

INFORMATION TO USERS

This was produced from a copy of a document sent to us for microfilming. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help you understand markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure you of complete continuity.
2. When an image on the film is obliterated with a round black mark it is an indication that the film inspector noticed either blurred copy because of movement during exposure, or duplicate copy. Unless we meant to delete copyrighted materials that should not have been filmed, you will find a good image of the page in the adjacent frame. If copyrighted materials were deleted you will find a target note listing the pages in the adjacent frame.
3. When a map, drawing or chart, etc., is part of the material being photographed the photographer has followed a definite method in "sectioning" the material. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.
4. For any illustrations that cannot be reproduced satisfactorily by xerography, photographic prints can be purchased at additional cost and tipped into your xerographic copy. Requests can be made to our Dissertations Customer Services Department.
5. Some pages in any document may have indistinct print. In all cases we have filmed the best available copy.

University
Microfilms
International

300 N. ZEEB RD., ANN ARBOR, MI 48106

8220042

Miller, Michael Joseph

**AUTOMATIC-REPEAT-REQUEST SYSTEMS FOR ERROR CONTROL IN
DIGITAL TRANSMISSION**

University of Hawaii

PH.D. 1982

**University
Microfilms
International** 300 N. Zeeb Road, Ann Arbor, MI 48106

AUTOMATIC-REPEAT-REQUEST SYSTEMS FOR
ERROR CONTROL IN DIGITAL TRANSMISSION

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN ELECTRICAL ENGINEERING
MAY 1982

By

Michael Joseph Miller

Thesis Committee:

Shu Lin, Chairman
N. Thomas Gaarder
Franklin F. Kuo
Edward J. Weldon
Edward A. Bertram

ACKNOWLEDGEMENTS

I would like to thank the members of my dissertation committee for their support and, particularly the chairman, Professor Shu Lin. I also acknowledge the encouragement and support of my colleagues at the South Australian Institute of Technology, particularly, Alan Bolton and Ross A. Frick. My thanks for excellent typing assistance to Isobel Keegan. I also want to express special gratitude to my wife and family for their willingness to make many sacrifices on my behalf.

ABSTRACT

Automatic-repeat-request (ARQ) systems have been the most popular means for error control in digital transmission systems. They provide a relatively simple and highly reliable means for eliminating transmission errors. However, the throughput of an ARQ system may deteriorate badly with increasing bit-error rates especially if there are significant transmission delays such as experienced in satellite or long terrestrial circuits.

This dissertation first proposes a class of mixed-mode protocols which incorporate a selective-repeat mode of retransmission. This is combined with a secondary mode to prevent receiver buffer overflow. The throughput analysis for these schemes is presented and shows that they can significantly outperform the conventional Go-Back-N procedure for transmission over circuits with delay. The analysis also shows how throughput is related to the size of buffer provided at the receiver. It is also demonstrated that the choice of secondary retransmission mode does not have a significant effect on the throughput but has a bearing on complexity.

Further improvement in performance may be achieved by use of a hybrid ARQ system incorporating forward-error correction as well as retransmission. The dissertation considers some parity-retransmission schemes in which blocks of parity bits are used for retransmissions rather than repetition of the original information block. This enables the system to

adaptively incorporate error correction as well as detection when channel bit-error rates increase. An analysis procedure is presented which permits comparison of throughput efficiency for a variety of ARQ retransmission protocols and forward-error correction codes. Particular attention is focussed on the use of half-rate convolutional codes for error correction. A trellis algorithm is shown to be useful for computation of the error correction capability of modestly powerful convolutional codes with sliding-block feedback decoding. An alternative approach using combinatorial procedures is also presented. New rate one-half codes are found which are related to optimum orthogonalizable rate one-third codes. These related code pairs can be used in parity-retransmission schemes to provide more powerful error correction when channel bit-error rates deteriorate badly.

Finally the throughput and reliability performance of the hybrid schemes is outlined for various combinations of retransmission protocols and error-correction systems. Results are presented for convolutional codes and block codes and indicate the possible tradeoffs between complexity and performance. It is concluded that convolutional codes with relatively simple sliding-block decoding can ensure high throughput is maintained on a hybrid ARQ system despite significant bit-error rates and transmission delays.

*'Tis a lesson you should heed:
Try, try, try again.
If at first you don't succeed,
Try, try, try again.*

HICKSON

*Nihil est quod non expugnet
pertinax opera,
et intenta ac diligens cura.*

SENECA

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
1. INTRODUCTION	
1.1 Error Control Alternatives	1
1.2 ARQ Systems-Historical	5
1.3 Dissertation Research Tasks	9
2. MIXED MODE ARQ SCHEMES	
2.1 Introduction	12
2.2 Selective-Repeat Plus Stutter (SR + ST) Schemes	13
2.3 Analysis of SR + ST Schemes	20
2.4 Selective-Repeat Plus Go-Back-N (SR + GBN) Schemes	29
2.5 Performance Calculations	33
2.6 Comparison with ARQ Schemes by Other Authors	39
2.7 Summary	43
3. PARITY RETRANSMISSION ARQ SCHEMES	
3.1 Introduction	45
3.2 Alternate Data-Parity Hybrid ARQ	46
3.3 Hybrid Schemes and Their State Diagrams	50
3.4 Throughput Analysis Procedure	62
3.5 Summary	71
4. ANALYSIS OF TYPE-II HYBRID SCHEMES	
4.1 Alternate Data-Parity with SR + ST Protocol	74
4.2 Alternate Data-Parity with SR + GBN Protocol	78
4.3 Alternate Data-Parity Scheme with SR Protocol and Infinite Buffer	80
4.4 Lin and Yu Hybrid Schemes	81
4.5 Rate 1/2, 1/3 Parity Retransmission Scheme	84
4.6 Alternate Data-Parity with ST + GBN Protocol	84
4.7 Summary	87

5.	BLOCK ERROR RATE FOR CONVOLUTIONAL CODES	
5.1	Introduction	89
5.2	Decoder Characteristics	93
5.3	Trellis Algorithm for Block Error Probability (P_B^i)	95
5.4	Combinatorial Approach	101
5.5	Computed P_B^i Values	113
5.6	Convolutional Versus Block Codes	117
6.	RELATED $R = 1/2$ and $R = 1/3$ ORTHOGONALIZABLE CODES	
6.1	Introduction	120
6.2	New Orthogonalizable Code Design	123
6.3	Encoding and Decoding	135
7.	PERFORMANCE OF HYBRID ARQ SCHEMES	
7.1	Introduction	140
7.2	Conditional Probabilities q_1 and q_2 for Block Codes	142
7.3	q_1 and q_2 for Convolutional Codes	144
7.4	Throughput for Type-II ARQ Schemes	147
7.5	Reliability	155
7.6	Error Detection Procedures	158
8.	SUMMARY AND CONCLUSIONS	
8.1	Summary of Research	163
8.2	Conclusions	168
APPENDICES		
A.	A Note on the Derivation of Equation (2.7)	171
B.	Trellis Algorithm for Calculating Block Error Rate for Convolutional Codes After Decoding	173
C.	Enumeration of Correctable Error Patterns	175
D.	Conditional Probabilities q_1, q_2 of Recovery of Data Without Error	182
BIBLIOGRAPHY		188

LIST OF TABLES

<u>Table</u>		<u>Page</u>
2.1	Mixed mode Scheme properties	34
2.2	ARQ schemes by other authors	40
3.1	Rate 1/2, 1/3 hybrid ARQ	61
4.1	Distribution of number of transmissions	77
5.1	Codes with maximal d_{free}	94
5.2	Orthogonalizable codes	95
5.3	Number of correctable error patterns	111
5.4	Block error rate versus BER	114
6.1	Related orthogonalizable codes	134
D.1	Error patterns for $\Pr(D_1 D_2 F_2)$	183

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2.1	Basic ARQ illustrated	15
2.2	SR+ST Scheme 1 ($v = 1$)	17
2.3	SR+ST Scheme 1 ($v = 2$)	19
2.4	SR+ST Scheme 2	21
2.5	ACK/NAK Sequence for evaluation of P_{NN}	26
2.6	Example of ACK/NAK indicator window	28
2.7	SR+GBN Scheme ($v = 1$)	30
2.8	SR+ST Scheme 1 Throughput	36
2.9	Throughput for various v	37
2.10	Comparison of SR+ST Scheme 2 with others	38
2.11	Comparison with Go-Back-N Schemes	41
2.12	Comparison with Yu and Lin Scheme	42
3.1	State diagram for Hybrid ARQ	53
3.2	State diagram for conventional ARQ	55
3.3	State diagram for Type-I hybrid ARQ	57
3.4	State diagram for (2,1)(3,1) ARQ	58
3.5	State diagram for inferior hybrid ARQ	67
5.1	Alternate data and parity blocks :	90
5.2	Another alternate data/parity scheme	92
5.3	Trellis for $t=2$ sliding blocks decoding	97
5.4	Trellis for $t=3$ decoding	102
5.5	Trellis for $t=4$ decoding	103

5.6	Correctable error patterns ($t=1$)	106
5.7	Block error rate for convolutional codes	116
5.8	Block codes versus convolutional codes	119
6.1	Typical convolutional encoders	124
6.2	Threshold decoder	136
7.1	Throughput of hybrid ARQ ($d_{\text{free}}=5$)	150
7.2	Performance of block and convolutional codes	152
7.3	Throughput of hybrid ARQ ($d_{\text{free}}=7$)	153
7.4	Performance of SR+GBN	154
C.1	Correctable error patterns $N(3, N_B)$	178
C.2	Correctable error patterns $N(4, N_B)$	180

1. INTRODUCTION

1.1 Error Control Alternatives

In recent years, there has been a rapid development of systems for transmission, storage and processing of signals in digital form. Digital data networks represent one such area reflecting the fast growing demand for communications between computers and associated terminal equipment. National digital networks for speech as well as data have been under development in many countries and requirements for international transmission systems can be expected to expand also.

A major concern of digital transmission system designers is to minimize the likelihood of errors in the most efficient and economical way. Within individual transmission links, errors can be caused by such factors as thermal noise, interference, frequency dependent distortion due to fading on radio links or as a result of imperfect synchronization resulting in jitter or slips. A primary task for the system designer is to provide sufficient error control procedures in the network to ensure that with high probability, all transmission errors are corrected before the data is delivered to the customer.

It is possible to classify alternative approaches to the problem of error control into three categories as follows:

- *Forward-error-correction* (FEC) procedures in which sufficient redundancy is provided by the transmission of parity bits associated with powerful error correcting codes.
- *Automatic-repeat-request* (ARQ) procedures in which certain classes of higher rate block codes are used only for high

reliability detection of forward transmission errors.

Whenever errors are detected, retransmissions of faulty sequences are initiated. Many different retransmission procedures are possible.

- *Hybrid ARQ* procedures incorporate forward error correction as well as detection and retransmission. Hybrid schemes offer the potential for better performance if appropriate ARQ and FEC schemes can be properly combined.

Currently ARQ schemes are the most popular means for error control on digital transmission systems when an appropriate feedback channel is available. For example, the current CCITT standard X.25 for packet-switched public data networks [1] envisages encoding data packets with a shortened cyclic code to produce 16 parity bits for error detection. For packets of the order of 1000 bits, this provides a relatively simple and highly reliable means for detecting transmission errors. The use of the high rate block code also minimizes the overhead incurred by the redundant parity bits. Of the several possible retransmission protocols that are possible, many public data networks use Go-Back-N procedures whereby the receiver discards all blocks subsequent to a faulty block. The transmitter goes back to the faulty block and begins retransmission of that and all subsequent blocks.

For satellite or long terrestrial circuits, the throughput efficiency of conventional ARQ systems using the Go-Back-N procedure can fall off rapidly for bit-error-rates greater than 10^{-6} [2, 3]. This occurs if there are significant round-trip delays between transmission of a block and receipt of its error status information (ACK or NAK) at the transmitter. For example, a terrestrial connection from Sydney to

Perth (Australia) involves approximately 30 ms round trip delay. If data were being transmitted in 1000-bit packets at a data rate of 1 Mbps, say, then whenever an error is discovered in a packet, approximately 30 subsequent packets must be discarded even though a high percentage of them may have been received without errors. For satellite circuits with round trip delays of the order of 700 ms the situation is considerably worse.

A selective-repeat ARQ procedure involving retransmission of only the faulty blocks offers better throughput performance. However, it is impractical because of the need for an infinite receiver buffer store. This is required to avoid loss of blocks due to buffer overflow.

In this dissertation several alternative retransmission procedures are examined to determine their throughput efficiency for various sets of channel conditions. New "mixed-mode" ARQ schemes are proposed involving combinations of Selective-Repeat retransmission (to maintain high throughput) and other protocols such as the Go-Back-N (to ensure buffer overflow is avoided for finite receiver buffers). A throughput analysis for these schemes is presented and shows that their performance approaches that of the selective-repeat scheme with infinite buffer.

This dissertation also considers hybrid ARQ schemes which incorporate forward-error-correction (FEC) as well as retransmission. It is possible to classify hybrid ARQ schemes into two general classes which have been called Type-I and Type-II hybrid schemes respectively [2].

Type-I hybrid schemes are the best known. The initial transmission (and subsequent transmission if requested) uses an (n, k) linear code which is designed for simultaneous error correction and detection. When a received word is detected in error, the receiver first attempts to locate and correct the errors. If an uncorrectable error pattern is

detected, the receiver rejects the received word and requests a retransmission. The process is repeated until the codeword is successfully received and decoded. For the code used in a Type-I hybrid scheme to be able to simultaneously correct certain error patterns and to detect other patterns, more overhead parity bits are required than for conventional ARQ where detection only is required. As a result, the Type-I scheme suffers a major drawback in that the throughput may be considerably less than for conventional ARQ when channel error rates are low. Type-I schemes may be best suited for systems in which a fairly constant level of noise or interference is anticipated on a channel.

