TWO HYBRID ARQ ERROR CONTROL SCHEMES FOR NEAR EARTH SATELLITE COMMUNICATIONS *

Shu Lin
Department of E.E.
Texas A & M University
College Station, Texas 77843

Tadao Kasami
Faculty of Engineering Science
Osaka University
Toyonaka, Osaka, Japan 560

ABSTRACT

In this report, two hybrid ARQ error control schemes are proposed for NASA near earth satellite communications. Both schemes are adaptive in nature, and employ cascaded codes to achieve both high reliability and throughput efficiency for high data rate file transfer.

* This work was supported by NASA Grant No. NAG 5-778.

TWO HYBRID ARQ ERROR CONTROL SCHEMES FOR NEAR EARTH SATELLITE COMMUNICATIONS

1. Introduction

In this report, we propose two hybrid ARQ error control schemes for NASA near earth satellite communications. The schemes are particularly designed to provide high system reliability and throughput for high data rate file transfer.

A hybrid ARQ scheme is a combination of a forward-error-correction (FEC) scheme and an automatic-repeat-request (ARQ) scheme [1]. It has been shown that a properly designed hybrid ARQ scheme provides both high system reliability (i.e., low error probability) and throughput [1-5]. There are two types of hybrid-ARQ schemes, type-I and type-II. In a type-I hybrid ARQ scheme, a code which is designed for simultaneous error correction and detection [1] is used. When a received word is detected in error, the receiver first attempts to correct the errors. If the number of errors (or the length of an error-burst) is within the designed error-correcting capability of the code, the errors(or error-burst) will be corrected and the decoded message will be delivered to the user or saved in a buffer at the receiver until it is ready to be passed to the user. If an uncorrectable error pattern is detected, the receiver rejects the received word and requests a retransmission. The retransmission is the same codeword. When the retransmitted codeword is received, the receiver again attempts to correct the errors (if any). If the decoding is not successful, the receiver again rejects the received word and requests another retransmission. This continues until the codeword is either successfully received (i.e., zero syndrome) or successfully decoded. The retransmission can be any of three basic types, the stop-and-wait ARQ, the go-back-N ARQ and the selective-repeat ARQ [1]. Selective-repeat ARQ is the most efficient retransmission scheme and provides the highest throughput efficiency. For high data rate file transfer over satellite links, only selective-repeat ARQ provides satisfactory throughput and reliability for high channel bit-error-rate. Of course, selective-repeat ARQ scheme is more complicated to implement than the other two ARQ schemes.

Type-I hybrid ARQ schemes are best suited for communications systems in which a

fairly constant level of noise and interference is anticipated on the channel. In this case, enough error correction can be designed into the system to correct the vast majority of received words, thereby greatly reducing the number of retransmissions and enhancing the system performance. However, for a nonstationary channel where the bit-error rate changes, a type-I hybrid ARQ scheme has some drawbacks. When the channel bit-error rate is low (for example, a satellite channel in good weather), the transmission is smooth and no (or little) error correction is needed. As a result, the extra parity-check symbols designed for error correction included in each transmission represent a waste since not much error correction is needed. When the channel is very noisy, the designed error-correcting capability may become inadequate. As a result, the frequency of retransmission increases and hence reduces the throughput.

For a channel with nonstationary bit-error rate, an adaptive hybrid ARQ system is more desirable. When the channel is quiet, the system behaves just like a pure ARQ (or a type-I hybrid ARQ) system with only parity- check symbols for error detection (or simultaneous error correction and detection) being included in each transmission. However, when channel becomes (or very) noisy, extra parity-check symbols designed for error correction are transmitted to the receiver to recover the erroneous message. That is, extra parity-check symbols for error correction are transmitted only when they are needed. This concept forms the basis of the type-II hybrid ARQ schemes. A message in its first transmission is coded with parity-check symbols for error detection (or simultaneous error correction and detection) as in a pure ARQ scheme (or a type-I hybrid ARQ scheme). When the receiver detects the presence of errors in a received word and fails to recover the message, it saves the erroneous word in a buffer, and at the same time requests a retransmission. The retransmission is not the original codeword but a block of parity-check symbols which is formed based on the original message and a second error-correcting code. When this block of parity-check symbols is received, it is used to correct the errors in the erroneous word stored in the receiver buffer. If error correction is not successful, the receiver requests a second retransmission. The second retransmission may be either a repetition of the original codeword or another block of parity-check symbols. This depends on the retransmission

