N ARQ's becomes unacceptable, and selective-repeat ARQ must be used.

The throughput efficiency η of a data communication system is defined as the ratio of average number of message (or information) bits successfully accepted by the receiver per unit of time to the total number of bits that could be transmitted per unit of time. Suppose that an ideal selective-repeat ARQ (with infinite receiver buffer) is incorporated in the error control scheme proposed in this paper. Then the throughput efficiency of the scheme is

$$\eta_{SR} = \left(\frac{k_2}{n}\right)(1 - P_r) \tag{29}$$

where k_2/n is the rate of the overall concatenated code C and P_r is the retransmission probability given by (11). Using the value of P_r given in Figure 4, 6, 8 and 10, we can compute η_{SR} for various bit-error-rate and m.

In practice, only finite buffer can be provided at the receiver. In this case, buffer overflow may occur in a selective-repeat ARQ scheme, this reduces the throughput efficiency. However, if a sufficiently long buffer is used and if buffer overflow is properly handled, even with a reduction in throughput, selective-repeat ARQ still significantly outperforms the other two ARQ schemes. Practical schemes have been devised for handling buffer overflow [9-12]. One such scheme is the selective-repeat plus go-back-N (SR+GBN) ARQ devised by Miller and Lin [11]. With SR+GBN ARQ scheme, retransmission of an erroneous block, say block i, is first carried out in selective-repeat mode. If block i fails to be successfully decoded at the ν-th retransmission ($\nu \geq 1$),

the transmitter switches to the go-back-N mode. That is, it sends no more new blocks but backs up to block i and resends that block and the N-1 succeeding blocks that were transmitted after the ν-th retransmission of block i. Retransmission in go-back-N mode continues until block i is successfully decoded, the transmitter is then switched back to selective-repeat mode. With SR+GBN mode, buffer overflow is prevented if the receiver buffer is capable of storing $(N-1)\nu+1$ decoded blocks. If the SR+GBN ARQ is incorporated in the proposed error control scheme, the throughput efficiency is then

$$\eta_{SR+GBN} = \frac{1 - P_r}{1+(N-1)P_r^{\nu+1}} \left(\frac{k_2}{n}\right) . \tag{30}$$

We see from (30) that, for large ν, the throughput performance of the SR+GBN ARQ approaches the throughput performance of an ideal selective-repeat ARQ. For many data communication systems where bit-error-rate ε is not very high, SR+GBN ARQ with ν=1 or 2 would provide very good throughput efficiency. Consider a satellite communications system with a data rate 1.54 Mbps and a round-trip delay of 700 ms. Throughput efficiencies of the three example schemes with SR+GBN ARQ are shown in Figures 11, 12 and 13 for various m, k_1, ε and ν. We see that all three example schemes with SR+GBN ARQ provide good throughput efficiency.

7. Conclusion

In this paper, a concatenated coding scheme for error control is presented. The probability of a decoding error for this scheme is derived for a binary symmetric channel. An efficient method for computing this error probability is presented. Three specific examples are analyzed, and their probabilities of a decoding error for various bit-error-rates are computed. All the example schemes provide very high reliability even for very high

bit-error-rate ε. For bit-error-rate $\varepsilon=10^{-4}$, a probability of decoding error in the order less than 10^{-16} is achieved. The first example scheme is proposed and has been adopted by NASA for error control in NASA Telecommand System. The proposed error control scheme also provides high throughput performance if a proper retransmission scheme is used.

APPENDIX A

Proof of Lemma 1

It follow from (14) that

$$\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots,i_m}\prod_{h=1}^{m}\left[\sum_{j_h=0}^{n_1}\sum_{s_h=0}^{n_1} W_{j_h,s_h}^{(i_h)} X_h^{j_h} Y_h^{s_h}\right]$$

$$= \sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots,i_m}\prod_{h=1}^{m}(1+X_hY_h)^{n_1-i_h}(X_h+Y_h)^{i_h} . \qquad (A\text{-}1)$$

By generalized MacWilliams' Theorem [7, p. 147] we obtain the following identity,

$$\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots,i_m}\prod_{h=1}^{m} X_h^{n_1-i_h} Y_h^{i_h}$$

$$= 2^{-(n-k_2)}\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m}\prod_{h=1}^{m}(X_h+Y_h)^{n_1-i_h}(X_h-Y_h)^{i_h} \qquad (A\text{-}2)$$

The right-hand side of (A-1) can be rewritten as

$$2^{-(n-k_2)}\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m}$$

$$\cdot\prod_{h=1}^{m}(1+X_h)^{n_1-i_h}(1-X_h)^{i_h}(1+Y_h)^{n_1-i_h}(1-Y_h)^{i_h} . \qquad (A\text{-}3)$$

Let $P_s(\cdot,\cdot)$ be the Krawtchouk polynomial [7, p. 129]. Since $(1+Y)^{n_1-i}(1-Y)^{i} = \sum_{s=0}^{n_1} P_s(i,n_1)Y^s$ [7, p. 130], it follows from (A-1) and (A-3) that

$$\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots,i_m}\prod_{h=1}^{m}\left[\sum_{j_h=0}^{n_1}\sum_{s_h=0}^{n_1} W_{j_h,s_h}^{(i_h)} X_h^{j_h} Y_h^{s_h}\right]$$

$$= 2^{-(n-k_2)}\sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_1}\cdots\sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m}$$

$$\cdot\prod_{h=1}^{m}\left[(1+X_h)^{n_1-i_h}(1-X_h)^{i_h}\sum_{s_h=0}^{n_1} P_{s_h}(i_h,n_1)Y_h^{s_h}\right] . \qquad (A\text{-}4)$$

Taking the terms on both sides of (A-4) for which the degree of Y_h is t or less for $1 \le h \le m$ and substituting one for Y_h with $1 \le h \le m$, we have that

$$\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots,i_m} \prod_{h=1}^{m} \left[\sum_{j_h=0}^{n_1} \sum_{s_h=0}^{t} W_{j_h,s_h}^{(i_h)} X_h^{j_h} \right]$$

$$= 2^{-(n-k_2)} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m}$$

$$\cdot \prod_{h=1}^{m} \left[(1+X_h)^{n_1-i_h} (1-X_h)^{i_h} \sum_{s_h=0}^{t} P_{s_h}(i_h,n_1) \right] . \qquad \text{(A-5)}$$