This dissertation considers the performance of Type-II hybrid schemes where the parity bits for error correction are not transmitted unless they are needed. For the initial transmission a high rate (n, k) code C_0 is used as in conventional ARQ. This code provides for error detection only. If errors occur and retransmission is requested, a block of parity is sent instead of repeating the information bits. The parity retransmissions are based on a lower rate (perhaps $\frac{1}{2}$ rate) code C_1 designed for error correction. The FEC code may be either a block or convolutional code. Type-II hybrid schemes are adaptive to changes in channel conditions. Parity bits for error correction are only sent when channel errors occur. When error rates are low, parity overheads are minimum so throughput is higher than for Type-I hybrid schemes.

Particular emphasis is given in this dissertation to the use of convolutional codes for forward-error-correction in hybrid ARQ schemes. Techniques for analysis of throughput efficiency are discussed with particular emphasis on alternate data/parity retransmission schemes

using parity retransmission for forward-error-correction. The block error probability after bound distance decoding is computed for selected block and convolutional codes. Results are given for throughput efficiency for various parity retransmission schemes using mixed-mode protocols and convolutional and block codes. Results indicate that convolutional codes with relatively simple sliding-block decoding may ensure high throughput is maintained on an ARQ system despite significant bit-error-rates and transmission delays.

1.2 ARQ Systems--Historical

For many years, ARQ schemes have been the primary error-control procedure for data transmission links. They are a reliable and relatively simple means of ensuring that the probability of faulty data blocks being delivered to the user is very small. On the other hand, analytical developments in error-control techniques have tended to concentrate on forward-error correction [2, 38]. With the development of increasing bit-rates and significant delays in long circuits, more attention is now being given to the search for error-control procedures that will provide high throughput while maintaining high reliability. As a result, the analysis of new ARQ systems and the possibility of combining ARQ and FEC procedures has received considerable attention in recent years.

Early ARQ systems implemented, for example, by the IBM Binary Synchronous Communication (BISYNC) procedure [2] were of the stop-and-wait type. The performance of such systems was studied by Rieffen, Schmidt and Yudkin [5] for data transmission over telephone circuits. Cowell and Burton [6] examined how various channel error models affect

reliability and throughput. Chang [42], Harris and Morgan [43] and Metzner and Morgan [4] studied the basic nature of ARQ systems, classified types of ARQ schemes and examined the effects of noise in the feedback channel. Information theoretic studies of feedback systems for the binary symmetric channel included Shannon's demonstration that although feedback could not increase the capacity of a memoryless channel, it could improve the reliability. Weldon [43] also derived performance bounds for block codes on channels with feedback.

By the early 1960's, a number of different retransmission techniques and procedures had been developed including a selective-repeat type of procedure suggested by Stuart [44]. Benice and Frey [8, 9] provided the first analysis of the throughput and reliability of stop-and-wait, selective-repeat and Go-Back-N schemes taking account of errors in feedback channels and the possibility of undetected errors in either forward or reverse channels.

In the early 1970's ARQ systems were in extensive use in packet-switched and other data networks. Higher data rates and utilization of satellite circuits with long round-trip delays established the need for continuous ARQ strategies such as the Go-Back-N to replace earlier stop-and-wait procedures. International standards organizations such ISO and CCITT [1] began making efforts for protocol standardization. This resulted in the HDLC (high-level data link control) and the X.25 recommendations for node-to-node link-level procedures using Go-Back-N ARQ on full duplex links. Burton and Sullivan [13], Gatfield [14], Kersey [16] and Kaul [22] reported on throughput performance of such ARQ systems on typical terrestrial and satellite channels. Morris [23] examined the optimization of block lengths for such schemes operating

on random-error and Poisson-distributed block-error channels. Towsley and Wolf [27] developed a model to study the statistical behavior of queue lengths and waiting times at the nodes for the stop-and-wait and Go-Back-N ARQ strategies. For the random error channel, numerical results for the mean waiting time and mean transmitter queue occupancy were presented. More recently Easton [30] and Labetoulle and Pujolle [45] analyzed the HDLC Go-Back-N system throughput and mean response time for non-uniform frame sizes when the link is not saturated.

Variations on the conventional Go-Back-N procedure to improve throughput performance were suggested by Sastry [17], Morris [21], and Lin and Yu [31]. Analysis of throughput of the Go-Back-N ARQ scheme and its variations showed that it could provide adequate performance for moderate error rates. However, when data transmission rates and circuit round-trip delays increase, the throughput suffers severe deterioration when errors occur.

Selective-repeat ARQ was well known to be capable of providing considerably improved throughput under such conditions (see for example, Gatfield [14]). However, its implementation has been restricted by problems relating to receiver buffer size and buffer management. Infinite receiver buffer is required to achieve optimum throughput efficiency and avoid loss of blocks due to buffer overflow. A selective-repeat type of ARQ scheme with a finite range of block sequence numbers was suggested by Metzner [20] and studied by Easton [25,36]. However, no throughput analysis was available to show its dependence on the size of receiver buffer. Yu and Lin [32] suggested a selective-repeat scheme which employed a specified receiver buffer size and finite range of sequence numbers yet avoided overflow problems. A lower bound on

throughput performance was obtained. There still remained considerable work to be done in studying different methods of handling receiver buffer overflow in selective-repeat ARQ systems. No exact methods of analyzing throughput for finite buffer systems had been reported. As a result, it was not clear which practical retransmission schemes could achieve maximum throughput and minimum complexity.

Hybrid ARQ schemes combining forward-error correction as well as retransmission have also been the subject of considerable research in the past decade. Type-I hybrid schemes were analyzed by Rocher and Pickholtz [12] employing a high-rate BCH code capable of correcting up to three errors. Simultaneously the code provided sufficient detection capability to ensure high reliability. Results for the random-error case showed an improvement in throughput was possible for bit error rates worse than 10^{-4} . Sastry and Kanai [18, 19] also studied Type-I hybrid ARQ schemes for burst error channels. The use of convolutional codes for forward-error correction in a hybrid scheme has been described by Kahn [46]. Drukarev and Costello [35] have studied the use of sequential decoding with a time-out condition. Yamamoto and Itoh [47] have examined system performance using Viterbi decoding.

A Type-II hybrid ARQ scheme involving retransmissions of blocks of code parity bits for forward-error correction has recently received attention. This concept was first introduced by Mandelbaum [15] and extended by Metzner [24], Ancheta [29] and by Lin and Yu [34]. Mandelbaum suggested encoding the information with a Reed-Solomon code for which the parity bits can be broken up into a number of sub-blocks. The initial transmission is to be the information plus one of the parity sub-blocks. If uncorrectable errors remain, the transmitter sends an additional

parity sub-block. The code must have the property that it can be punctured, that is, some of the parity symbols can be discarded. Also, its minimum distance must increase approximately proportional to the number of parity sub-blocks. Metzner was the first to suggest schemes involving half-rate block or convolutional codes for parity retransmission. If an error occurs in a block then instead of retransmitting the original data block, an equal sized block of parity bits is sent and error correction attempted. Simple decoding schemes were compared in relation to the mean duration to the first decoding ambiguity. Lin and Yu [34] presented and analyzed the first practical Type-II hybrid scheme using half-rate block codes with parity retransmission. Throughput analysis for Type-II hybrid ARQ schemes using convolutional codes remained unavailable in the literature. It is the subject of the latter section of this dissertation.

1.3 Dissertation Research Tasks

This work has been directed towards investigating two classes of ARQ systems: (a) ARQ without FEC and (b) Type-II hybrid ARQ systems using convolutional codes for FEC. It is intended to provide useful results for high bit-rate transmission over random-error channels with significant round-trip delays.

In Chapter 2, some new ARQ procedures referred to as "mixed-mode" protocols are presented and analyzed. These schemes are of the selective-repeat type but avoid the requirement for infinite receiver buffer by use of a secondary transmission mode. Motivation for studying these schemes stemmed in part from the fact that they are analytically tractable. Throughput analysis for a range of mixed-mode schemes is presented in Chapter 2. The results show the benefits to be gained by having at least the first retransmission of a faulty block in the selective-repeat mode.

This will provide throughput performance superior to Go-Back-N systems. The variation of throughput as receiver buffer size is increased is also presented. Some alternative secondary transmission modes are suggested for use if the selective-repeat retransmissions fail. The secondary modes prevent buffer overflow. It is shown that the choice of secondary mode does not have a significant effect on the throughput efficiency.

The remainder of the work has been concerned with evaluating various Type-II hybrid ARQ systems to explore the throughput improvements achievable if forward-error correction is introduced. In Section 3, parity retransmission ARQ schemes are described. Then a unified procedure for throughput analysis of such schemes is outlined such that it can be used for a number of different combinations of parity retransmission schemes and ARQ protocols. Section 4 contains the throughput analysis for various Type-II hybrid ARQ systems.

The results are obtained in forms such that they can be applied to cases where either block or convolutional codes are to be used for forward-error correction. Results from Lin and Yu [32] can then be applied to compute throughput values for schemes using block codes.

The following sections deal with the evaluation of the performance of convolutional codes in the hybrid ARQ schemes. It is assumed that sliding block decoding schemes such as threshold decoding or syndrome lookup decoding are used with convolutional codes of modest error correction capability. It is first necessary to find ways of determining block error probabilities after decoding of the combined received information and parity blocks. An algorithm is presented to show how this can

be computed for codes with modest constraint lengths. An alternative approach using combinatorial analysis is also outlined. In Section 6 some new related rate $1/2$ and $1/3$ convolutional code pairs are found. These permit the implementation of one of the hybrid schemes suggested in Section 3. The performance of various codes and hybrid ARQ schemes is presented in Section 7 and procedures to ensure high reliability are discussed. The results indicate the throughput characteristics to be expected for various combinations of retransmission scheme, parity correction procedure and choice of code.

2. MIXED MODE ARQ SCHEMES

2.1 Introduction

Selective-Repeat (SR) types of Automatic-Repeat-Request (ARQ) schemes offer much better throughput performance than more conventional Go-Back-N schemes, especially for high bit rate channels with considerable round-trip delay between transmitting a data block and receiving an acknowledgement. The class of SR schemes is characterized by the fact that if a negative acknowledgement (NAK) is received for a given message block, the transmitter retransmits that block once and then continues transmitting other new data blocks in sequence from the point that was reached when the NAK arrived at the transmitter.

Two factors limit the application of SR type schemes, namely:

- The buffer size needed at the transmitter and receiver
- The sequence number range needed to number each block

The former is primarily a hardware complexity consideration; the latter affects the number of overhead bits in each block. This section examines two types of SR schemes which require finite receiver buffers and sequence number lengths. The schemes are described as "mixed-mode" schemes and labelled for convenience:

Selective-Repeat plus Stutter (SR + ST)

Selective-Repeat plus Go-Back-N (SR + GBN)

The SR + ST scheme provides for a Selective-Repeat (SR) type of scheme for the first v retransmissions, say, of a faulty block and a repetitive "stutter" (ST) scheme for any subsequently required retransmissions. Provided that the block error rate is not too high, this

scheme combines the high throughput of SR type ARQ and, by means of the ST mode, ensures that the buffer size required is finite. Two types of SR + ST transmission modes are examined, the first being simple to analyze but involving greater complexity in implementation. It also provides a means for improving throughput performance by increasing the number of SR type retransmissions but at the expense of increased receiver buffer storage. The second SR + ST scheme requires relatively simpler transmitter and receiver logic and buffer storage size at the receiver equal to $2S$ where:

S = round trip delay in blocks from the end of a transmission of a given block to the return of its ACK/NAK back at the transmitter.

The schemes are amenable to an exact performance analysis for the random channel.

The second class of schemes (SR + GBN) provides for one or more retransmissions of faulty blocks in a Selective-Repeat mode followed by a Go-Back-N procedure if still not successful. Throughput performance is derived and compared with the SR + ST schemes. The results show that, at least in principle, both the SR + ST and the SR + GBN schemes could be arranged to have a performance (throughput) approaching arbitrarily close to the ideal Selective-Repeat scheme with infinite buffer providing sufficient provision of buffer store is made in the receiver.

2.2 Selective-Repeat Plus Stutter (SR + ST) Schemes

Consider a computer communications system in which data is transmitted over a link in equal length blocks containing N_b bits per block. This is taken to include provision for sequence numbering of each block,

sequence numbers counting from zero to some fixed number and then recycling. In addition, the block is taken to contain sufficient parity check bits so that the receiver can detect with high reliability whether or not errors have occurred during transmission. After checking each block, the receiver sends back to the transmitter either a positive acknowledgement (ACK i) that block i , say, has been transmitted successfully or a negative acknowledgment (NAK) if the parity check procedure shows one or more errors to have occurred.

Figure 2.1 illustrates a basic ARQ scheme in which there is a finite round-trip delay of S block lengths between the end of transmission of a given block and the return to the transmitter of its ACK/NAK indication. For clarity, Figure 2.1 shows $S = 4$ blocks although on high data rate satellite circuits S may be up to several hundred blocks delay. In the following analysis it is assumed that the physical communications channel, once established, is assumed to have a constant delay S . In general, the actual round-trip delay interval will not equal exactly an integer number of block intervals, but S is taken to be the smallest integer number of block intervals just greater than or equal to the actual delay. The value of S is assumed known at the transmitter and receiver perhaps as the result of the use of an initial "training" sequence or synchronization procedures using the successfully transmitted numbered blocks and their ACK indications.