strategy and the type of error correcting code to be used. The type-II hybrid ARQ scheme proposed by Lin and Yu [2] employs alternate parity-data retransmission strategy. In their scheme the parity block is of the same length as the data block, and the parity block contains the same amount of information as the data block. In an error-free situation, data can be retrieved from the parity block by taking inversion. In a noisy situation, the data block and its corresponding parity-block from a codeword in a half-rate code, and errors then can be corrected. As a result, their scheme provides high throughput even at high bit-error rates. The scheme employs two codes. If these two codes are properly chosen, high reliability can also be achieved [1,2].

In this report we propose two hybrid ARQ schemes, one is a variation of the type-I hybrid ARQ Scheme, and the other is a variation of Lin-Yu's type-II hybrid ARQ scheme. Both schemes are adaptive in nature and are designed to provide both high throughput and reliability.

2. Scheme – I

The scheme to be proposed here is a mixture of type-I and type-II hybrid ARQ schemes. A cascaded code C is used. The inner code C_1 of C is a binary (n_1, k_1) linear code with minimum distance d_1. C_1 is designed to correct t_1 or fewer errors and simultaneously detect λ_1 (with $\lambda_1 > t_1$) or fewer errors where $t_1 + \lambda_1 - 1 \leq d_1$. The outer code of C is a code obtained by interleaving a maximum-distance-separable (n_2, k_2) code C_2 over $GF(2^l)$ with minimum distance d_2. Let m_1 be the interleaving degree. Then the outer code C_o of C is an $(m_1 n_2, m_1 k_2)$ code over $GF(2^l)$. Note that $d_2 = n_2 - k_2 + 1$ [1]. The code C_2 is designed for correcting symbol errors and erasures. We assume that the following conditions hold,

$$k_1 = m_1 l, \quad n_2 = m_2(n_2 - k_2) = m_2(d_2 - 1), \tag{1}$$

with $m_1 \geq 1$ and $m_2 \geq 2$. Let C_r be the half-rate maximum-distance-separable $(2d_2 - 2, d_2 - 1)$ code obtained by shortening the code C_2. Then C_r is invertible, i.e., knowing

only the $d_2 - 1$ parity-check symbols of a codeword, the corresponding $d_2 - 1$ information symbols can be uniquely determined by an inversion operation on the $d_2 - 1$ parity-check symbols [1].

Before we describe our proposed scheme, let us examine some properties of the codes, C'_2 and C'_r. First we note that, since C'_r is a shortened code obtained from C'_2, C'_2 and C'_r can be encoded and decoded by the same circuits. Let $\bar{u}$ be a sequence of $d_2 - 1$ information symbols from $GF(2^l)$. Let $R_r(\bar{u})$ denote the sequence of $d_2 - 1$ parity-check symbols formed based on the information sequence $\bar{u}$ and the code C_r. Then $(\bar{u}, R_r(\bar{u}))$ is a codeword in C'_r. Let $\bar{v}_i$ be a codeword in C_2. Since $n_2 = m_2(d_2 - 1)$, we can divide $\bar{v}_i$ into m_2 subsequences, $\bar{v}_{i1}, \bar{v}_{i2}, \cdots, \bar{v}_{i,m_2}$; each consists of $d_2 - 1$ symbols. For $1 \leq j \leq m_2$, let $R_r(\bar{v}_{ij})$be the sequence of $d_2 - 1$ parity-check symbols formed based on $\bar{v}_{ij}$ and C_r. Clearly $(\bar{v}_{ij}, R_r(\bar{v}_{ij}))$ is a codeword in C'_r. Let

$$R(\bar{v}_i) = (R_r(\bar{v}_{i1}), R_r(\bar{v}_{i2}), \cdots, R_r(\bar{v}_{i,m_2})) \tag{2}$$

Then it can be shown that $R(\bar{v}_i)$ is also a codeword in C'_2 [see Appendix A]. In fact, $\bar{v}_i$ is a codeword in C'_2 if and only if $R(\bar{v}_i)$ is a codeword in C_2. This property will be used in our proposed error control scheme. For convenience, we call $R(\bar{v}_i)$ the parity word of $\bar{v}_i$.