Substituting $\varepsilon/(1-\varepsilon)$ for X_h on both sides of (A-5) for $1 \le h \le m$, we obtain

$$\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} A_{i_1,i_2,\ldots i_m} \prod_{h=1}^{m} \left[\sum_{j_h=0}^{n_1} \sum_{s_h=0}^{t} W_{j_h,s_h}^{(i_h)} \varepsilon^{j_h} (1-\varepsilon)^{n_1-j_h} \right]$$

$$= 2^{-(n-k_2)} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m} \prod_{h=1}^{m} \left[(1-2\varepsilon)^{i_h} \sum_{s=0}^{t} P_s(i_h,n_1) \right]$$

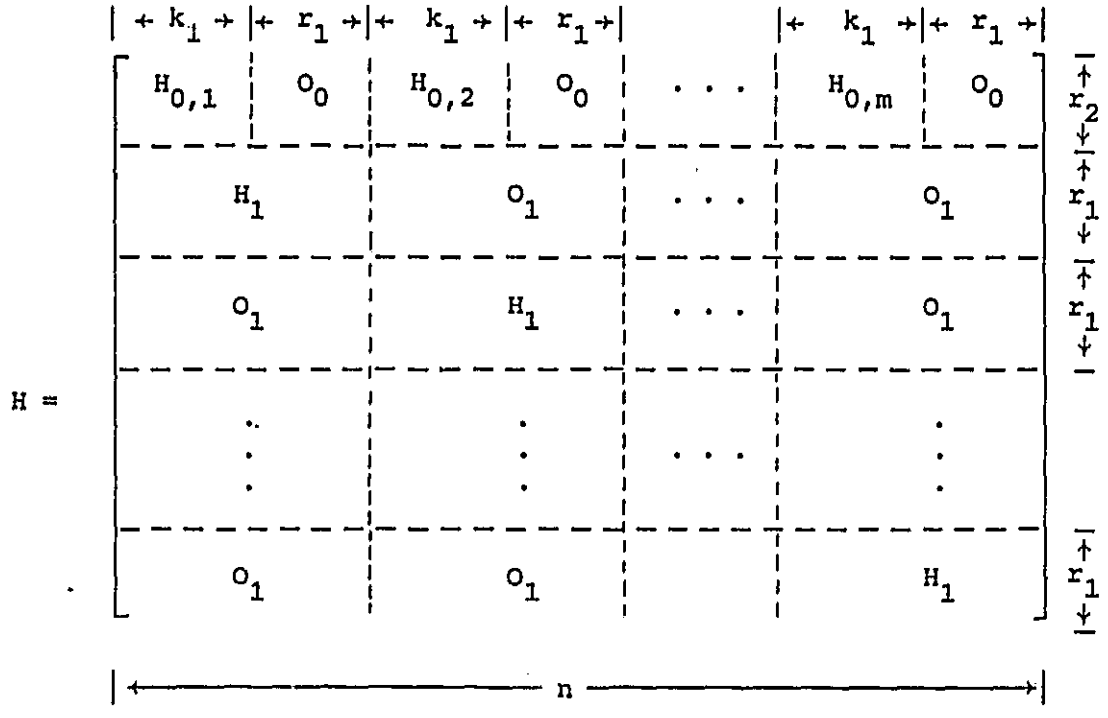
(A-6)

It follows from (17) and (A-6) that we obtain (19).

APPENDIX B

Proof of Lemma 2

Consider a parity-check matrix H of the overall concatenated code C. H has the following form



where (1) $[H_{0,1} \quad H_{0,2} \quad \cdots \quad H_{0,m}]$ is a parity-check matrix of the outer code C_2;

(2) H_1 is a parity-check matrix of the inner code C_1;

(3) O_0 is a $r_2 \times r_1$ zero matrix;

(4) O_1 is a $r_1 \times n_1$ zero matrix.

Let $C^{\perp}_{2,ex}$ be the code generated by the first r_2 rows of the matrix H. For a codeword $\bar{v}$ in $C^{\perp}_{2,ex}$, its projection $\rho(\bar{v})$ is a codeword in $C^{\perp}_2$, and the components in $\bar{v}$ but not in $\rho(\bar{v})$ are zeros. $C^{\perp}$ is the direct sum of $C^{\perp}_{2,ex}$ and $C^{\perp}_1 \times C^{\perp}_1 \times \ldots \times C^{\perp}_1$, the m-th direct-product of $C^{\perp}_1$. Then, (21) follows directly from the definition of $B^{(1)}_i(\bar{u})$.

APPENDIX C

Proof of Theorem 1

It follows from (19) and (21) that

$$P_d = 2^{-(n-k_2)} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} B_{i_1,i_2,\ldots,i_m} \prod_{h=1}^{m} [(1-2\varepsilon)^{i_h} \sum_{s=0}^{t} P_s(i_h,n_1)]$$

$$= 2^{-(n-k_2)} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} \sum_{j=0}^{2^{r_2}-1} \prod_{h=1}^{m} B_{i_h}^{(1)}(\bar{v}_h^{(j)}) \prod_{h=1}^{m} [1-2\varepsilon)^{i_h} \sum_{s=0}^{t} P_s(i_h,n_1)]$$

$$= 2^{-(n-k_2)} \sum_{j=0}^{2^{r_2}-1} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_1} \cdots \sum_{i_m=0}^{n_1} \prod_{h=1}^{m} [B_{i_h}^{(1)}(\bar{v}_h^{(j)})(1-2\varepsilon)^{i_h} \sum_{s=0}^{t} P_s(i_h,n_1)]$$

$$= 2^{-(n-k_2)} \sum_{j=0}^{2^{r_2}-1} \prod_{h=1}^{m} \left[\sum_{i=1}^{n_1} B_i^{(1)}(\bar{v}_h^{(j)})(1-2\varepsilon)^i \sum_{s=0}^{t} P_s(i_h,n_1) \right]$$

REFERENCES

1. G.D. Forney, Jr., Concatenated Codes, MIT Press, Cambridge, Mass., 1966.

2. S. Lin and D.J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, New Jersey, 1983.

3. CCITT: Recommendation X.25, "Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Terminals Operating in Packet Mode on Public Data Networks," with Plenary Assembly, Doc. No. 7, Geneva, 1980.