At the transmitter, all blocks which have been transmitted are held in the buffer until positively acknowledged. Whenever the transmitter receives a NAK i for a given block i a retransmission procedure is initiated. For ease of explanation in what follows, we refer to numbered

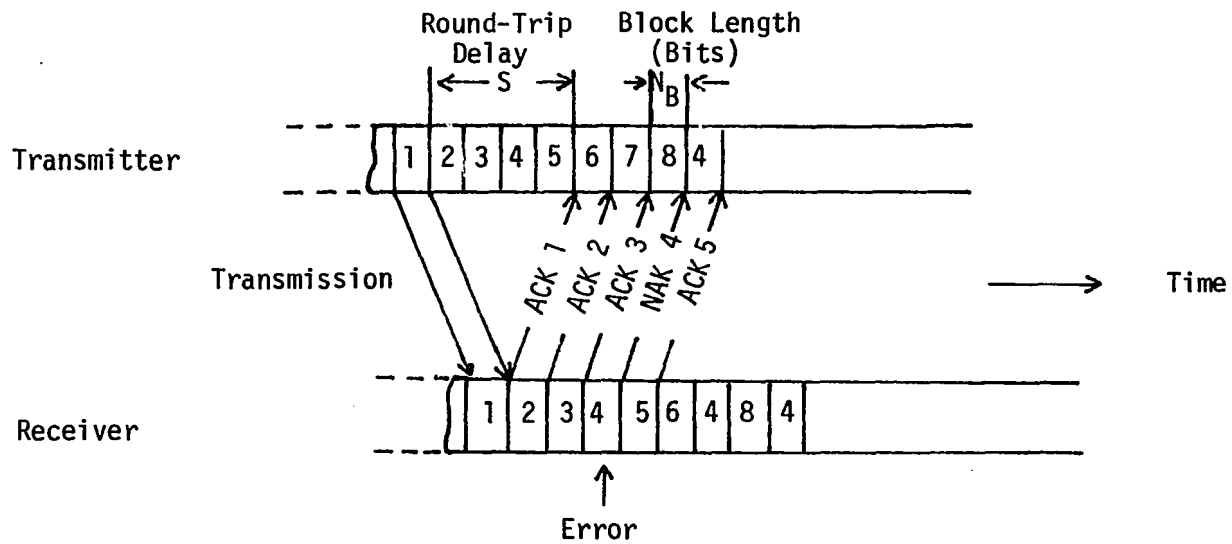


Figure 2.1

Basic ARQ Illustrated

NAK indications, NAK i being associated with block i ; however, since S is assumed fixed and known, the receiver is only required to send a NAK after receipt of a faulty block. Assume initially that all blocks previous to i having been successfully transmitted. Then the following types of so-called SR + ST retransmission schemes will be considered here:

SR + ST Scheme 1. Consider the case where the number (v) of SR type retransmissions is one. On the first receipt of a NAK i , the transmitter retransmits block i as the next block and then continues transmitting other new data blocks from the sequence number reached when the NAK arrived at the transmitter. This is referred to as the Selective-Repeat (SR) mode. If another NAK i is received indicating the second transmission of block i was unsuccessful, the transmitter then retransmits block i repetitively until an ACK i is received for that block. This mode, called a Stutter (ST) mode after Towsley [28] ensures that ultimately each block is transmitted correctly with a finite receiver buffer size and block sequence number length. We assume the parity check and ACK/NAK procedures function correctly. Figure 2.2 illustrates the procedure. Note that the maximum required receiver buffer size is $2S$ blocks.

If a NAK j is received at the transmitter while it is in the ST mode repetitively transmitting block i ($j > i$), then the transmitter stores block j until an ACK i is received to transfer operation back to the SR mode. Then block j and any other NAK'ed blocks are transmitted, followed by new data blocks from the sequence point reached when the transmitter entered the ST mode.

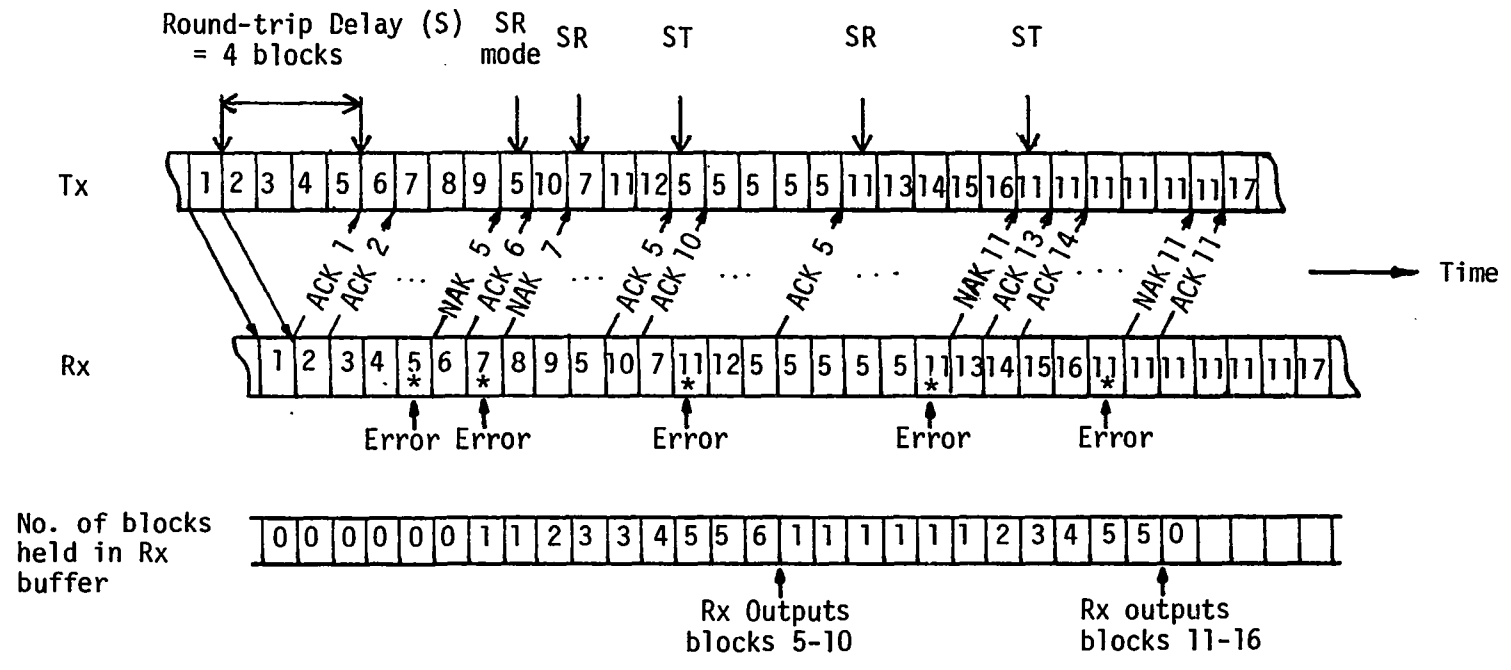


Figure 2.2
SR + ST Scheme 1 (v = 1)

In this scheme, it is necessary for the transmitter to keep a count of the number of transmission attempts for each block in its buffer store. When a NAK is received for any block, the transmitter will retransmit it in the SR mode if there had been only one previous transmission of that block; otherwise, repetitive ST mode transmission is required.

This scheme can be extended to allow for $v > 1$ SR mode retransmission attempts for any block i , the failure of such attempts (and the first transmission) causing the transmitter to revert to the ST mode and repeating block i until successfully acknowledged. A receiver buffer size of $S(v+1)$ blocks would avoid overflow. Figure 2.3 illustrates such a scheme for $v = 2$. Note that if a block sequence number scheme were used, modulo- $S(v+2)$ sequence numbers would be required to avoid ambiguities.

SR + ST Scheme 2. This scheme is similar to the above but only requires the transmitter to keep check on the number of transmission attempts of the first NAK'ed block in its buffer. The transmitter and receiver logic are somewhat simpler for this scheme in that all NAK'ed blocks concurrently in the transmitter buffer are treated in the same retransmission mode (either SR or ST) whereas in the SR + ST scheme a mixed retransmission strategy was provided depending on the NAK count for each block. The SR + ST Scheme 2 is as follows.

In error-free operation the transmitter sends blocks continuously. When the transmitter receives a NAK(i) indicating that block i is received in error, the retransmission mode is chosen according to whether the transmitter is in the Flag Set (FS) or Flat Not Set (FNS) state. This is determined as follows:

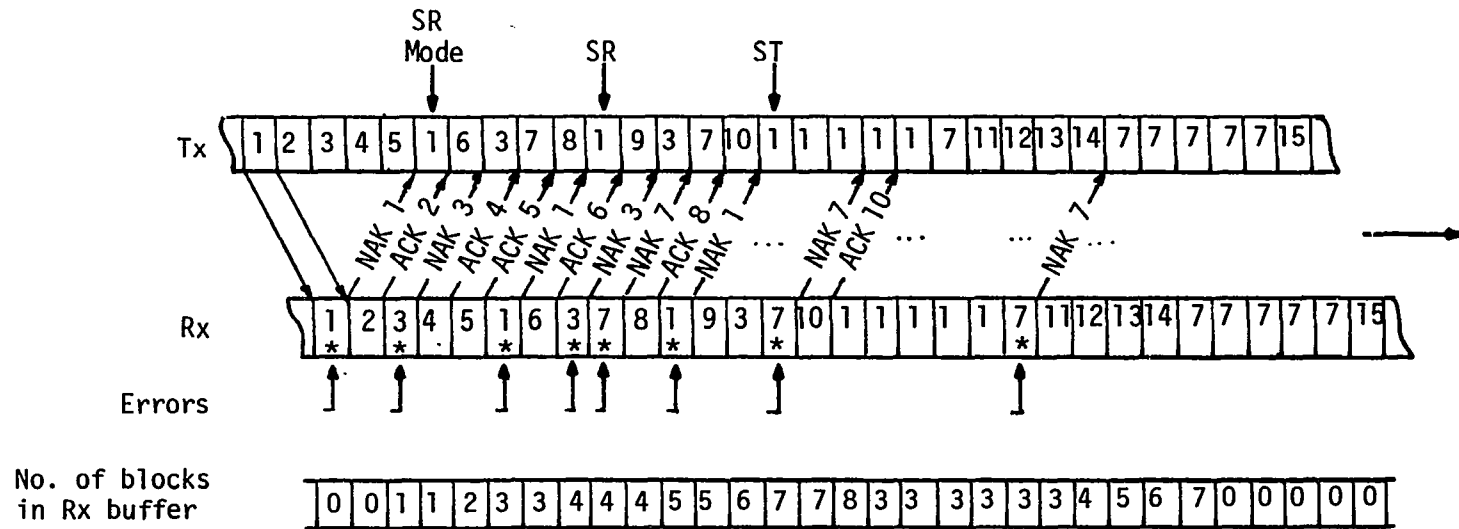


Figure 2.3

SR + ST Scheme 1 Illustrated for $v = 2$

- FS--Assuming the transmitter initially in the FNS state, then it enters the FS mode immediately on the reception of two successive NAK(i)'s for any given block i. That is, the failure of two transmission attempts for any block causes the transmitter to enter the FS mode.
- FNS--Once the transmitter enters the FS mode it remains there until all NAK'ed blocks in the retransmission buffer have been transmitted successfully and appropriate ACK's received back at the transmitter. Once that occurs, the transmitter enters the FNS condition.

When in the FS mode, the first NAK'ed block waiting for retransmission is transmitted continuously (Stutter mode) until an ACK is received for that block. Then the process is repeated for the second block (if there is one) waiting for retransmission and so on until all NAK'ed blocks in the transmitter buffer have been sent successfully. Upon entering the FNS mode, the transmitter sends new data blocks unless a single NAK(i) is received for a block i in which case that block will be immediately retransmitted ahead of subsequent new data blocks. The procedure is illustrated in Figure 2.4. The maximum receiver buffer size required is $2S$ blocks to avoid overflow, the same requirement as for the previous scheme for $v=1$.