Encoding

A message consists of a string of $k_1 \times k_2$ information bits. This string is divided into k_2 segments, each segment consists of k_1 information bits. Each segment is further divided into m_1 l-bit bytes. Each l-bit byte is regarded as a symbol in $GF(2^l)$. The encoding operation consists of two stages as shown in Figure 1. For each input message of $k_1 k_2$ bits, the output is an $n_1 n_2$-bit codeword in the cascaded code C. A codeword in two-dimensional format is shown in Figure 2. The transmission is done column by column and from left to right. At the first stage of encoding, each k_1-bit segment is encoded into an n_1-bit codeword in the inner code C_1, which is called a frame. At the same time, the m_1

l-bit bytes are multiplexed into m_1 C_2-code encoders to form parity-check symbols for m_1 codewords in C_2'. As soon as the k_2 segments of a message have been shifted into the overall encoder, k_2 frames have been formed and transmitted. Also all the parity-check symbols of m_1 codewords in C_2' have been formed and are in the registers of the m_1 C_2'-code encoders. Then these parity-check symbols ($m_1(n_2 - k_2)$ of them) are multiplexed and shifted into the inner code encoder to form $n_2 - k_2$ more frames (they are parity frames). These $n_2 - k_2$ parity frames and the k_2 data frames formed at the first stage together form a complete codeword array in the cascaded code C.

There is another part of the encoder. This part is for retransmission (if needed). It consists of an encoder for the half-rate code C_r, and a buffer. Let $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_{m_1}$ be codewords in C_2' which are formed by the upper part of the overall encoder. The function of C_r-encoder is to form the m_1 parity codeword in C_2',

$$R(\bar{v}_1), R(\bar{v}_2), \cdots, R(\bar{v}_{m_1})$$

corresponding to the m_1 codewords, $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_{m_1}$, where $R(\bar{v}_i)$ is given by (2). These m_1 parity words are temporarily stored in a buffer for possible retransmission.

There is another encoding arrangement. We can use a single C_2'-encoder to form m_1 codewords in C_2' (the first m_1 rows in Figure 2) and store them (in an array form) in a buffer. Then encode the n_2 columns into n_2 frames, and transmit them column by column. This encoding arrangement requires more buffer store.

Corresponding to each message of $k_1 \times k_2$ information bits, the output of the overall encoder shown in Figure 1 is a string of n_2 frames. These n_2 frames are said to form a data-block. Consider the data-block shown in Figure 2. The top k_1 rows of the array is actually regarded as m_1 l-bit byte rows. Each of these l-bit byte rows is a codeword in C_2' and is called a data-section of the code array. The m_1 data-sections form a subarray which is called a data-segment array. Each column of a data-segment array is a data-segment. There are k_2 message segments $n_2 - k_2$ parity segments. Each data section is further

divided into m_2 subsections, each subsection consists of $d_2 - 1$ symbols (or l-bit bytes) from $GF(2^l)$. The m_1 parity words, $R(\bar{v}_1), R(\bar{v}_2), \cdots, R(\bar{v}_{m_1})$, form a parity-segment array in the buffer. Each column of this array will be called a parity-segment. Each row is called a parity-section, and consists of m_2 subsections.

Decoding of a Data Block

The decoding of a data-block is basically the same as the one described in our earlier technical report on, "A Cascaded Coding Scheme for Error Control", [3]. It consists of two stages. The first stage of decoding. Depending on the number of errors in a received frame, the inner code decoder performs one of the three following operations: error-correction, erasure and leave-it-alone (LIA) operations. When a frame in a data block is received, its syndrome is computed based on the inner code C_1. If the syndrome corresponds to an error pattern $\bar{e}$ of t_1 or fewer errors, error correction is performed by adding $\bar{e}$ to the received frame. The $n_1 - k_1$ parity bits are removed from the decoded frame, and the decoded m_1-byte data segment is stored in a receiver buffer for the second stage of decoding. A successfully decoded data segment is called a decoded segment with no mark. Note that the decoded segment is error-free, if the number of transmission errors in a received frame is t_1 or less. If the number of transmission errors in a received frame is more than λ_1, the errors may result in a syndrome which corresponds to a correctable error pattern with t_1 or fewer errors. In this case, the decoding will be successful, but the decoded frame (or segment) contains undetected errors. If an uncorrectable error pattern is detected in a received frame, the inner code decoder will perform one of the following two operations based on a certain criterion:

1. Erasure Operation – The erroneous segment is regarded being erased. In fact this segment is not really removed from the buffer, it is still stored there for later use. This segment is called an erased segment. Each l-bit byte of an erased segment is regarded as an erasure for the outer code decoding.