4. E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968.

5. W.W. Peterson and E.J. Weldon, Jr., Error-Correcting Codes, Second Edition, Cambridge, Mass., The MIT Press, 1972.

6. J. MacWilliams, "A Theorem on the Distribution of Weights in a Systematic Code," Bell System Technical Journal, Vol. 42, pp. 79-94, 1963.

7. F.J. MacWilliams and N.J.A. Sloane, Theory of Error-Correcting Codes, North Holland Amsterdam, 1977.

8. T. Fujiwara, T. Kasami, A. Kitai and S. Lin, "On the Undetected Error Probability for Shortened Hamming Codes," IEEE Transactions on Communications, Vol. COM-33, No. 6, June, 1985.

9. J.J. Metzner, "A Study of an Efficient Retransmission Strategy for Data Links," NTC '77 Conference Records, pp. 3B:1-1-3B:15.

10. P.S. Yu and S. Lin, "An Efficient Selective-Repeat ARQ Scheme for Satellite Channels and Its Throughput Analysis," IEEE Transactions on Communications, Vol. COM-29, pp. 353-363, March 1981.

11. M.J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," IEEE Transactions on Communications, Vol. COM-29, pp. 1307-1315, September 1981.

12. E.J. Weldon, Jr., "An Improved Selective-Repeat ARQ Strategy," IEEE Transactions on Communications, Vol. COM-30, pp. 480-486, March 1982.