2.3 Analysis of SR + ST Schemes

The channel model. The analysis is based on a random error channel model. Let

$$p = \text{Pr} \{ \text{a transmitted bit is received in error} \}$$

$$P_B = \text{Pr} \{ \text{a transmitted } N_B \text{ bit block is received in error} \}$$

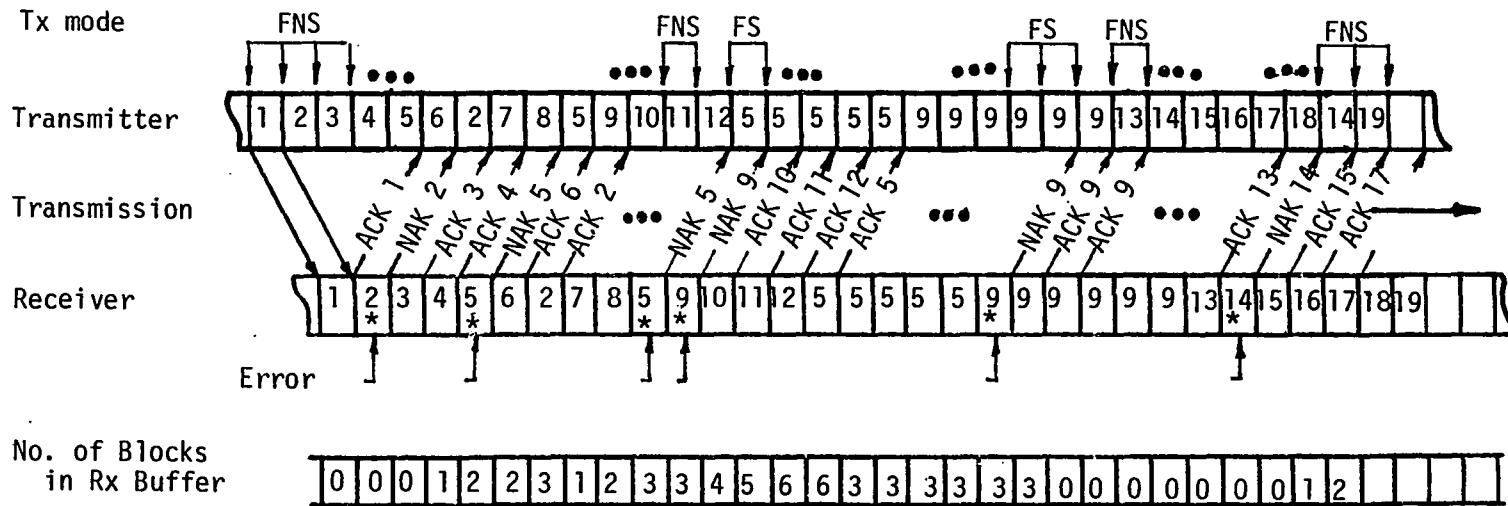


Figure 2.4

SR + ST Scheme 2 Procedure

Then

$$P_B = 1 - (1 - p)N_B \quad (2.1)$$

Let N_i = the total number of transmissions required for block i to be successfully received without error (including the first transmission).

Since the channel errors are assumed randomly distributed, then the random variables N_i can be considered independent and identically distributed with

$$P_N(n) \triangleq \Pr \{N=n\} = (1-P_B)P_B^{n-1} \quad n = 1,2,\dots \quad (2.2)$$

where the generic random variable N is used to replace the N_i 's. The analysis of ARQ throughput in the following sections assumes the model of equation (2.2) which describes a system where there are no errors in the parity detection of block errors and where there are no errors in the ACK/NAK message fed back to the transmitter.

Should it be required, however, to take account of random errors on the return channel, this could be done in a straight forward manner as follows.

Let $P_R = \Pr \{a \text{ transmitted ACK/NAK is received in error}\}$

Then assuming errors on the return channel are random and independent of these on the forward channel, the random variable N_i can be modelled [27] with probability generating function

$$G_N(z) = \frac{(1-P_B)(1-P_R)z}{(1-P_B z)(1-P_R z)} \quad (2.3)$$

Throughput for SR + ST Scheme 1. The analysis of each of the ARQ schemes is facilitated by the use of random variables N_i (the number of transmissions of block i) and M_i which is defined as

M_i = the total number of block transmission intervals required
for the successful transmission of message block i

For SR + ST scheme 1 if $v = 1$, possible M_i values are:

- 1 if block i is transmitted successfully on its first transmission attempt
- 2 if block i is transmitted successfully only on its second attempt (the latter being in the SR mode)
- 3+S if block i is transmitted successfully only on its third attempt (the latter being in the ST mode and therefore followed by S unused repetitions of block i)
- 4+S if block i is transmitted successfully only on its fourth attempt (and followed by the unavoidable S unused repetitions of block i), etc.

For the class of ARQ schemes analyzed in this section, the random variables M_i can be assumed independent and identically distributed. Furthermore, for the retransmission scheme, the total number of transmission intervals (in blocks) required to successfully transmit a given number of message blocks is equal to the sum of the M_i 's.

For the SR + ST scheme 1 ($v = 1$)

$$M_i = \begin{cases} N_i & \text{for } N_i = 1 \text{ or } 2 \\ N_i + S & \text{for } N_i = 3, 4, \dots \end{cases}$$

The throughput efficiency (η) is defined as follows:

$$\eta = \frac{\text{average no. of data blocks received successfully per time interval } t}{\text{no. of data blocks that could have been transmitted in the interval } t \text{ if there were no errors}}$$

Since the number of transmission slots (M_i) counted for transmission of any one message i are disjoint from those from any other M_j for message j ($j \neq i$), then η is given by

$$\eta = \lim_{L \rightarrow \infty} [L/E\{\sum_{i=1}^L M_i\}] = \frac{1}{E\{M\}} \quad (2.4)$$

Using the above $N_i \rightarrow M_i$ mapping and the channel model (Equation 2.2) this expected value can be readily evaluated as follows

$$\begin{aligned} E\{M\} &= (1-P_B) 1 + 2P_B + 3P_B^2 + \dots + (v+1)P_B^v + \sum_{i=v+2}^{\infty} (i+S)P_B^{i-1} \\ &= 1 + \frac{P_B}{1-P_B} + SP_B^{v+1} \end{aligned} \quad (2.5)$$

the first term being associated with the first transmission, the second with the SR-type retransmission and the last with the ST-type retransmissions.

Hence, the throughput for the SR + ST scheme is obtained as

$$\eta = \frac{1-P_B}{1+SP_B^{v+1}(1-P_B)} \quad (2.6)$$

Throughput for SR + ST Scheme 2. Let the random variables N_i and M_i be defined as before. Note that if $N_i=2$ transmissions for error-free reception of block i , then the total number (M_i) of transmission slots needed depends on whether or not the transmitter is in the FNS or FS modes when block i is transmitted for the second time. If there were no previous un-ACK'ed blocks with two NAK's, then the transmitter would be in the FNS condition and the second transmission of the block i would be in the SR mode. This would be the normal situation unless the channel is extremely noisy. If, however, one or more blocks previous to block i remain in the transmitter buffer and have been unsuccessfully transmitted (NAK'ed) twice, then the transmitter will be in the FS condition. In

this case, the second transmission of block i will be in the ST mode and, if successfully, $M_i=2+S$ since it includes S unused repetitions of block i . Clearly this will somewhat reduce the throughput over the SR + ST scheme 1.

Due to the round-trip delay, the first NAK $_i$ will be received at the transmitter S blocks after block i was first transmitted. Furthermore, the transmitter must be in the FNS condition when block i was first transmitted; no new blocks could have been sent unless the transmitter was in the FNS state, in this case $S+1$ blocks prior to NAK $_i$ being received. Therefore, the problem is to find the probability (P_{NN}) that the transmitter will be in the FS state $S+1$ time slots after transmission of block i is commenced, given that the transmitter is in the FNS condition at the time when block i is first transmitted.

The set of ACK/NAK signals that must be considered in the combinatorial analysis required to find P_{NN} can be illustrated by Figure 2.5. It is necessary to consider all combinations of ACK/NAK indications received back at the transmitter for the $2S+1$ occasions prior to the time of receipt of the first NAK $_i$ for block i . P_{NN} is the probability of two NAK's being received exactly $S+1$ time intervals apart, the first NAK being one of the S ACK/NAK's prior to any block i being first transmitted and the second being one of S ACK/NAK's immediately following. This can be seen from the following.

As shown in Figure 2.5, let the ACK/NAK indication received at the transmitter $2S+1$ block intervals before NAK $_i$ be designated ACK/NAK $_j$. Subsequent ACK/NAK signals are ACK/NAK $_{j+1}$, ACK/NAK $_{j+2}$ and so on up to ACK/NAK $_{j+2S}$. The next ACK/NAK is NAK $_i$. Now if say ACK/NAK $_j$ and ACK/NAK $_{j+S+1}$ were both NAK's then they must have related to the same block sequence number since the first NAK would have resulted in the

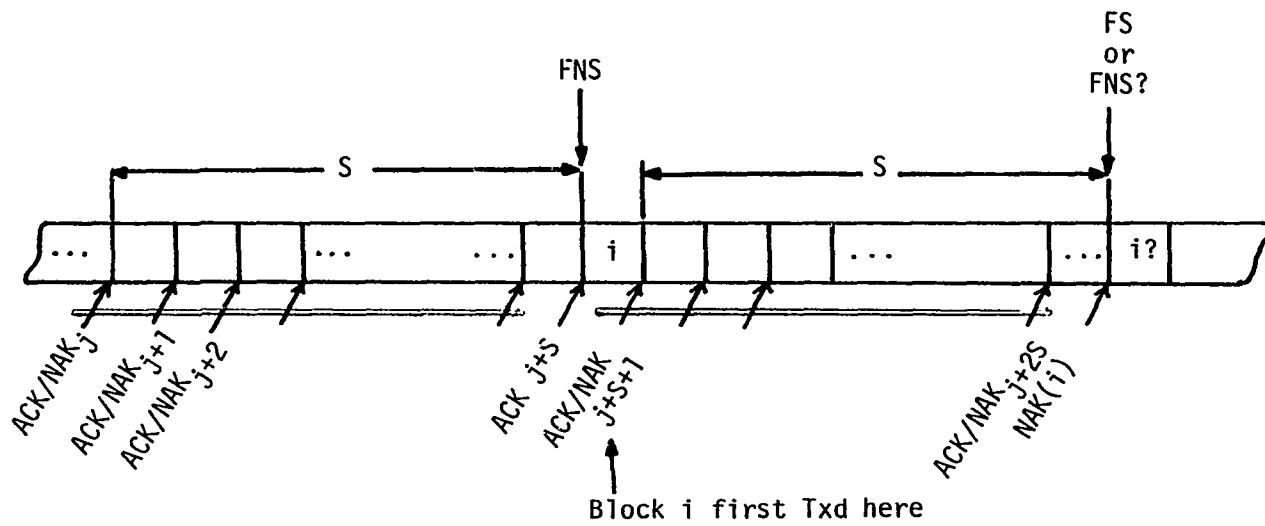


Figure 2.5
 ACK/NAK Sequence for Evaluation of P_{NN}

immediate retransmission of the associated faulty block. Hence, two NAK's received at times j and $j+S+1$ would cause the FS condition to be set at the transmitter and this condition would remain for at least $S+1$ block intervals. Such a condition would ensure that the FS condition had already been set when NAK $_i$ is received. Likewise, when NAK $_i$ is received a prior FS condition would be in operation if there had occurred any pair of NAK's (due to faulty block transmissions) separated $S+1$ units in time over the interval from ACK/NAK $_j$ to ACK/NAK $_{j+2S}$.

To illustrate, Figure 2.6 shows the ACK/NAK "window" to be considered for the case where the round-trip delay is $S=4$ blocks. In this example $i=9$ and the transmitter is in the FS condition when NAK i is received at the transmitter, the FS condition resulting from two previous NAK(2)'s in the "window" interval from ACK/NAK $_j$ (= ACK 1) to ACK/NAK $_{j+2S}$ (= ACK 8). Hence, in this case, block 9 would not receive an SR type second transmission but would be held in the transmitter store until an ACK 2 is received and then block 9 would be transmitted repetitively in the ST mode.

Using the combinatorial principle of Inclusion and Exclusion one can count the events and their probability values in the sample space of P_{NN} resulting

$$P_{NN} = \sum_{j=0}^{S-1} \sum_{k=2j}^{2S-2} \frac{(-1)^j S!}{(S-j-1)!(j+1)!} \frac{(2S-2-2j)!}{(k-2j)!(2S-2-k)!} p_B^{2+k} (1-p_B)^{2S-2-k} \quad (2.7)$$

Further details on the derivation of this expression are found in Appendix A.

It is now possible to calculate the throughput (η) for the SR + ST scheme 2. The expected value of the M_i 's can now be evaluated using (2.7) and the channel model (2.2).

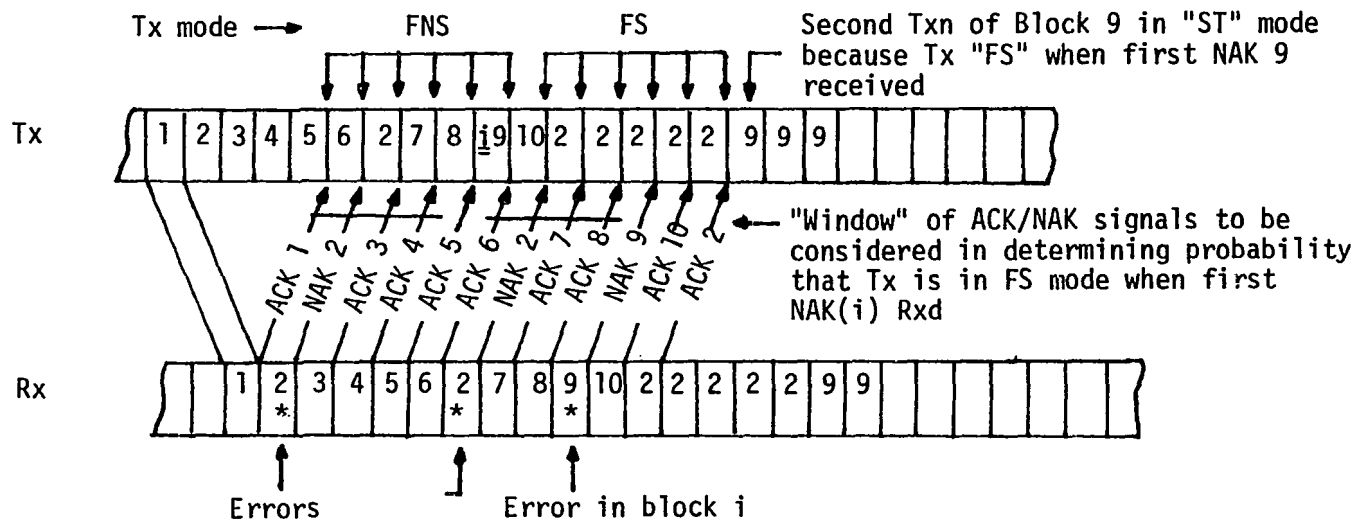


Figure 2.6

Example ($S=4$) of ACK/NAK Indicator Window to be Considered for Evaluation of P_{NN}

Hence, the throughput (η) for this SR + ST scheme 2 becomes

$$\eta = \frac{1 - P_B}{1 + SP_B^2(1 - P_B) + SP_B P_{NN}(1 - P_B)^2} \quad (2.8)$$

Note that if $P_{NN} = 0$, equation (2.8) reduces to (2.6) for the SR + ST scheme 1 with $\nu = 1$. The term containing P_{NN} therefore represents the reduction in performance of scheme 2 over scheme 1, a trade off against logic circuit complexity.