2. Leave-it-alone (LIA) Operation – The erroneous segment is stored in the receiver buffer with a mark. We call such a segment a marked segment.

Thus, after n_2 frames of a received block have been processed, the receiver buffer may contain three types of segments: decoded segments without marks, erroneous segments with marks, and erased segments.

The above inner code decoding consists of three operations: error-correction, erasure and LIA operations. An inner code decoding which performs only the error-correction and erasure operations is called an erasure-only decoding. On the other hand, an inner code decoding which performs only the error-correction and LIA operations is called a LIA-only decoding.

When n_2 frames in a received data-block have been processed. The decoder buffer contains a decoded data-segment array with m_1 l-bit byte rows. Each of these l-bit byte rows is regarded as a received codeword from C'_2, which may contain erroneous symbols (marked or unmarked) and erasures. The code C'_2 and its decoder are designed to correct the combinations of symbol erasures and symbol errors. Maximum-distance-separable codes (or Reed-Solomon codes) with symbol from $GF(2^l)$ are most effective in correcting symbol erasures and errors.

At the second stage of decoding, the C'_2-decoder attempts to decode the rows of the data-segment array. Let i and h be the numbers of erased and marked segments respectively. The receiver stops the decoding process and requests a retransmission for the erroneous data-block if either of the following two events occurs:

(1.) the number i is greater than a certain pre-designed erasure threshold T_{es} with $T_{es} \leq d_2 - 1$.

(2.) the number h is greater than a certain pre-designed threshold $T_{el}(i)$ with $T_{el}(i) \leq (d_2 - 1 - i)/2$ for a given i.

If none of the above two events occurs, the C_2 - decoder starts the error-correction operation on the m_1 erroneous sections (or rows) of the data segment array, one at a time (they can be processed at the same time if we use m_1 C_2- decoders). The i symbol erasures and the symbol errors with or without marks in each section are corrected based on the code C_2. Let $t_2(i)$ be the error-correction threshold for a given i where

$$t_2(i) \leq (d_2 - 1 - i)/2. \tag{3}$$

If the syndrome of a section in the data-segment array corresponds to an error pattern of i erasures and $t_2(i)$ or fewer symbol errors, error-correction is performed. The values of the erased symbols, and the values and the locations of symbol errors are determined based on a certain algorithm. If more than $t_2(i)$ symbol errors are detected, then the receiver stops the decoding process and requests a retransmission for the erroneous data-block. If all the m_1 sections of a data segment array are successfully decoded, then the k_2 decoded data-segments are either delivered to the user or saved in the buffer until they are ready to be passed to the user.

Retransmission Strategy

When the receiver fails to decode a data-block $\bar{v}$, it saves the erroneous data-segment array of $\bar{v}$ in a buffer and requests a retransmission for $\bar{v}$. The retransmission is not $\bar{v}$ itself but a parity-block $P(\bar{v})$ corresponding to $\bar{v}$. The parity-block $P(\bar{v})$ is formed based on $\bar{v}$. Let $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_{m_1}$ be the m_1 sections of the data segment array of $\bar{v}$. Recall that each section $\bar{v}_i$ is a codeword in C_2. For each section $\bar{v}_i$, the encoder has already constructed a corresponding parity codeword

$$R(\bar{v}_i) = (R_r(\bar{v}_{i1}), R_r(\bar{v}_{i2}), \cdots, R_r(\bar{v}_{i,m_2}))$$