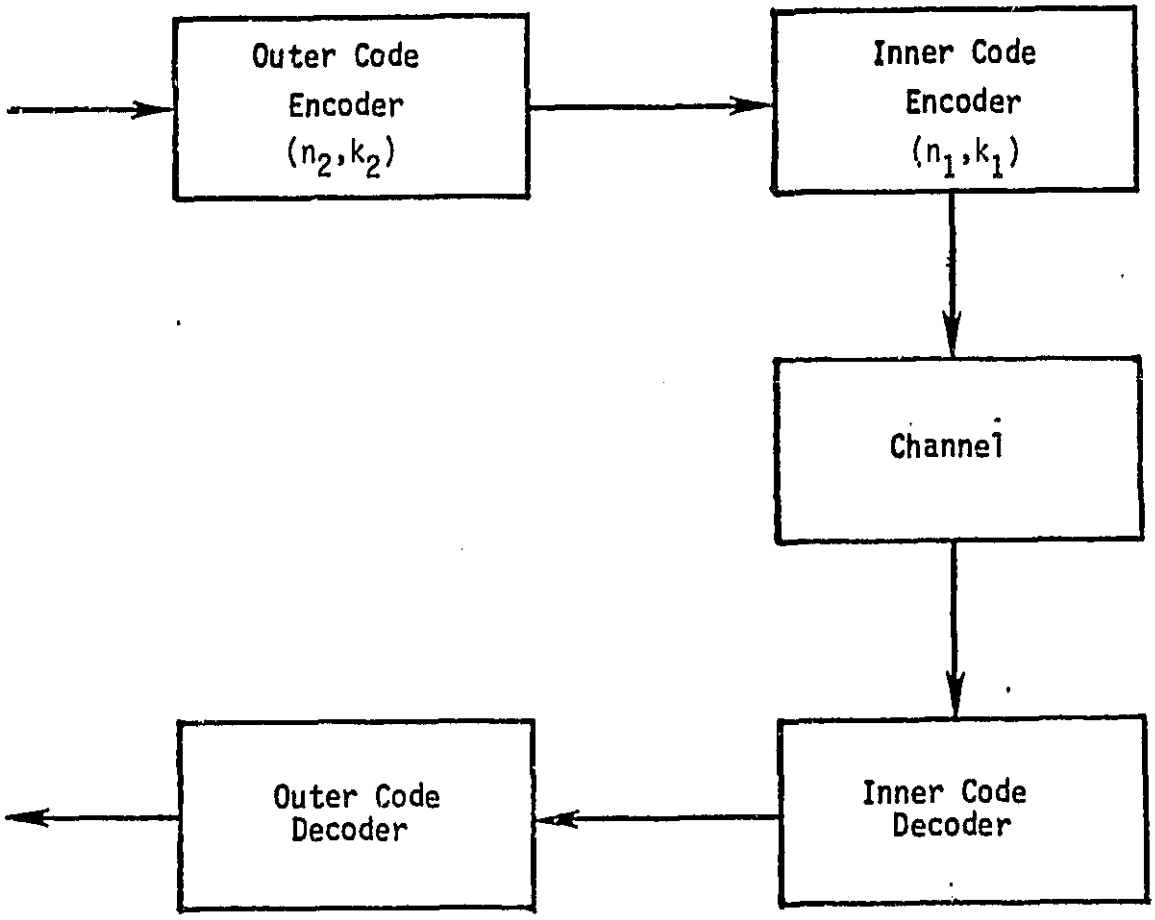


Figure 1 A concatenated coding system

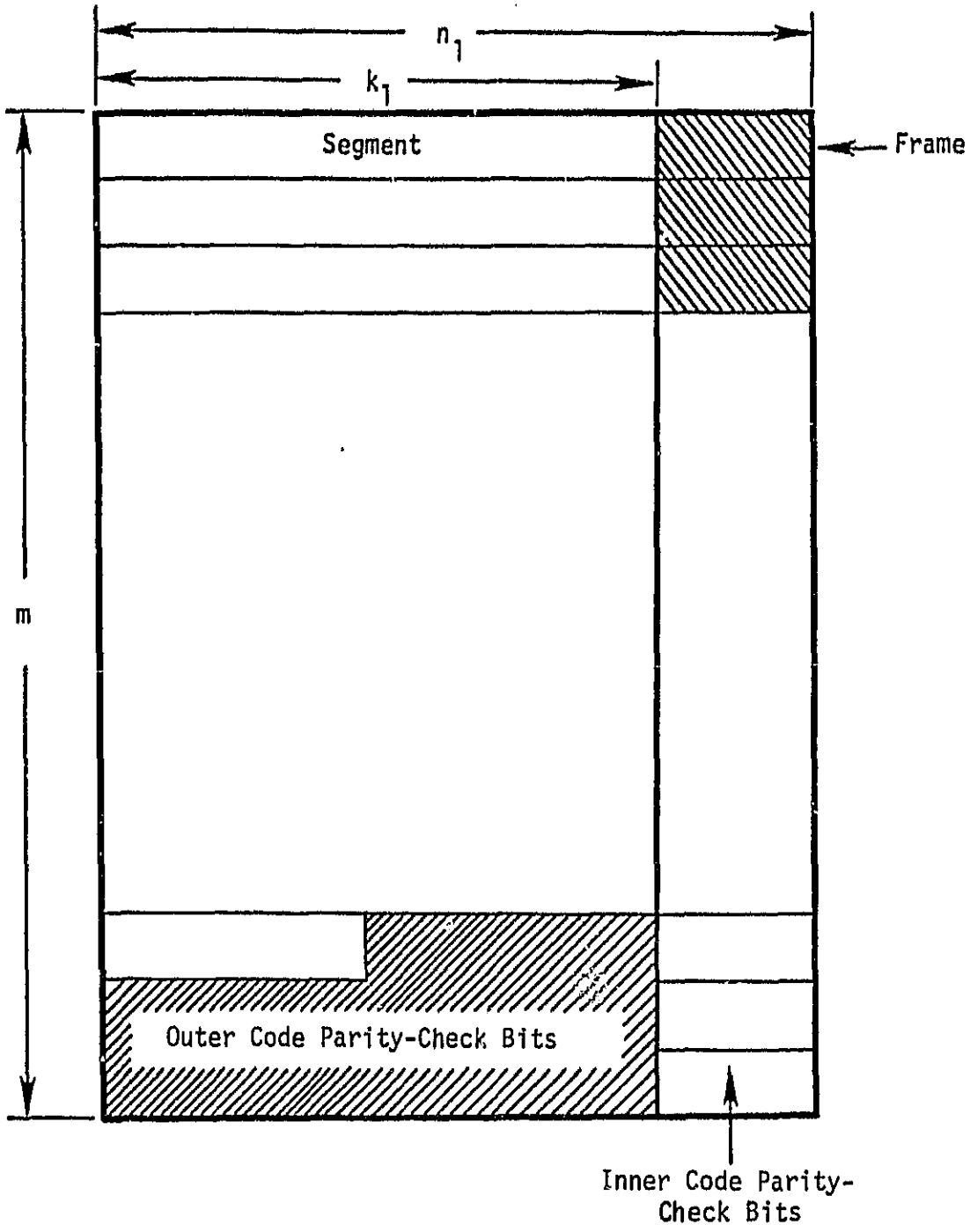


Figure 2 Block format

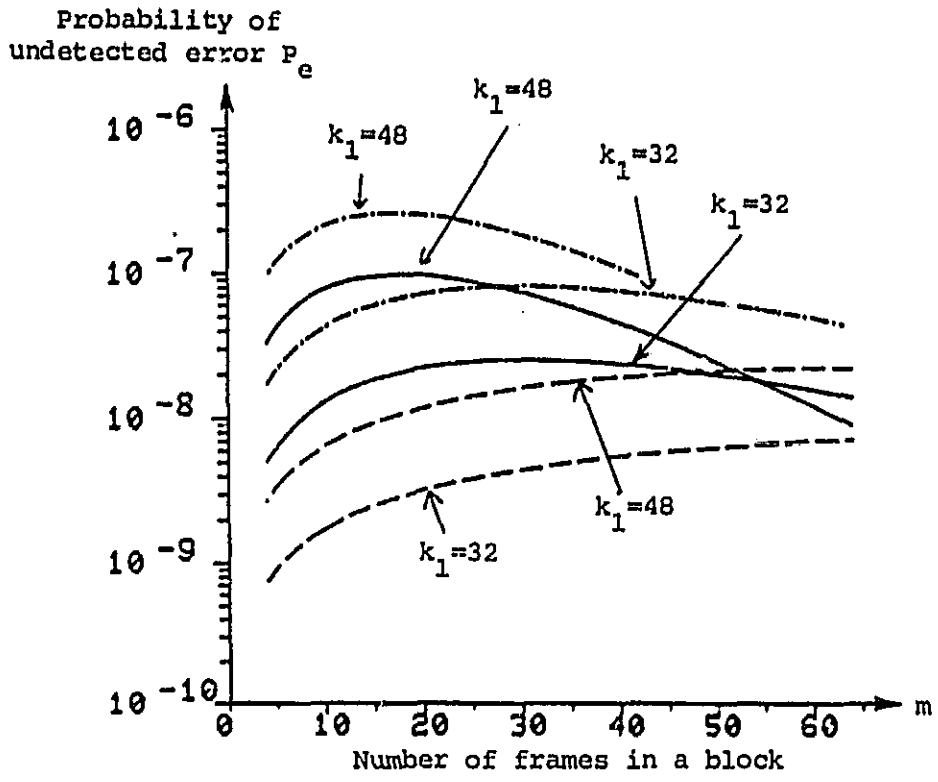


Figure 3 The probability of undetected error P_e for bit error rate $\varepsilon = 10^{-2}$ and the number of information bits in a frame $k_1 = 32$ or 48.