2.4 "Selective-Repeat Plus Go-Back-N" Schemes

Essentially, this scheme provides for two modes of retransmission of a block following an error in its previous transmission. The modes are

- Selective-repeat type retransmission of the faulty block ν times after its first ν consecutive faulty transmission attempts,
- Go-Back-N retransmission of the block for subsequent attempts following $N_i \geq \nu + 1$ consecutive failures in transmission attempts for that block. That is, the transmitter stops sending new blocks, backs up to the NAK'ed data block and resends that block and the S succeeding blocks that were transmitted during the round-trip delay.

Figure 2.7 illustrates the transmission/retransmission procedure for the case $\nu = 1$, namely one attempt at S-R retransmission and all subsequently attempts being Go-Back-N. Following the second consecutive NAK for a given block (e.g. block 7 in Figure 2.7) the receiver simply discards all S subsequent blocks received until the successful reception of the faulty block. In the transmitter and the receiver, provision

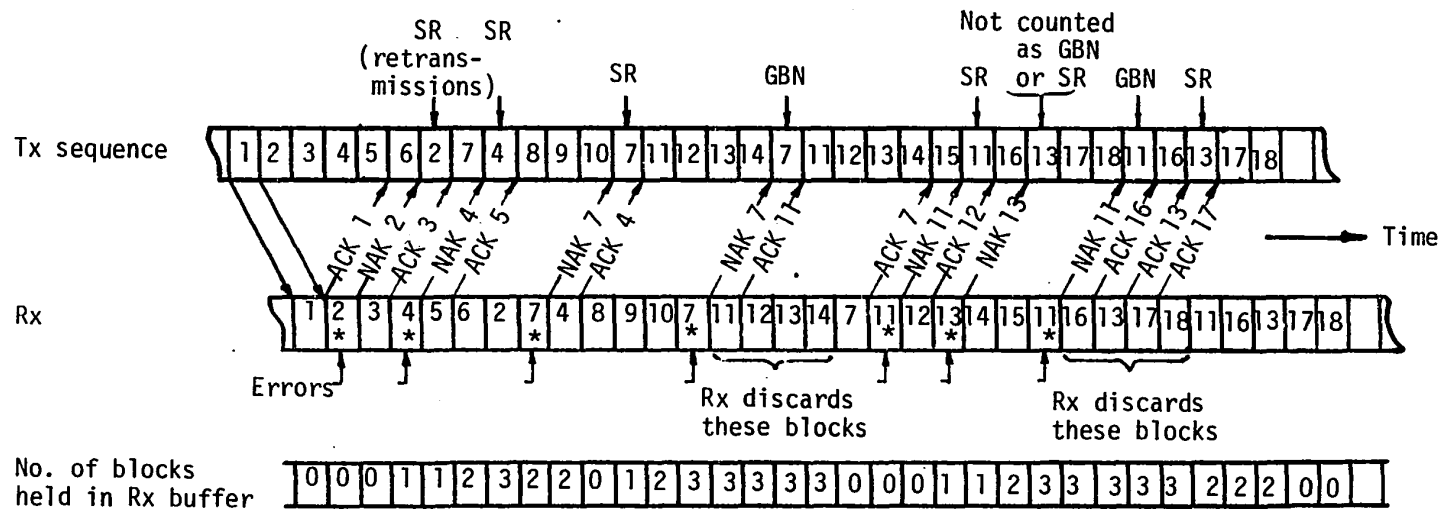


Figure 2.7

SR + GBN Scheme for $v = 1$

would be required for counters to keep check on the number of transmission attempts made for each block in order to determine whether a SR or GBN type of retransmission mode was required.

If there are more than one block stored in the transmitter retransmission store such blocks having been transmitted twice unsuccessfully then they must be queued for successive independent Go-Back-N type retransmissions. The earliest double-NAK'ed block i say will be retransmitted in Go-Back-N mode that is followed by the S blocks which were transmitted previously after block i and subsequently discarded by the receiver. This will be repeated a sufficient number of times until an ACK is received for block i . Then the same procedure is repeated for any subsequent double-NAK'ed blocks in the retransmission queue. On the second successive NAK for a given block the receiver discards all S subsequent blocks received; and also the transmitter counters for those blocks must be reduced by one. For example, in Figure 2.7, the first transmission attempt of block 11 resulted in a discarding of the block by the receiver because of a prior double error for block 7. The next transmission attempt of block 11 resulted in an error so the next (third) transmission attempt is in the SR mode. This also fails, so subsequent blocks received (16, 13, 17, 18) are discarded and must be subsequently retransmitted along with block 11. One of them (block 13) had previously been NAK'ed, and was discarded following an attempt at SR retransmission. Accordingly, the transmitter counter for retransmissions of that block would be reset and the block subsequently given another SR retransmission attempt. It should be mentioned that these procedures are not only sensible from a practical point of view but also makes it possible to carry out exact throughput analysis.

In principle, the scheme could be extended to allow for more than one retransmission attempt in SR mode just as was described for the previous SR + ST scheme. That is, values of $v = 2, 3, \dots$ could be used, at the expense of complexity of transmitter and receiver logic and of increased required receiver buffer store. The amount of receiver buffer store required to avoid overflow in this scheme is equal to $v(S+1)$ blocks.

For small v , this represents a notable saving over the SR + ST schemes which required $S(v+1)$ blocks of buffer store. As will be seen, the throughput performance of the SR + GBN is only slightly inferior to the equivalent SR + ST scheme with the same value of v .

Analysis of SR + GBN Scheme. Consider first the case where $v=1$. As in the analysis of the SR + ST scheme we use N_i and M_i to denote the number of transmission attempts and slots respectively associated with message block i . In this scheme, the value of M_i is taken to include all S blocks following each GBN type retransmission of block i since these blocks are discarded by the receiver while successful transmission of block i is sought. The retransmission strategy ensures that the M_i 's relate to disjoint sets of transmission intervals. To ensure this holds for any error pattern, the transmitter count of the number of transmission attempts for any block i must be reduced by one if any block in the S blocks transmitted prior to block i is NAK'ed for the second successive time. This is the case illustrated in Figure 2.7 for the third transmission attempt of block 13.

There is therefore a one-to-one correspondence between the N_i random variable values and the associated M_i , the mapping being represented by

$$M_i = N_i + S(N_i - v - 1) U(N_i - v - 2) \quad (2.9)$$

where $U(\cdot)$ is the unit step function.

Following the technique of the previous section, it is straightforward to see that $E\{M\}$ as a function of v is given by:

$$\begin{aligned} E\{M\} &= \sum_{n=1}^{v+1} n P_N(n) + \sum_{n=v+2}^{\infty} \{(n-v-1) S + n\} P_N(n) \\ &= 1 + \frac{P_B}{1-P_B} + \frac{S P_B^{v+1}}{1-P_B} \end{aligned} \quad (2.10)$$

and hence, the throughput (η) follows as

$$\eta = \frac{1-P_B}{1+S P_B^{v+1}} \quad (2.11)$$

2.5 Performance Calculations

In this section some typical values for throughput (η) are calculated and compared for each of the schemes outlined above. Table 2.1 summarizes the key parameters for each of the schemes. To provide comparative performance bounds, the parameter values for conventional Go-Back-N and Selective-Repeat schemes are also included.

The table shows clearly how the throughput efficiency values range between values for the conventional Go-Back-N (worst case) and the ideal Selective Repeat Scheme. In each case, the trade off between throughput performance and provision for adequate receiver buffer store is evident. As indicated, the complexity of the logic systems required in transmitter and receiver would be greater for those schemes with better throughput performance.

Table 2.1

Summary of Mixed-Mode Scheme Properties

ARQ Scheme	Throughput (η)	Required Receiver Buffer Size (# of Blocks)	Relative Complexity of * = simplest **** = most complex
Ideal Selective Repeat	$1 - P_B$	Infinite	Not practical
SR + ST Scheme 1	$\frac{1 - P_B}{1 + SP_B^{v+1}(1 - P_B)}$	$S(v+1)$	***
SR + ST Scheme 2	$\frac{1 - P_B}{1 + SP_B^2(1 - P_B) + SP_B P_{NN}(1 - P_B)^2}$	$2S$	**
SR + GBN Scheme	$\frac{1 - P_B}{1 + SP_B^{v+1}}$	$v(S+1)$	****
Conventional Go-Back-N	$\frac{1 - P_B}{1 + SP_B}$	1	*

Figures 2.8-2.12 show for the various ARQ schemes how throughput (η) varies as a function of

- Bit-error rate P_{BER}
- Block size N_B (bits)
- Round-trip delay S (blocks)

Figure 2.8 shows throughput values for SR + ST scheme 1 ($v=1$) for various values of round-trip delay (S). Note that for negligible delay ($S=1$) the performance approximates closely that of the ideal Selective-Repeat scheme. For such channels, there is negligible advantage to be gained by using the SR + ST scheme over the conventional Go-Back-N scheme. However, as S increases (particularly, say for the satellite channel) the SR + ST scheme 1 is markedly superior to the Go-Back-N ARQ system.

Figure 2.9 shows how the throughput can be improved by allowing $v > 1$ retransmissions in the SR mode. As v increases, the throughput (η) approaches the ideal Selective-Repeat performance. The differences are small for values of $\eta \geq 0.6$. However, as the bit-error rate deteriorates further, the residual effect of the ST mode retransmissions becomes significant.

SR + ST Scheme 2. Figure 2.10 shows values of throughput for $N=524$ and $S=128$ for the SR + ST scheme 2 in comparison with other ARQ schemes. The SR + ST scheme 2 is significantly better than the Go-Back-N convention especially for bit-error rates in the region 10^{-6} to 10^{-4} .

SR + GBN Scheme. Figure 2.9 also demonstrates the throughput performance of the SR + GBN scheme for various values of v in comparison with the SR + ST scheme 1. Note that the latter scheme gives slightly