in C_2 where $R_r(\bar{v}_{ij})$ is the parity-check part formed based on the j-th subsection $\bar{v}_{ij}$ of $\bar{v}_i$ and the half-rate $(2d_2 - 2, d_2 - 1)$ code C_r(*i.e.*, $(\bar{v}_{ij}, R_r(\bar{v}_{ij}))$ is a codeword in C_r). The m_1 parity codewords, $R(\bar{v}_1), R(\bar{v}_2), \cdots, R(\bar{v}_{m_1})$ are stored as a $m_1 \times n_2$ segment array in the transmitter buffer, which is called a parity-segment array. When the transmitter receives a

request for a retransmission for data-block $\bar{v}$, the inner code encoder encodes each segment of the parity-segment array into a frame (a codeword in C_1). Hence a parity-block $P(\bar{v})$ is formed, which is also a codeword in the cascaded code C. The parity-block $P(\bar{v})$ is then transmitted to the receiver.

When a parity-block $P(\bar{v})$ is received, the receiver starts to decode it. The decoding of $P(\bar{v})$ is the same as the decoding of a data-block $\bar{v}$. If the decoding of $P(\bar{v})$ is successful, inversion is then performed on the first k_2 decoded segments of the parity-segment array. This inversion gives the k_2 data-segments of $\bar{v}$. These decoded data-segments are then delivered to the user or saved in the receiver buffer until they are ready for delivery. At this time, the erroneous data-segment array which is stored in the receiver buffer is discarded.

If the decoder fails to decode the parity-block $P(\bar{v})$, then the parity-segment array of $P(\bar{v})$ and the data-segment array of $\bar{v}$ (which is stored in the receiver buffer) together are used for error correction based on the half-rate code C_r. The receiver puts $\bar{v}_{ij}$ and $R_r(\bar{v}_{ij})$ together to form $(\bar{v}_{ij}, R_r(\bar{v}_{ij}))$. Then the C_r-decoder decodes $(\bar{v}_{ij}, R_r(\bar{v}_{ij}))$ into an estimate $\bar{v}^*_{ij}$ for $\bar{v}_{ij}$. After $m_1 \times m_2$ such decodings, the receiver contains the following estimated data-segments array:

$$\begin{array}{lcllll} \bar{v}^*_1 & & \bar{v}^*_{11}, & \bar{v}^*_{12}, & \cdots, & \bar{v}^*_{1,m_2} \\ \bar{v}^*_2 & = & \bar{v}^*_{21}, & \bar{v}^*_{22}, & \cdots, & \bar{v}^*_{2,m_2} \\ \vdots & & \vdots & \vdots & & \vdots \\ \bar{v}^*_{m_1} & & \bar{v}^*_{m_1,1}, & \bar{v}^*_{m_1,2}, & \cdots & \bar{v}^*_{m_1,m_2} \end{array}$$

Then the receiver checks whether each $\bar{v}^*_i$, for $1 \leq i \leq m_1$, is a codeword in C_2. Note that this time C_2 is used only for error detection. If all $\bar{v}^*_1, \bar{v}^*_2, \cdots, \bar{v}^*_{m_1}$ are codewords in C_2, then the decoding is successful and the k_2 estimated data-segments are accepted by the receiver. If any $\bar{v}^*_i$ is not a codeword in C_2, the receiver discards the erroneous data-segment array of $\bar{v}$ (stored in the buffer) and save the parity-segment array of $P(\bar{v})$ for

later use. At the same time the receiver requests a second retransmission for $\bar{v}$. The second retransmission is the data-block $\bar{v}$ itself. When $\bar{v}$ is received, it is decoded as before. If decoder fails to decode $\bar{v}$, then the data-segment array of $\bar{v}$ and the parity-segment array of $P(\bar{v})$ (stored in the buffer) together are used for error correction based on C'_r. If the correction process is not successful, then the receiver discards the parity-segment array of $P(\bar{v})$ and saves the data-segment array of $\bar{v}$. At the same time, the receiver requests a third retransmission for $\bar{v}$. The third retransmission for $\bar{v}$ is the parity block $P(\bar{v})$. When $P(\bar{v})$ is received, the receiver starts the decoding process again. Therefore, the retransmissions are alternate repetitions of the parity block $P(\bar{v})$ and the data-block $\bar{v}$ as shown in Figure 3. The retransmissions continue until the message in $\bar{v}$ is finally recovered by the receiver.