————————: Example scheme 1

——·——·—— Example scheme 2

----------: Example scheme 3

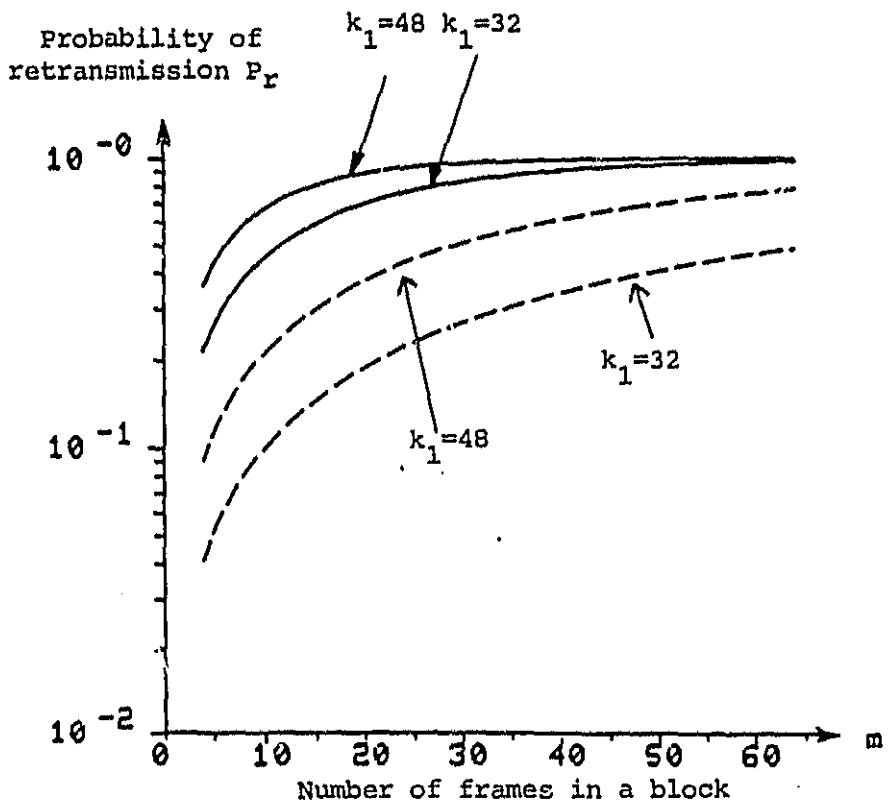


Figure 4 The probability of retransmission P_r for bit error rate $\varepsilon = 10^{-2}$ and the number of information bits in a frame $k_1 = 32$ or 48.

————: Example schemes 1 and 2
----------: Example scheme 3

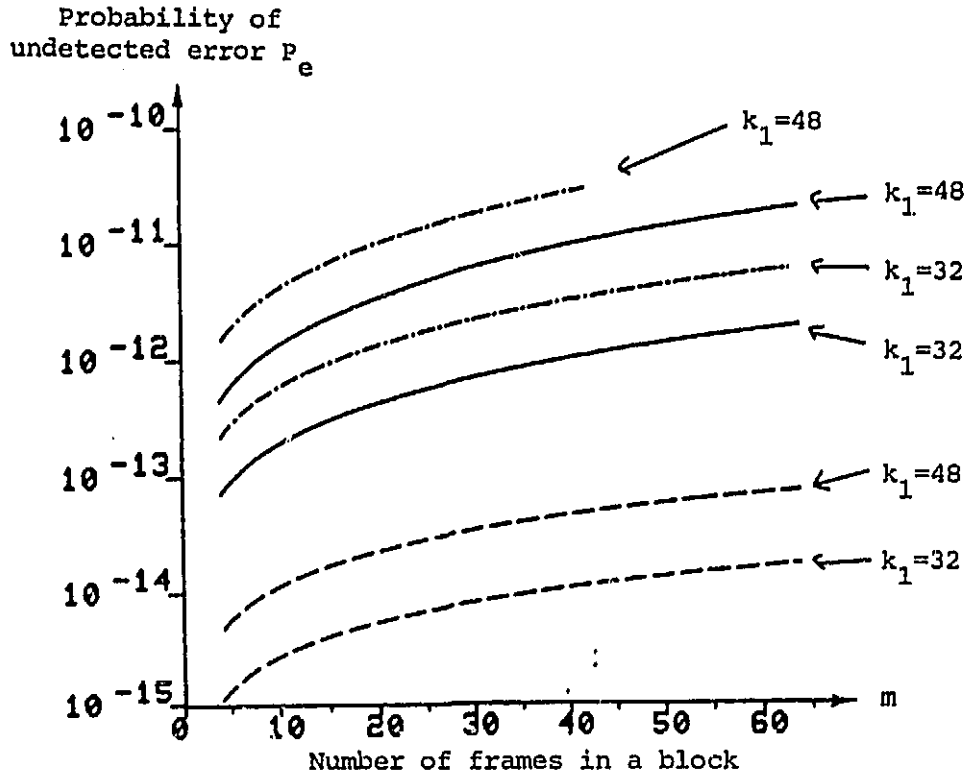


Figure 5 The probability of undetected error P_e for bit error rate $\varepsilon = 10^{-3}$ and the number of information bits in a frame k_1=32 or 48.

————————: Example schemes 1 and 2
----------: Example scheme 3

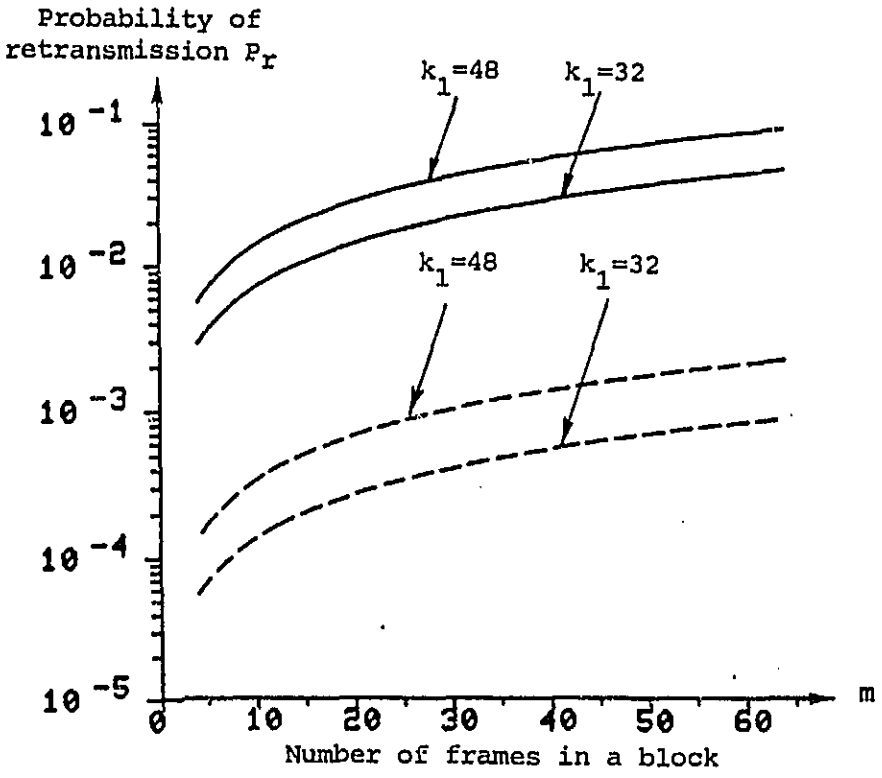


Figure 6 The probability of retransmission P_r for bit error rate $\varepsilon = 10^{-3}$ and the number of information bits in a frame $k_1 = 32$ or 48.