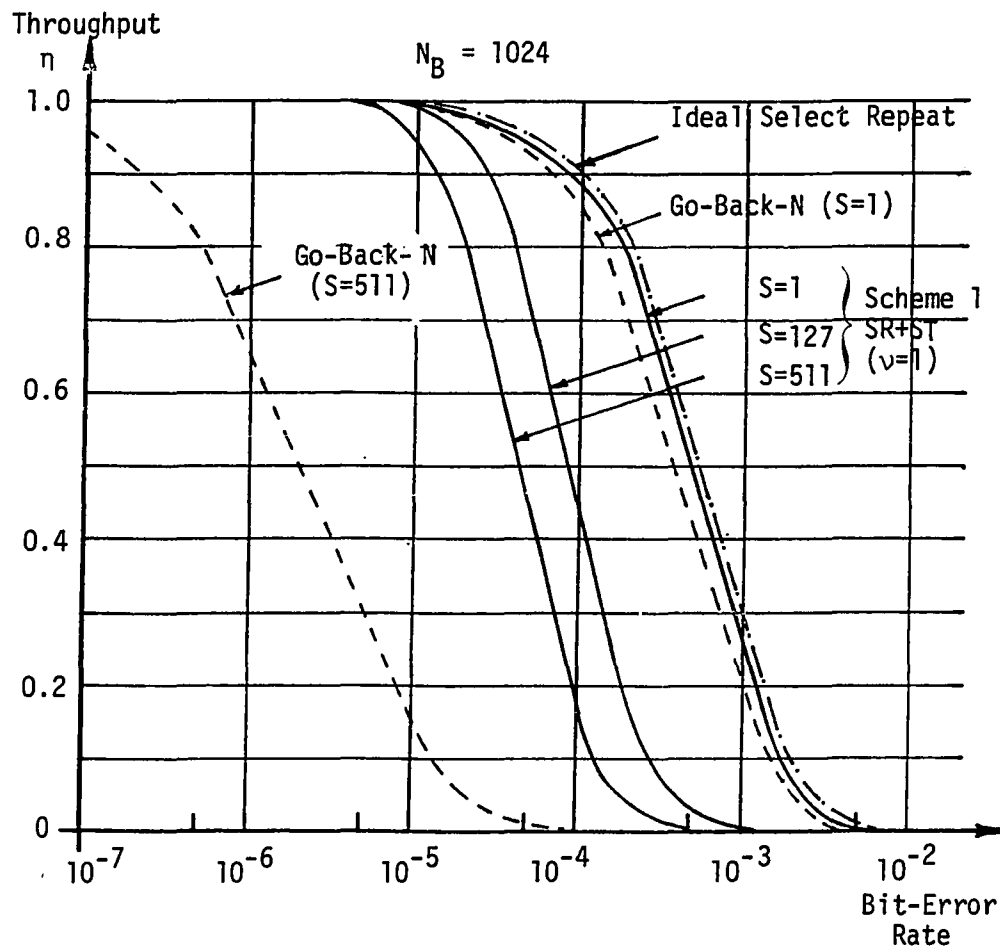


Figure 2.8

SR + ST Scheme 1: Throughput for Various Round Trip Delays (S) $N_B=1024$ Bits

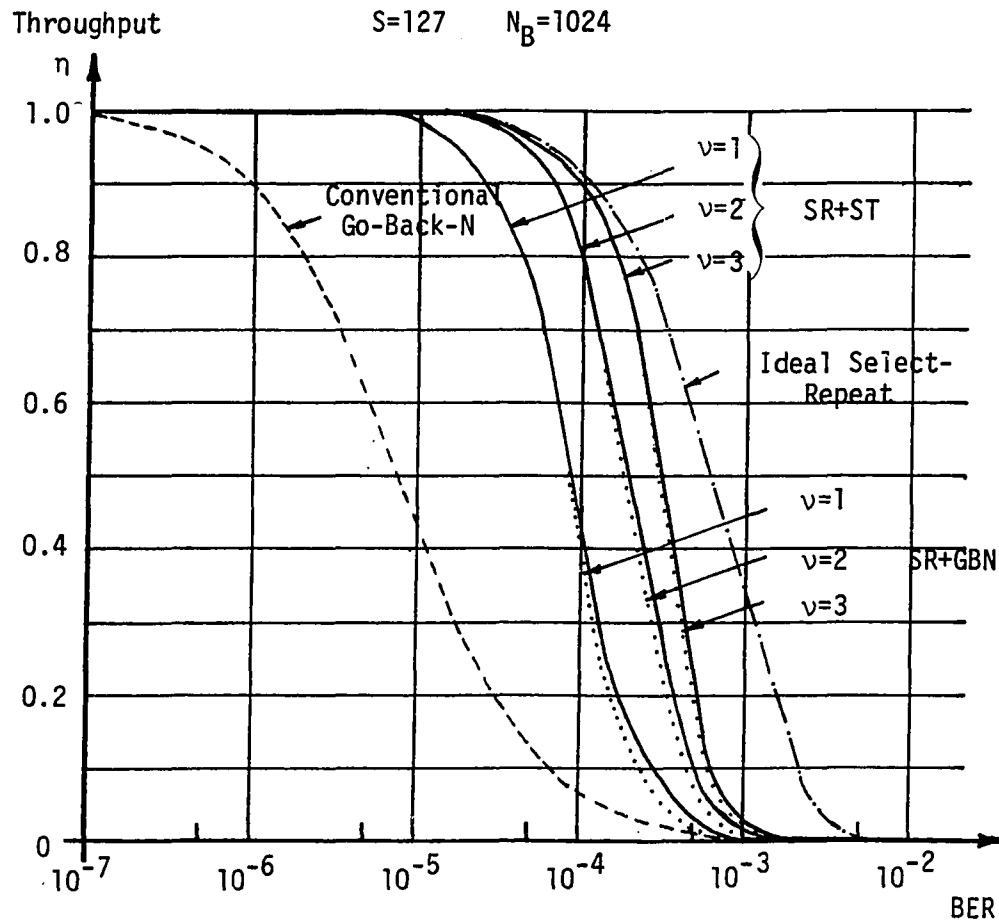


Figure 2.9

Throughput Performance for Various Values
of v (SR Mode Retransmission Attempts)

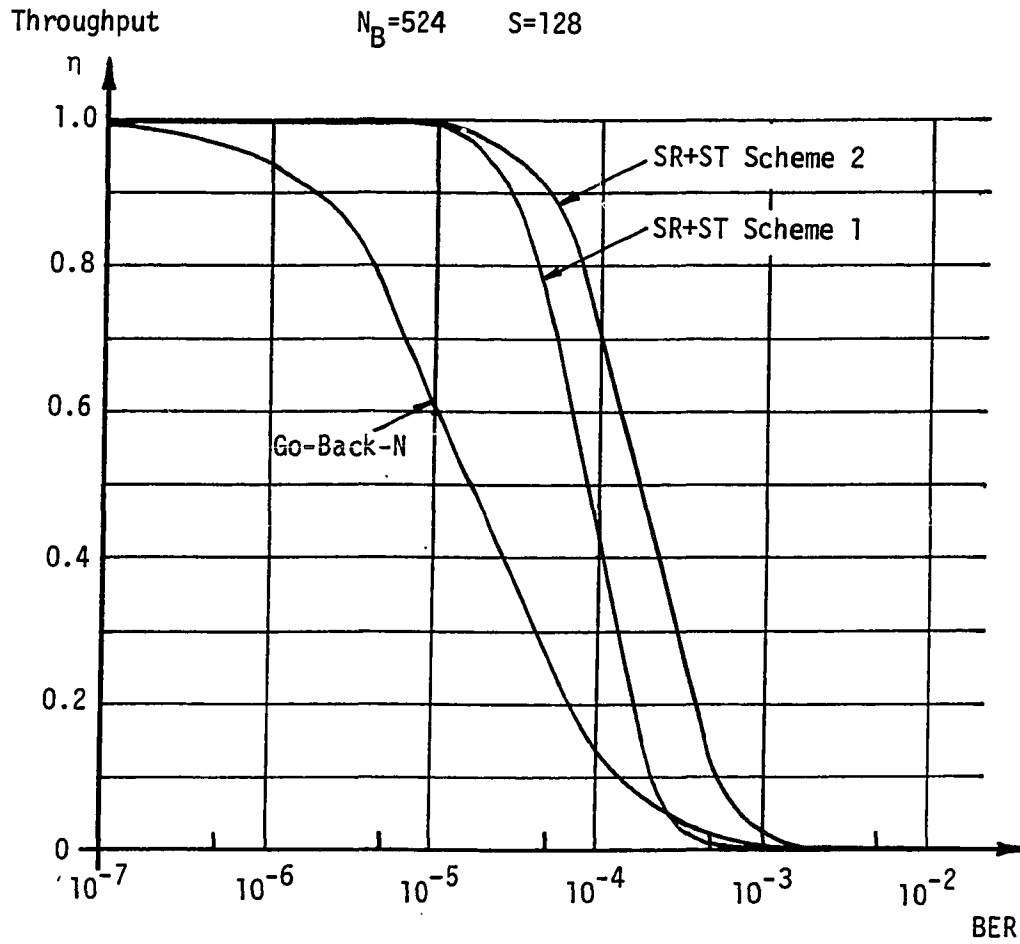


Figure 2.10

Comparison of SR + ST Scheme 2 With Others

better performance than its SR + GBN counterpart. However, as pointed out previously, this is at the expense of an increase in the required receiver buffer store capacity and block sequence number length; the latter requiring an extra overhead bit per block.

2.6 Comparison with ARQ Schemes by Other Authors

A number of other ARQ strategies have been suggested. These fall into three classes: selective-repeat ARQ, Go-Back-N ARQ, and Stop-and-Wait ARQ. Because of its inherent idle time, the latter is unsuitable for systems where the transmission rate is high or the round-trip delay is long. It is, therefore, not considered further here.

Table 2.2 summarizes ARQ schemes suggested in the literature for the other two ARQ categories. Where possible, the table lists expressions for the throughput (η) of each scheme.

Considering first the Go-Back-N (GBN) schemes, the first describes a simple form of continuous ARQ based on CCITT Rec. V41. When an error is indicated, the transmitter completes transmission of the current block and then goes back two blocks and repeats. It is assumed that the block lengths are large enough to allow for the round-trip transmission time. Sastry's scheme uses a stutter (ST) mode of retransmission for all NAK'ed blocks.

Morris' scheme uses the same ST mode of retransmission but a buffer is provided at the receiver to store the good data blocks following an erroneously received block.

The SETRAN scheme proposed by Lin and Yu also stores good data blocks at the receiver following an error, but the scheme has a different

Table 2.2
ARQ Schemes by Other Authors

GO-BACK-N SCHEMES			
Author	Ref. No.	Throughput	Comments
Gatfield Go-Back-2 (CCITT Rec. V41)	[14]	$\eta = \frac{1 - P_B}{1 + P_B}$	Only useful if block length is longer than the equivalent round-trip time (low data rates on satellite channel)
Sastry	[17]	$\eta = \frac{1 - P_B}{1 + 2SP_B(1 - P_B)}$	η greater than conventional GBN only when $P_B \geq 0.5$.
Morris	[21]	$\eta = \frac{1 - P_B}{1 + SP_B(1 - P_B)}$	η approximately same as for conventional GBN--slight improvement for high P_B .
Lin and Yu (SETRAN)	[31]	lower bound given in reference [31]	Outperforms first three schemes. See Figure 2.11.
Towsley (Stutter Go-Back-N)	[28]	$\eta = \frac{1 - P_B}{1 + SP_B}$	η same as for conventional GBN but input queue lengths decreased where system has idle periods.
SELECTIVE REPEAT SCHEMES			
Yu and Lin	[32]	η lower bounded in reference [31]	See Figure 2.12

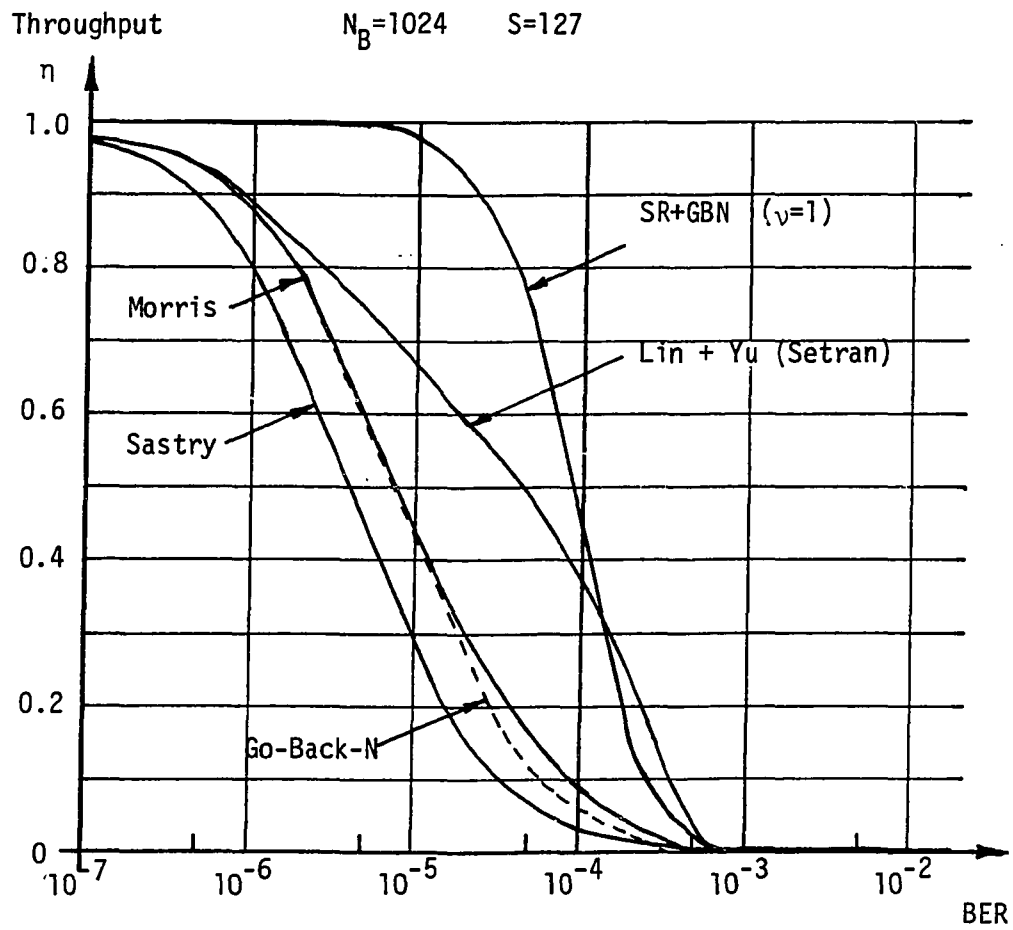


Figure 2.11

Comparison with Go-Back-N Schemes

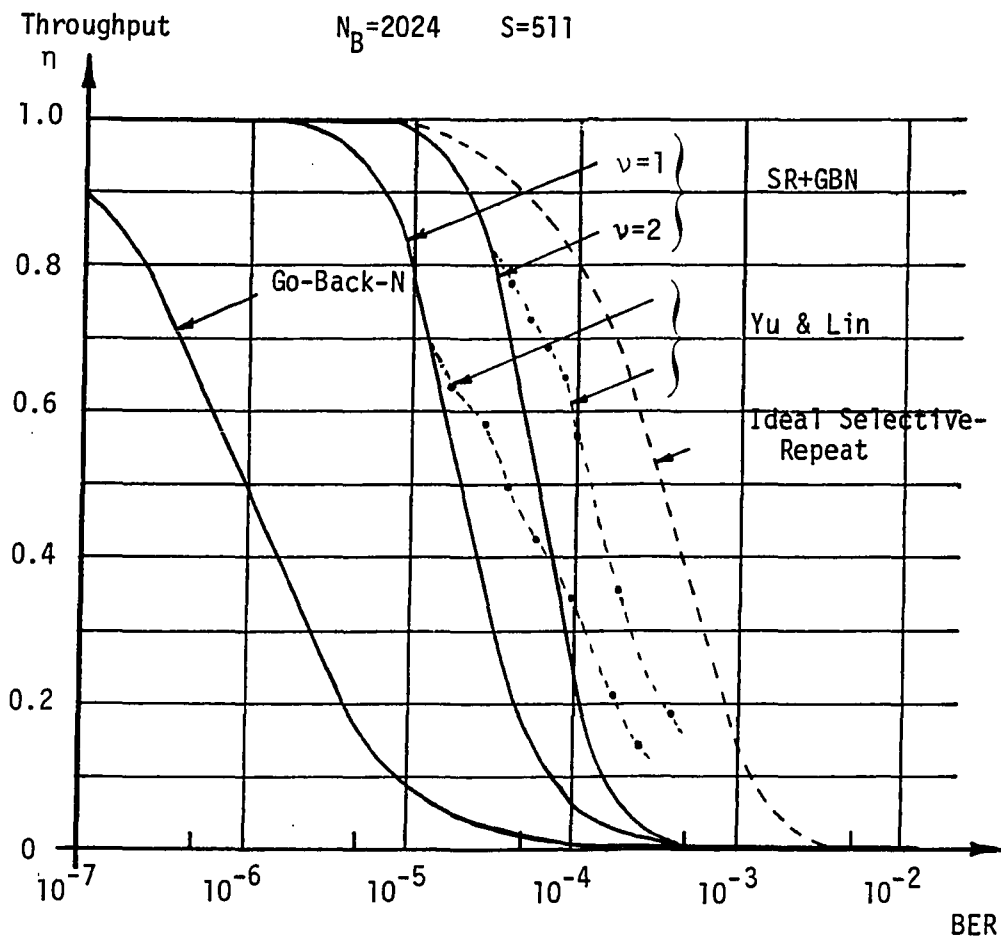


Figure 2.12

Comparison with Yu and Lin Selective-Repeat Scheme

retransmission strategy. Essentially, after an NAK has been received, use is made of the transmission time slots corresponding to the successfully received data blocks to repeat the first erroneously received block.