The major advantage of this error control scheme is that extra parity symbols for error correction are transmitted only when they are needed. These extra parity symbols are used without decreasing the rate of the cascaded code C. If the half-rate code C'_r is powerful enough, at most one retransmission is needed to recover a message. When the channel is not very noisy, the error correcting capability of the cascaded code C should be able to recover the message in its first transmission. In this situation, the system throughput is equal to the rate of C which is R_1R_2. When the channel is noisy, the first retransmission provides us the parity symbols of C_r for extra error correction capability. Since C_r is used for correcting errors only in a subsection of a codeword in C'_2, and since C'_r has the same error correcting capability as C_2, errors in the entire data-segment array should be corrected by C'_r. In this situation, the throughput of the system should be $R_1R_2/2$. If the noisy situation is rare, then the proposed error control schemes provides maximum throughput R_1R_2 most of the time.

Since C'_r is a shortened code obtained from C'_2, the decoder for C'_2 can be used for decoding C'_r. Therefore, only decoders for C'_1 and C'_2 are needed. Since the inner code C'_1 is binary and shorter, its decoder is much simpler than the decoder for C'_2.

In our earlier report [1], we showed that a cascaded coding scheme provides extremely high reliability. We expect the proposed scheme in this report will also provide extremely

high reliability. Analysis of the scheme will be given in our next report.

A special case for the above error control scheme is that $n_1 = k_1 = l$. In this case, no inner code is used, the outer code is simply C'_2 which is used for both error correction and detection. The code C'_r is used for error correction only.

Special Example Schemes

For NASA near earth satellite communications, we propose two specific schemes. For the first scheme, we choose $n_1 = k_1 = l = 8$. The outer code C_2 is the extended (256,224) Reed-Solomon (RS) code over $GF(2^8)$ (or a shortened version of this code). This code has 32 parity-check symbols and is capable of correcting any combination of t or fewer symbol errors and e or fewer symbol erasures provided that $2t + e \leq 32$. Note that the length of this code, 256, is a multiple of 32. The code C_r is the shortened (63,32) RS code obtained from shortening C_2. C_r is capable of correcting 16 symbol errors and is extremely powerful. Therefore, even in a very noisy situation, a transmitted data-block should be recovered at most with one retransmission.

Another specific scheme which we would like to propose to NASA uses an inner code C'_1. The inner code C_1 is a distance - 4 shortened (55,48) Hamming code. The code C_2 is still the (256,224) extended RS code over $GF(2^8)$. Note that l=8 and $m_1 = 6$. The code C'_r is again the (64,32) shortened RS code with symbols from $GF(2^8)$.

Note that the (256,224) RS code is actually the NASA standard code for TDRS Systems with an additional information symbol.

3. Scheme – II

The second proposed error control scheme is a type-I hybrid ARQ scheme, in which a high rate cascaded code C is used for the first transmission of a data-block and a low-rate cascaded code C' is used for retransmission of a data-block. The outer codes for C and C' are obtained by interleaving a maximum-distance-separable (n_1, k_2) code C'_2 over $GF(2^l)$.

Code C_2 has a minimum distance d_2, and is designed for correcting any combination of t_2 or fewer symbol errors and e or fewer symbol erasures where $2t_2 + e + 1 \leq d_2$. The inner code C_1 for C is an (n_1, k_1) binary code with minimum distance d_1 which is designed for correcting t_1 or fewer errors and simultaneously detecting $\lambda_1 (\lambda_1 > t_1)$ or fewer errors where $t_1 - \lambda + 1 \geq d_1$. We assume that k_1 is an even positive integer and

$$k_1 = m_1 l,$$

where m_1 is even. The inner code C_1' for C' is an $(n_1', k_1/2)$ binary code with minimum distance d_1' which is designed for correcting t_1' or fewer errors and simultaneous detecting $\lambda_1' (\lambda_1' > t_1')$ or fewer errors where $d_1' \geq \lambda_1' + t_1' + 1$. C_1' is designed with $d_1' > d_1$ and $t_1' > t_1$. The outer code for C is obtained by interleaving C_2 with a degree m_1. Hence a data-block in C is the same as shown in Figure 2. The outer code for C' is obtained by interleaving C_2 with a degree $m_1/2$. A data-block in C' also has the format shown in Figure 2 except it has only $m_1/2$ data-sections. We can see readily that C and C' are an $(n_1 n_2, k_1 k_2)$ and an $(n_1' n_2, k_1 k_2/2)$ codes respectively. If we choose $n_1' = n_1$, then both cascaded codes, C and C', have the same block length.