————————: Example schemes 1 and 2
- - - - - - - - - - - -: Example scheme 3

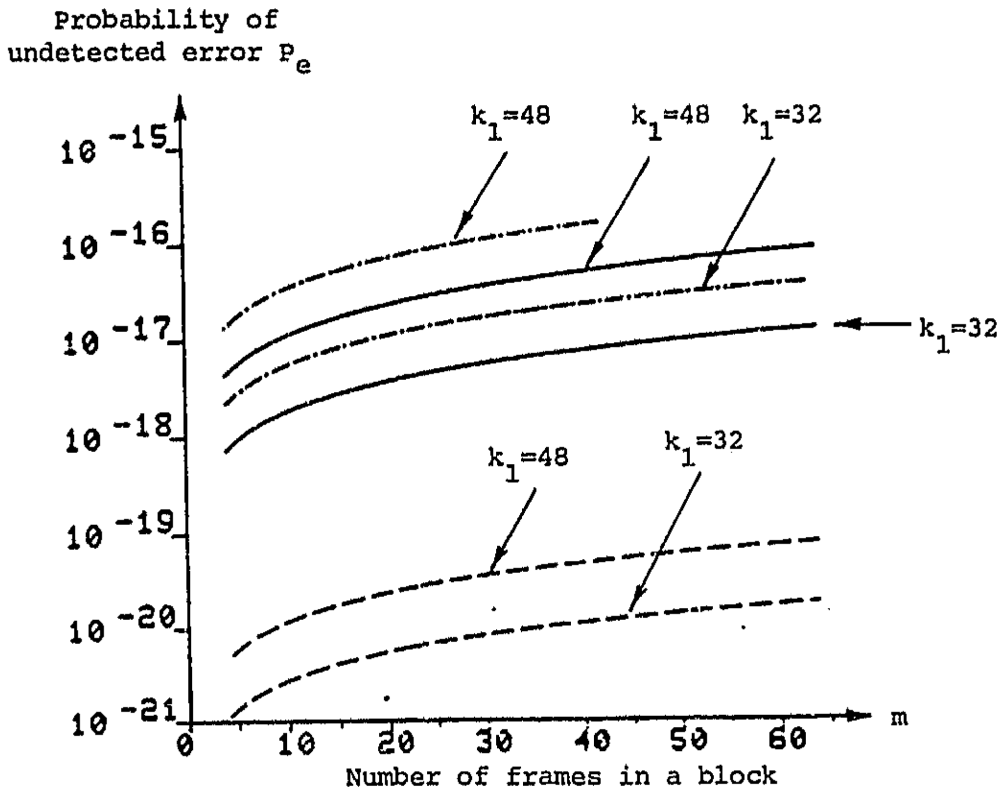


Figure 7 The probability of undetected error P_e for bit error rate $\varepsilon = 10^{-4}$ and the number of information bits in a frame $k_1 = 32$ or 48.

————————: Example scheme 1
—·—·—·—: Example scheme 2
- - - - - - - -: Example scheme 3

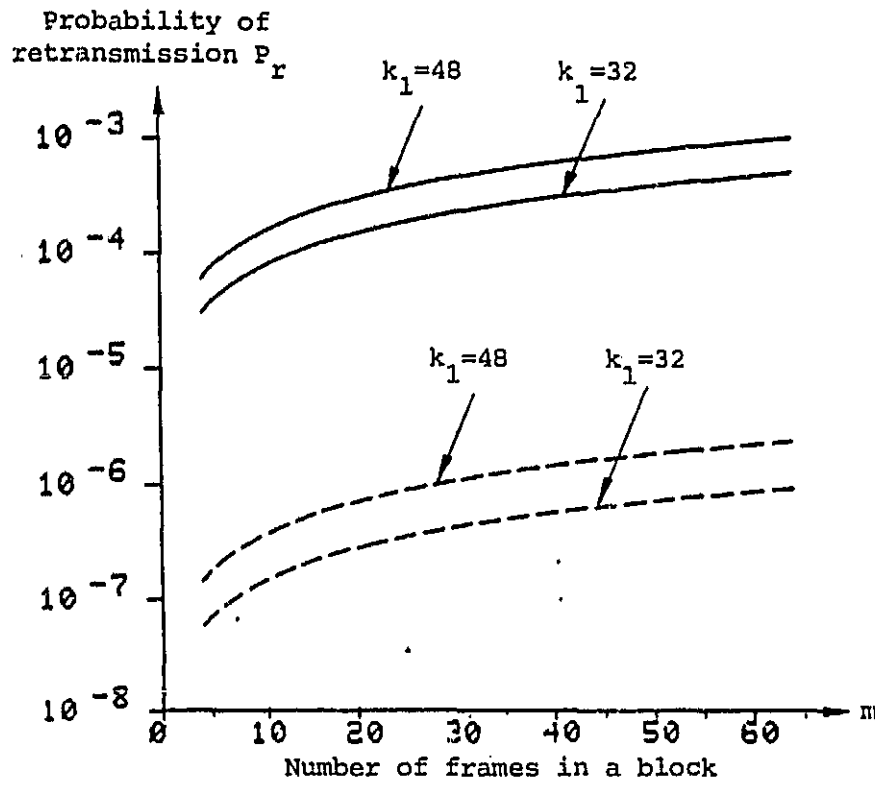


Figure 8 The probability of retransmission P_r for bit error rate $\varepsilon = 10^{-4}$ and the number of information bits in a frame $k_1 = 32$ or 48.