The Towsley scheme is designed for use on systems which are not heavily utilized so that the input queue has empty periods. During these periods, the previously transmitted block is retransmitted in an ST mode until either a new block arrives at the transmitter or an NAK returns for an earlier block. Throughput calculations imply systems have saturated inputs. In this case, Towsley's scheme becomes the conventional GBN.

The only Selective-Repeat scheme shown is that due to Yu and Lin for which exact analysis is not available. However, a lower-bound provides a useful basis for comparison. The scheme essentially provides for SR mode retransmission with a strategy that avoids overflow of a receiver buffer of size $S+1$ blocks.

Figures 2.11 and 2.12 compare the throughput performance of the above schemes with those described in Sections 2-5.

2.7 Summary

The class of mixed-mode ARQ models considered here has superior throughput efficiency to that of Go-Back-N schemes such as those proposed by Sastry [17], Morris [21], Towsley [28], or Lin and Yu [31].

The performance of the SR + ST and SR + GBN schemes demonstrates the general proposition that for best ARQ performance, at least the first retransmission of a block following an error should be in the selective-repeat mode.

Providing this is done, markedly superior throughput performance is achieved over Go-Back-N schemes for channels with high bit rate and large round-trip delay. This improvement is at the expense of required logic complexity and buffer storage provision in the transmitter and receiver.

For related SR + ST and SR + GBN schemes (that is, those with the same number ν of SR retransmissions), the throughput performance is very nearly the same indicating that the choice of the secondary retransmission mode (ST vs GBN) does not have a significant bearing on the throughput. The SR + ST scheme does, however, require the provision of more receiver buffer store than the SR + GBN.

An alternative SR + ST scheme (2) has been shown to require simpler tx/rx logic structure and provides useful performance up to bit-error-rates just above 10^{-4} (for $N_B=524$ and $S=128$). As illustrated in Figure 2.9, the other SR + ST and SR + GBN schemes for $\nu = 1$ are shown to have a useful throughput up to bit-error-rates approximately twice as great as the SR + ST scheme 2.

One feature of the method of analysis of these schemes is that it provides results for evaluating the performance in cases where the number of selective-repeat type retransmissions is greater than 1. Results have shown how such schemes can be used to give performances approaching the ideal selective-repeat scheme (infinite buffer) as the number of SR-type retransmissions is increased.

3. PARITY RETRANSMISSION ARQ SCHEMES

3.1 Introduction

It is clear from the results of Section 2 that, among all ARQ schemes, the Selective-Repeat ARQ protocol provides an upper bound on the throughput efficiency performance that can be obtained in the presence of channel errors and delays.

Mixed-mode and other ARQ schemes with initial retransmission attempts in a selective-repeat mode provide a practical means of obtaining throughput performance approaching the upper bound without excessive receiver buffer size. While such schemes offer considerable improvement over the conventional Go-Back-N ARQ, nevertheless for bit-error-rates greater than 10^{-4} , their throughput may fall off rapidly on satellite or long terrestrial circuits as shown in Figures 2-8 to 2-12.

Hybrid schemes involving the use of forward-error correction (FEC) as well as ARQ offer a potential means of obtaining throughput performance better than the selective-repeat upper bound for schemes using ARQ alone. As pointed out in Section 1.1, Type-II hybrid ARQ schemes in which parity bits for error correction are only transmitted when they are needed seem to offer the best solution. In a sense these schemes are adaptive. Their common feature is that when errors are detected in a particular block, parity bits for forward-error correction are transmitted rather than just retransmitting the original information block. Hence, they are referred to as Parity Retransmission ARQ schemes.

In this section the principles of such schemes will be outlined. Methods for representing them using state diagrams will be introduced. Then an approach to throughput analysis will be outlined which is intended to be sufficiently universal that it can be used for a number of different combinations of parity retransmission schemes and ARQ protocols.

To introduce the essential ideas, a Type-II hybrid scheme is first described using a half-rate block code for error correction. Convolutional codes may, however, provide a very competitive alternative to the FEC problem for hybrid ARQ systems. Methods for implementing hybrid schemes using convolutional codes are therefore emphasized in this and the following sections.

3.2 Alternate Data-Parity Hybrid ARQ

A Type-II hybrid scheme suggested by Lin and Yu [37] provides for alternate repetitions of data and correction parity based on a block code. Particular attention is given in this dissertation to the use of convolutional codes for forward-error correction. One possible Type-II hybrid scheme can be illustrated as follows. If the first transmission of a k -bit message u and $n-k$ parity bits $P_0(u)$ (based on code C_0) results in detected errors, the receiver stores the received message bits \hat{u} and requests a retransmission. The transmitter does not retransmit the message u but instead transmits k parity bits $P_1(u)$ based on the message u and a half-rate FEC code C_1 .

Additionally, the $n-k$ parity detection bits $P_0(P_1(u))$ based on the code C_0 are transmitted. In this case, the receiver first uses the code C_0 parity to determine whether this second transmission appears error

free. If so, then the receiver recovers the wanted message \tilde{u} from the parity $P_1(u)$ by inversion. In order to make this possible, the code C_1 must be "invertible" [34]; that is the k information digits can be determined from knowledge of only the k parity check digits.

When errors are detected in the second transmission, the receiver attempts error correction using the received parity $\hat{P}_1(u)$ and the previously stored message \hat{u} . If the errors in $\{\hat{P}_1(u), \hat{u}\}$ are correctible, the estimated message \tilde{u} is delivered to the user. If not, the receiver stores the parity word $\hat{P}_1(u)$ and requests a retransmission.

For the third and subsequent transmission attempts, alternate repetitions of the information vector u and the parity vector $P_1(u)$ are sent until one pair is correctible or no errors are detected in a \hat{u} or $\hat{P}_1(u)$.

This strategy can be incorporated with any of the basic forms of ARQ (Go-Back-N, Selective-Repeat or Mixed-Mode ARQ). No extra overhead is required. When the channel error rate is low, it has the same throughput efficiency as its corresponding ARQ scheme. With higher channel error rates, the error correction provided by the half-rate code C_1 will provide higher throughput.

In this section, a unified method of analysis is presented which facilitates the comparison of performance of convolutional or block codes used for FEC with a number of different ARQ protocols. The results of this analysis indicate that convolutional codes compare favorably with block codes in this application. Furthermore, although a Type-I hybrid scheme is appropriate for a Go-Back-N ARQ protocol, Type-II hybrid schemes provide significantly better throughput performance with a selective-repeat type of ARQ protocol such as a Mixed-Mode scheme.

The application of convolutional codes for FEC parity retransmission perhaps needs further explanation. It is, of course, true that for a (n_c, k_c) convolutional code, the n_c and k_c parameters are typically small integers such as 1, 2, 3, or 4 (with $k_c \leq n_c$) but large encoder memory order (m) ensures good error correction capability. Use of a $(2, 1)$ code, say, in a Type-II hybrid scheme would envisage that the ARQ transmitter would send an n -bit block $\{u, P_0(u)\}$ where u represents $(k-m)$ data bits followed by m zeros (or known bits) and $P_0(u)$ is the parity for u based on a high rate block code C_0 as for the case previously described.

At the same time, the transmitter would compute and store the k parity bits generated by the $(k-m)$ data bits and a systematic $(2, 1)$ convolutional code C_1 . The use of a "tail" of m known bits at the end of the data block is to ensure that the C_1 encoder memory is allowed to clear when generating the k parity bits.

These parity bits may now be used in a parity retransmission scheme as described above using alternate repetitions of data and parity. If the first transmission of a data block is received with detectable errors, a retransmission is requested. The transmitter sends $\{P_1(u), P_0(P_1(u))\}$ where the k parity bits $P_1(u)$ are generated by the systematic $(2, 1)$ convolutional code. The $n-k$ bits $P_0(P_1(u))$ are generated by the high-rate block code C_0 . At the receiver the data may be recovered from the received parity bits $\hat{P}_1(u)$ if the syndrome calculated using code C_0 is zero. Codes for which the k data bits can be uniquely determined from knowledge of only the k parity bits are said to be "invertible". If the syndrome is non-zero, error-correction is attempted using the received data \hat{u} and parity $\hat{P}_1(u)$. Fortunately, all systematic $(n, 1)$ convolutional codes are invertible. Let $u(X)$ represent in

polynomial form the input data sequence to a systematic (n,1) encoder. The encoding can be described [2] in terms of a generator matrix $\bar{G}(X)$ of the form

$$\bar{G}(X) = [1 \ g^{(2)}(X) \ g^{(3)}(X) \ \dots \ g^{(n)}(X)]$$

The n-tuple of output sequences

$$\bar{y}(X) = [y^{(1)}(X), \ y^{(2)}(X), \ \dots \ y^{(n)}(X)]$$

is obtained using

$$\bar{y}(X) = u(X) \bar{G}(X)$$

For the systematic (2,1) code described above, the parity bits $P_1(u)$ are given in polynomial form by

$$y^{(2)}(X) = u(X) g^{(2)}(X)$$

Hence, if we know $P_1(u)$ and the code generator $g^{(2)}(X)$ it is possible to obtain the data sequence $u(X)$ since no two code words will have the same parity check bits. Likewise for a (3,1) systematic code knowledge of the set of parity bits.

$$y^{(3)}(X) = u(X) g^{(3)}(X)$$

can provide the original data $u(X)$ by an inversion circuit. Other ARQ schemes using related systematic (2,1) and (3,1) convolutional codes have also been considered and will be discussed in the following sections.

The consideration of the use of convolutional codes in a Type-II hybrid scheme is motivated by at least two factors. Ensemble error rate bounds of the form

$$P_E \leq e^{-NE(R)}$$

have been derived (see for example, Viterbi and Omura [38]) for which the error exponent $E(R)$ for memoryless channels is larger for convolutional

codes than for block codes. (Strictly, this is on the basis of maximum likelihood decoding in each case, for which the computational complexity would be slightly greater for convolutional codes.)

A second reason why convolutional codes might be preferred in hybrid ARQ applications relates to the desirability of minimum complexity encoders and decoders. Threshold decodable codes, while not providing maximum error-correction efficiency, nevertheless, involve much simpler decoding equipment and minimum decoding delays. Since in an ARQ application, a retransmission facility is always available it is sensible from a practical point of view to choose the simpler threshold decoding than the more efficient Viterbi or sequential decoding schemes with considerable increase in complexity and decoding delays. Alternatively, if codes with moderate values of memory order (m) are used, current read-only-memory (ROM) technology development suggests that direct look-up decoding may be feasible. For example, there exists a rate one-half convolutional code for which all incorrect paths are at distance 7 from the correct path within 11 branches in the coding trellis. This offers the possibility of correcting any error pattern of up to three errors in any sequence of 22 symbols. This could be implemented in table look-up form by the use of currently available read-only-memories (ROM's) providing 2^{11} bits of storage capacity.

3.3 Hybrid Schemes and Their State Diagrams

It is useful to apply graphical procedures for representing succeeding events in ARQ schemes. The use of a state-diagram and probability transition matrix representation assists in the analysis of performance of a general class of ARQ schemes including the Type-II hybrid schemes

employing either block codes or convolutional codes for forward-error-correction. It is also applicable to Type I schemes or to conventional ARQ schemes without forward-error correction. This is illustrated for several ARQ schemes.

Alternate data-parity hybrid ARQ. Consider an ARQ retransmission procedure as described above in which

- The initial transmission of any block (u) of data consists of the n vector $\{u, P_0(u)\}$ containing the original data (k bits) and $n-k$ parity check bits $P_0(u)$ generated by a high rate block code C_0 used for forward error detection only.
- The first retransmission of that data block (initiated whenever errors are detected in the initial transmission) consists of the n -vector $\{P_1(u), P_0(P_1(u))\}$. The first k bits $P_1(u)$ are the parity bits generated for the original data (u) by a one-half rate forward-error-correction code C_1 . C_1 may be either an invertible block code or a systematic convolutional code. The $n-k$ bits $P_0(P_1(u))$ are the parity bits computed when $P_1(u)$ is the encoder input for the forward error detection code C_0 .

The receiver attempts to recover the data u either by inversion (if the first retransmission was received without errors being detected) or by error-correction (if errors were detected at the receiver). Otherwise, a further retransmission is requested.