Let $\bar{u}$ be a message of $k_1 k_2$ information bits. Let $\bar{v}$ be its corresponding code-block in C. Divide $\bar{u}$ into two parts, $\bar{u}_1$ and $\bar{u}_2$, such that $\bar{u}_1$ consists of the first half of information bits of $\bar{u}$ and $\bar{u}_2$ consists of the second half of information bits of $\bar{u}$. Let $\bar{v}_1'$ and $\bar{v}_2'$ be the code-blocks in C' for $\bar{u}_1$ and $\bar{u}_2$ respectively. Since the data-segment arrays of $\bar{v}_1'$ and $\bar{v}_2'$ are half of the data-segment array of $\bar{v}$, we call $\bar{v}_1'$ and $\bar{v}_2'$ half blocks of $\bar{v}$.

Transmitter has two modes, mode-F and mode-H. In mode-F, a message $\bar{u}$ of $k_1 k_2$ bits is encoded into a code-block $\bar{v}$ in C and transmitted. Then the message $\bar{u}$ is stored in buffer until a positive acknowledgement(ACK) is received. When a negative acknowledgement (NAK) is received. The transmitter switches to mode-H, and the NAK'ed message $\bar{u}$ is encoded into two half code-blocks, $\bar{v}_1'$ and $\bar{v}_2'$, in C'. The first half code-block $\bar{v}_1'$ is transmitted first. The second half code-block $\bar{v}_2'$ is also transmitted in mode-H. Rule for switching from mode-H to mode-F must be chosen based on the channel's noise level. The rule must be designed to maximize the average throughput. We have not come out with a

specific rule at this time. This will be a subject of our study for the next quarter.

The decoding of a code-block in C or C'' is the same as that described in the previous section. If an uncorrectable error pattern is detected, the receiver requests a retransmission.

A Specific Example Scheme

A specific scheme which we consider for NASA's near earth satellite communications is given here. Code C_2 is the (255,223) RS code (or its shortened version) with symbols from $GF(2^8)$. The inner code C_1 for the first transmission is the (55,48) shortened Hamming code with minimum distance 4 generated by $\bar{g}(x) = (1+x)(1+x+x^6)$. Then, $m_1 = 6$. Hence the outer code of the scheme for first transmission is obtained by interleaving C_2 by degree 6. The inner code C_1' for retransmission is a (54,24) shortened cylic code with minimum distance 12 generated by

$$\bar{g}_1'(x) = (1+x)\phi_1(x)\phi_3(x)\phi_5(x)\phi_7(x)\phi_9(x)\phi_{21}(x)$$

where $\phi_i(x)$ is the minimal polynomial of α^i with α as a root of $1+x+x^6$ [1]. C_1' is a subcode of a quadruple-error-correcting one-step majority-logic decodable code [1], and is also a subcode of a shortened quintruple-error-correcting primitive BCH code. The outer code for retransmission is obtained by interleaving C_2 with degree 3. The parameters, t_1, t_1', λ_1 and λ_1' are chosen as follows: $t_1 = 1$, $\lambda_1 = 2$ $t_1' = 4$, $\lambda_1' = 7$ (*or* $t_1' = 5$ and $\lambda_1' = 6$). Majority-logic decoding of C_1' with $t' = 4$ can be easily implemented.

4. Remarks

Performance analysis of the two proposed error control schemes is under way. We expect to have the analysis and some computation results in October, 986. We expect that both schemes would provide high system reliability and throughput. In general Scheme-I is more powerful and effective than Scheme-II. However it is more complicated to implement.

REFERENCES

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

2. S. Lin and P.S. Yu, "A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels", IEEE Transactions on Communications, Vol. COM-30, No. 7, July, 1982.

3. T. Kasami and S. Lin, "A Cascaded Coding Scheme for Error Control", NASA Technical Report No. 2, Grant NAG 5-407, December 10, 1985.

END

DATE

FILMED

AUG 31 1987