————————: Example schemes 1 and 2
----------: Example scheme 3

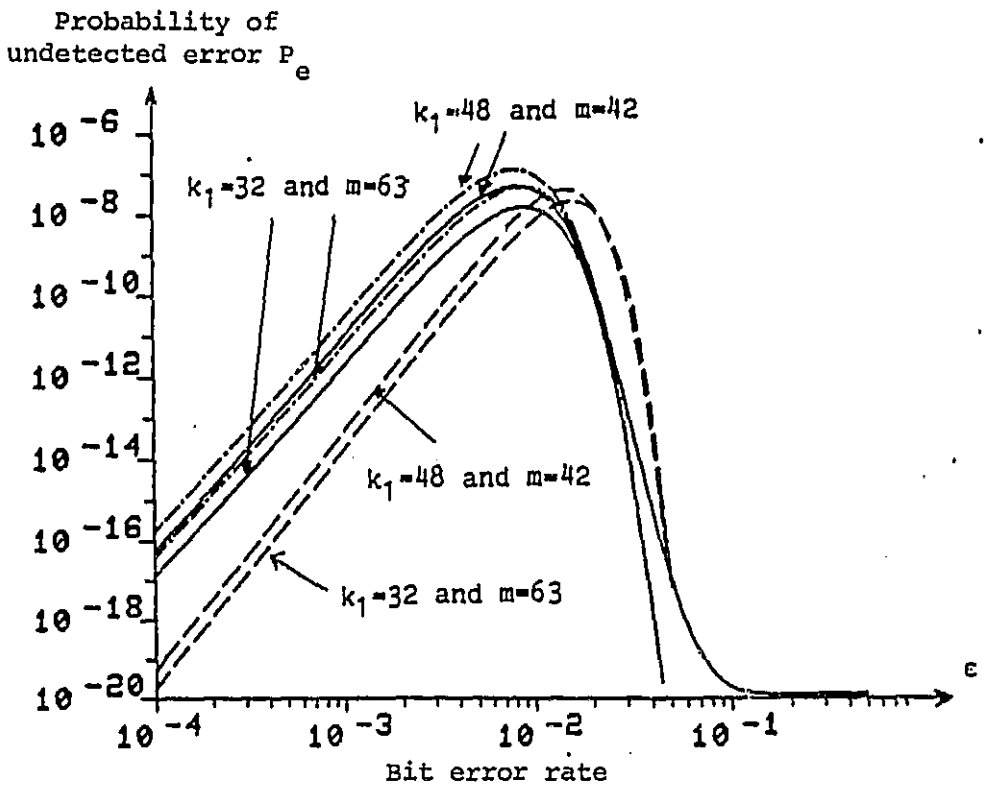


Figure 9 The probability of undetected error P_e for the number of information bits in a frame k_1=32 and number of frames in a block m=63, and k_1=48 and m=42.

————: Example scheme 1
—·—·—: Example scheme 2
--------: Example scheme 3

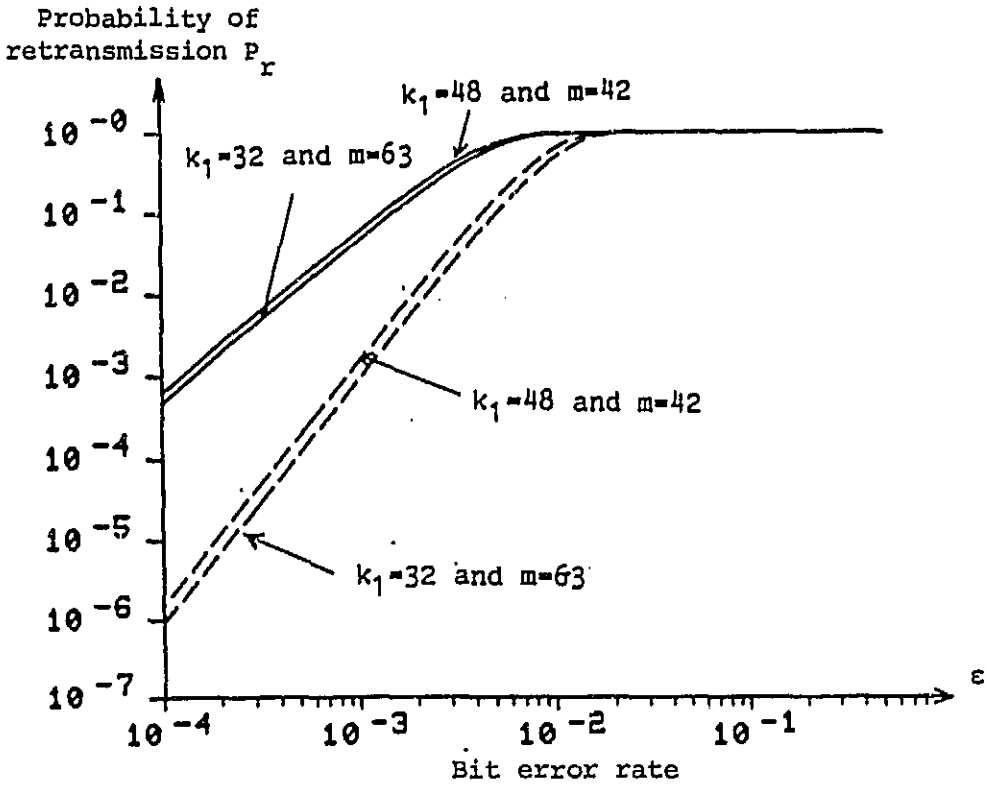


Figure 10 The probability of retransmission P_r for the number of information bits in a frame k_1=32 and the number of frames in a block m=63, and k_1=48 and m=42.

————: Example schemes 1 and 2
--------: Example scheme 3

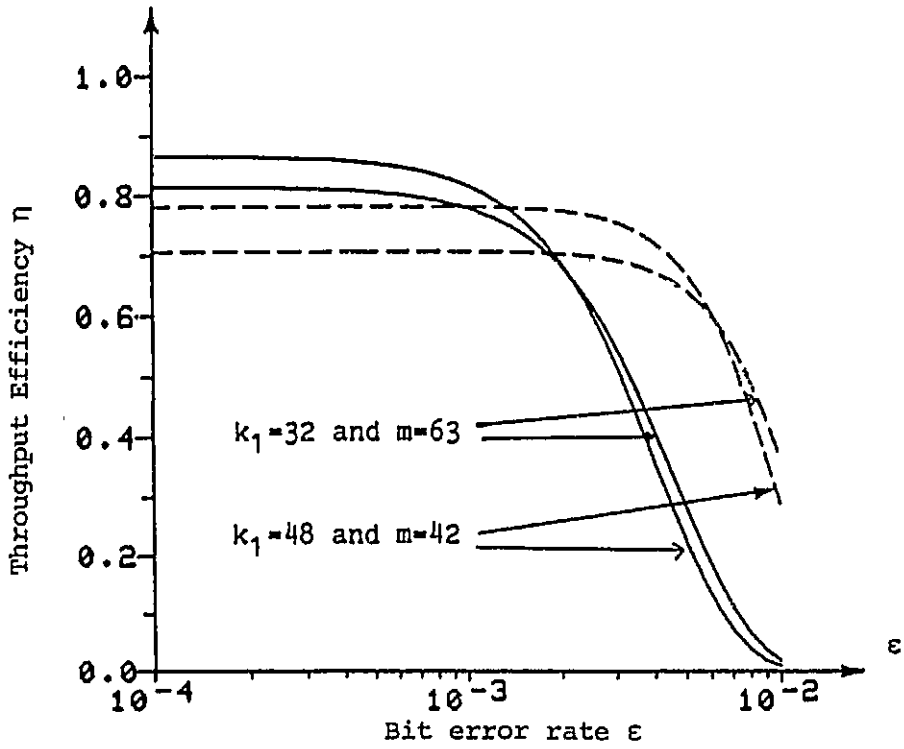


Figure 11 Throughput efficiencies of the SR+GBN ARQ with an infinite receiver buffer ($\nu=\infty$) for the number of information bits in a frame k_1=32 and the number of frames in a block m=63, and k_1=48 and m=42.

————: Example schemes 1 and 2
--------: Example scheme 3

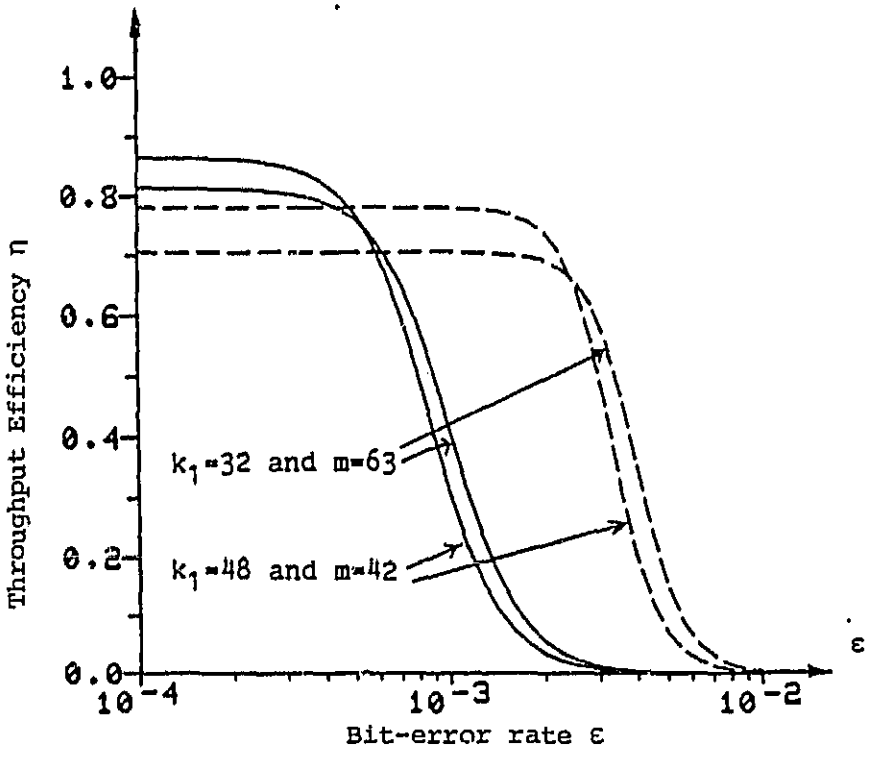


Figure 12 Throughput efficiencies of SR+GBN ARQ with $\nu = 1$ for the number of information bits in a frame k_1=32 and the number of frames in a block m=63, and k_1=48 and m=42.

————: Example schemes 1 and 2
--------: Example scheme 3

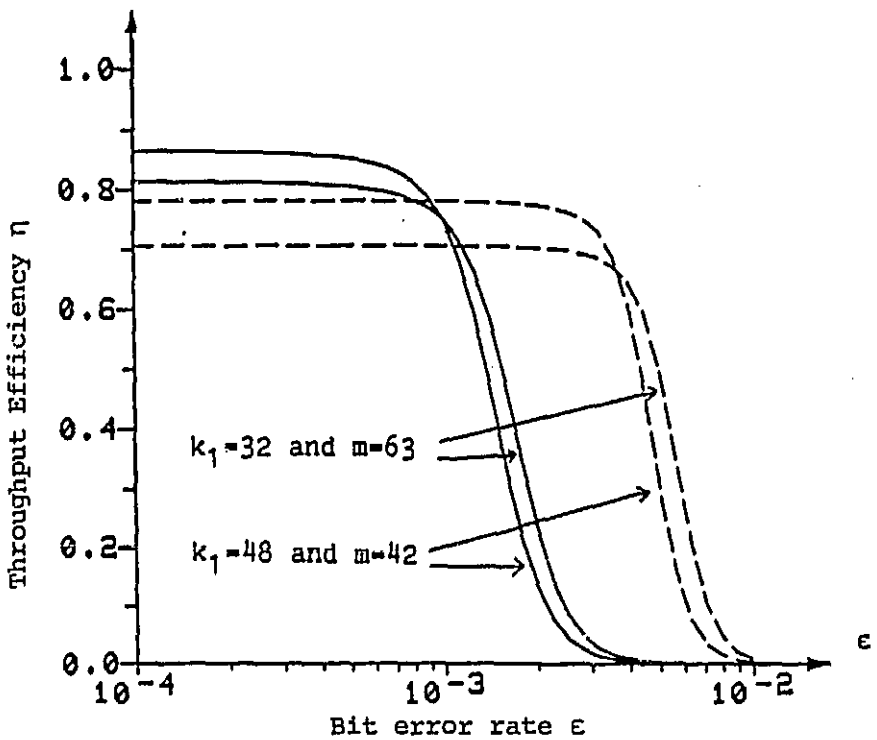


Figure 13 Throughput efficiencies of SR+GBN ARQ with $\nu = 2$ for the number of information bits in a frame k_1=32 and the number of frames in a block m=63, and k_1=48 and m=42.

————: Example schemes 1 and 2
--------: Example scheme 3