- The second retransmission consists of $\{u, P_0(u)\}$, a repeat of the first transmission. Again the receiver attempts to recover u either directly (if no errors are detected in the

last received block) or by error correction based on the k bits $\hat{P}_1(u)$ received in the first retransmission and the k bits \hat{u} just received.

- Subsequent retransmissions of alternatively a parity block $\{P_1(u), P_0(P_1(u))\}$ or a data block $\{u, P_0(u)\}$ are continued until the receiver sends an ACK for the block. This will be done if a $\hat{u}, \hat{P}_1(u)$ pair is correctible or the receiver detects no errors in a \hat{u} or a $\hat{P}_1(u)$. Then the receiver output is the estimated message \tilde{u} . The State Transition diagram of Figure 3.1 describes the possible outcomes in the transmission/retransmission and data recovery process.

All possible transitions are shown in Figure 3.1. States 1 and 2 represent the possible outcomes associated with the initial transmission of any block i . If the process arrives at State 1, the data in block i , is successfully recovered and another initial transmission (of block $i+1$) is initiated resulting in States 1 or 2. If the process arrives at State 2 (block $i+1$ is received with detectable errors) then that block must be retransmitted so the process moves to one of the States 3, 4 or 5. Transition lines represent successive outcomes for a given block number (irrespective of whether it is a "data" or "parity" transmission of that block). Dotted transition lines represent cases where a new block is transmitted. Solid lines represent "unsuccessful" transmissions.

If C_0 is chosen properly, the probability P_e of an error being undetected is very small and is upper-bounded [2] by

$$P_e \leq [1-(1-p)^k] 2^{-(n-k)} \quad (3.1)$$

In this section P_e is assumed to be negligible.

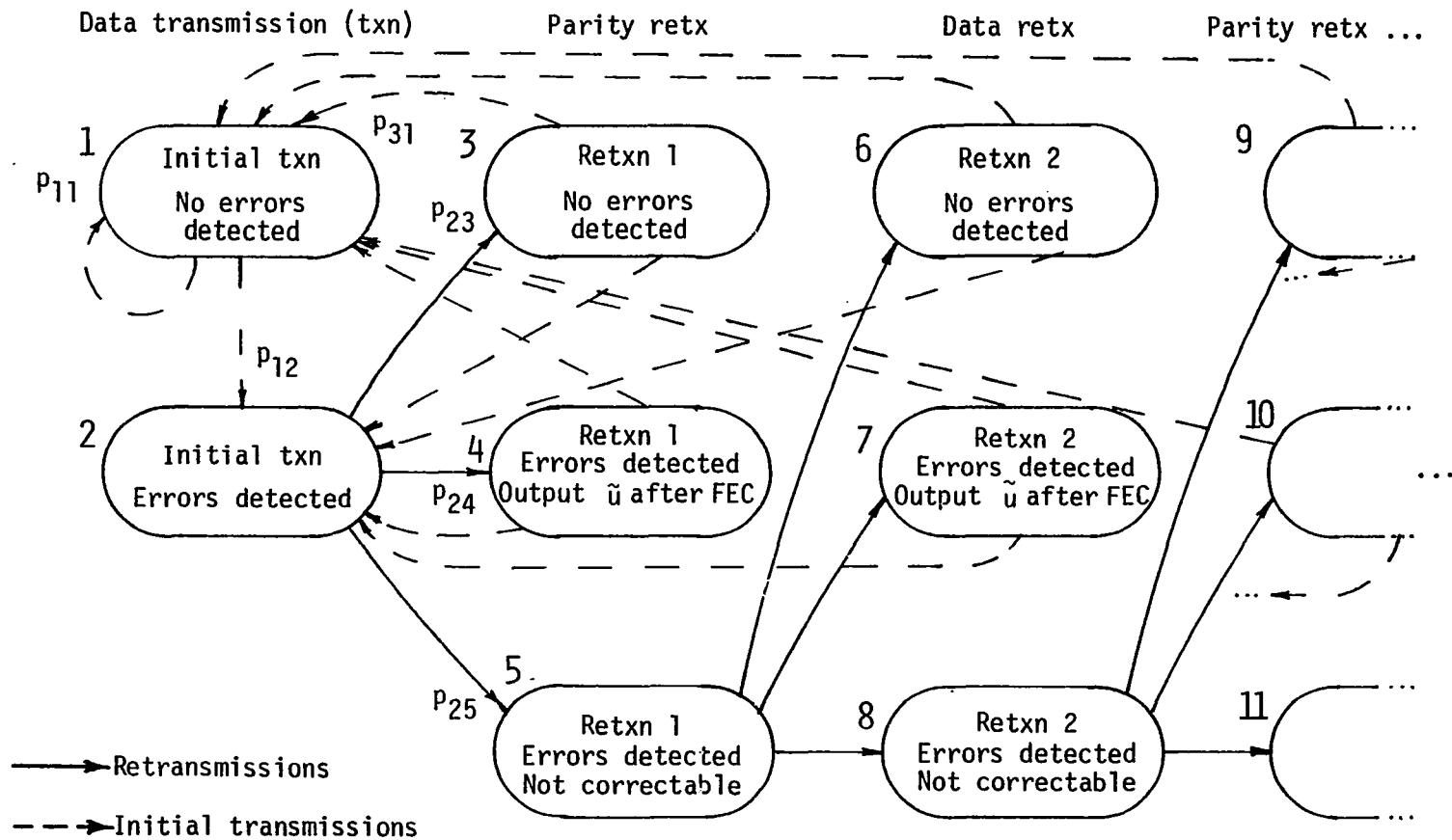


Figure 3.1

State Transition Diagram for Alternate Data/Parity Hybrid ARQ

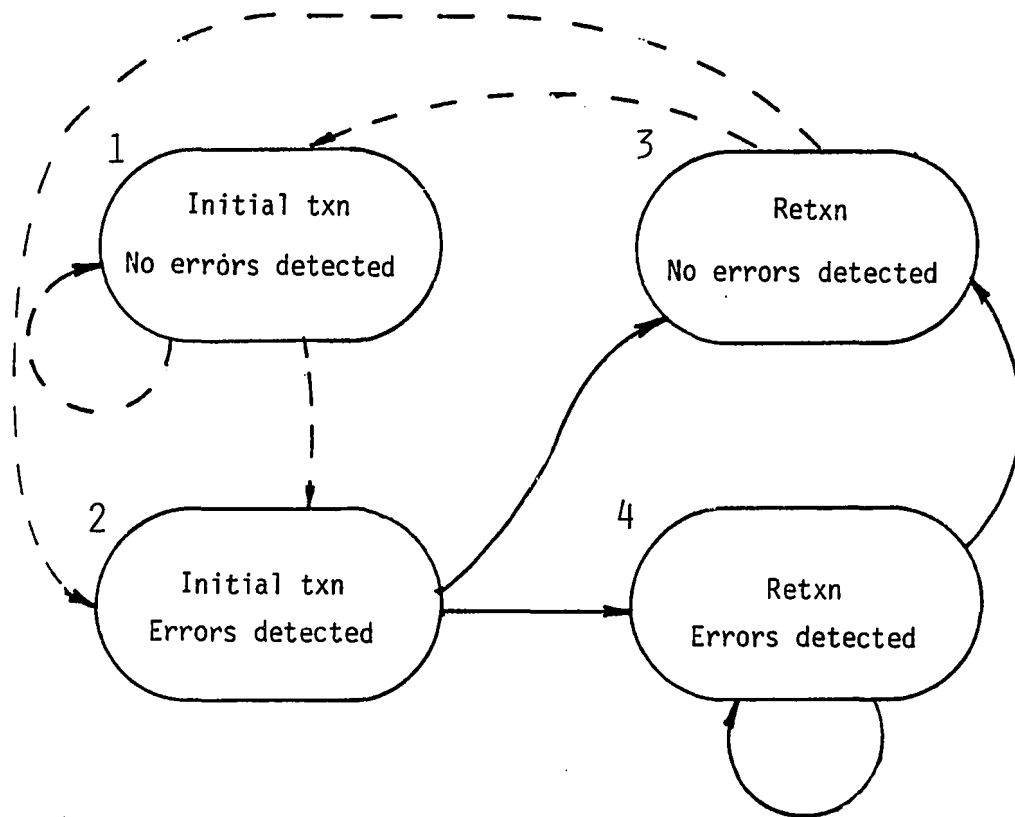


Figure 3.2

State Diagram for Conventional ARQ

The associated Probability Matrix for this case is of the form

$$[\Pi] = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 \\ 0 & 0 & p_{23} & p_{24} \\ p_{31} & p_{32} & 0 & 0 \\ 0 & 0 & p_{43} & p_{44} \end{bmatrix} \quad (3.3)$$

Type-I Hybrid ARQ. For Type-I hybrid schemes, forward-error correction is incorporated into each transmission of a data block. The transition diagram is shown in Figure 3.3. Its probability matrix is of the form

$$[\Pi] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 \\ p_{21} & p_{22} & p_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{34} & p_{35} & p_{36} \\ p_{41} & p_{42} & p_{43} & 0 & 0 & 0 \\ p_{51} & p_{52} & p_{53} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{64} & p_{65} & p_{66} \end{bmatrix} \quad (3.4)$$

Rate 1/2, 1/3 Type-II Hybrid ARQ. Another Type-II hybrid ARQ scheme is also proposed. It uses a 1/2 rate convolutional code C_1 and a related 1/3 rate convolutional code C_2 for forward error correction. It is illustrated by Figure 3.4.

As in the previous case, the initial transmission of any block is a conventional k -bit data block plus parity for error detection based on a high rate code C_0 . If a retransmission of the block is required, the first retransmission used is a "parity" retransmission based on the k parity bits generated by a half rate code C_1 as for the alternate data/parity scheme of the previous section. If this retransmission

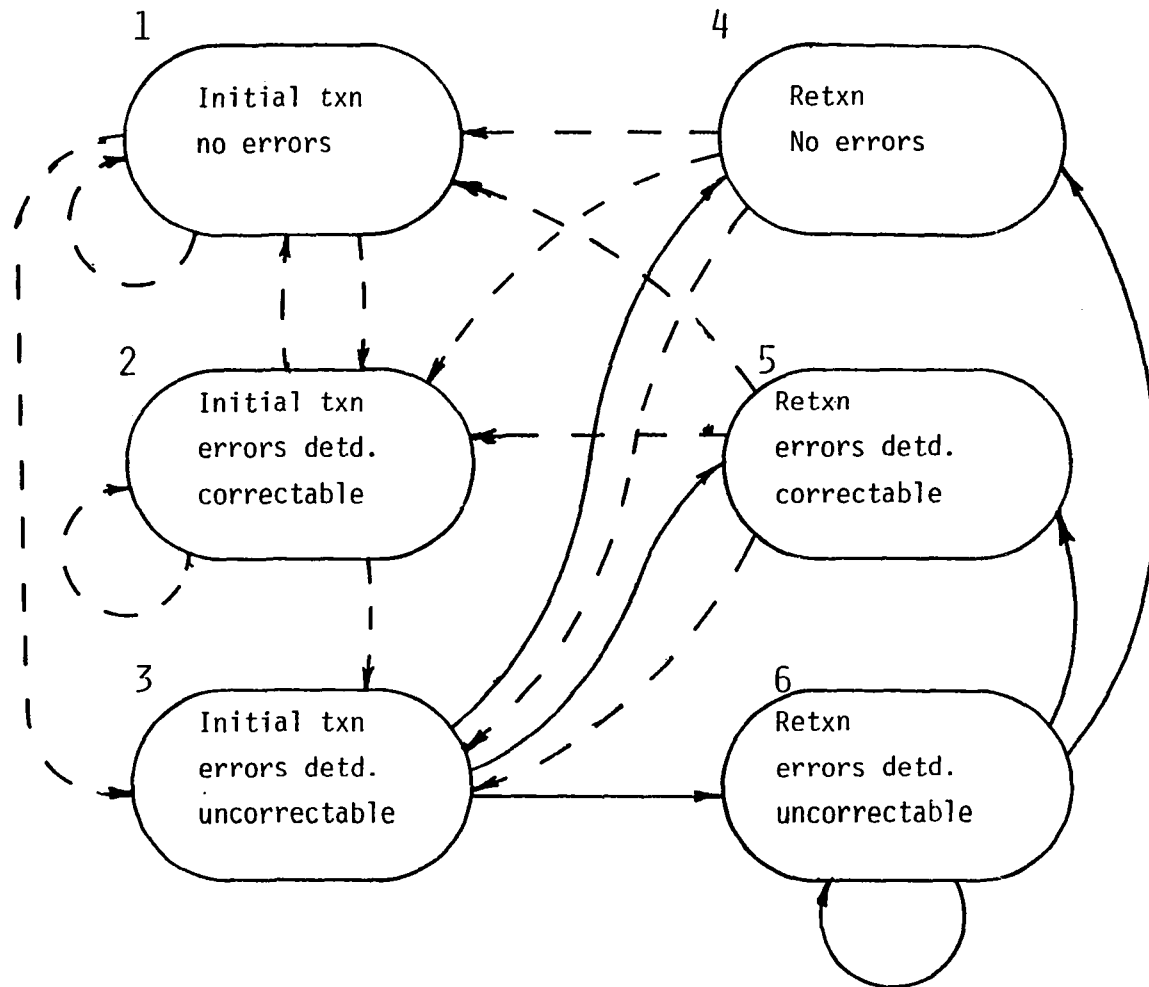


Figure 3.3

State Diagram for Type-I Hybrid ARQ

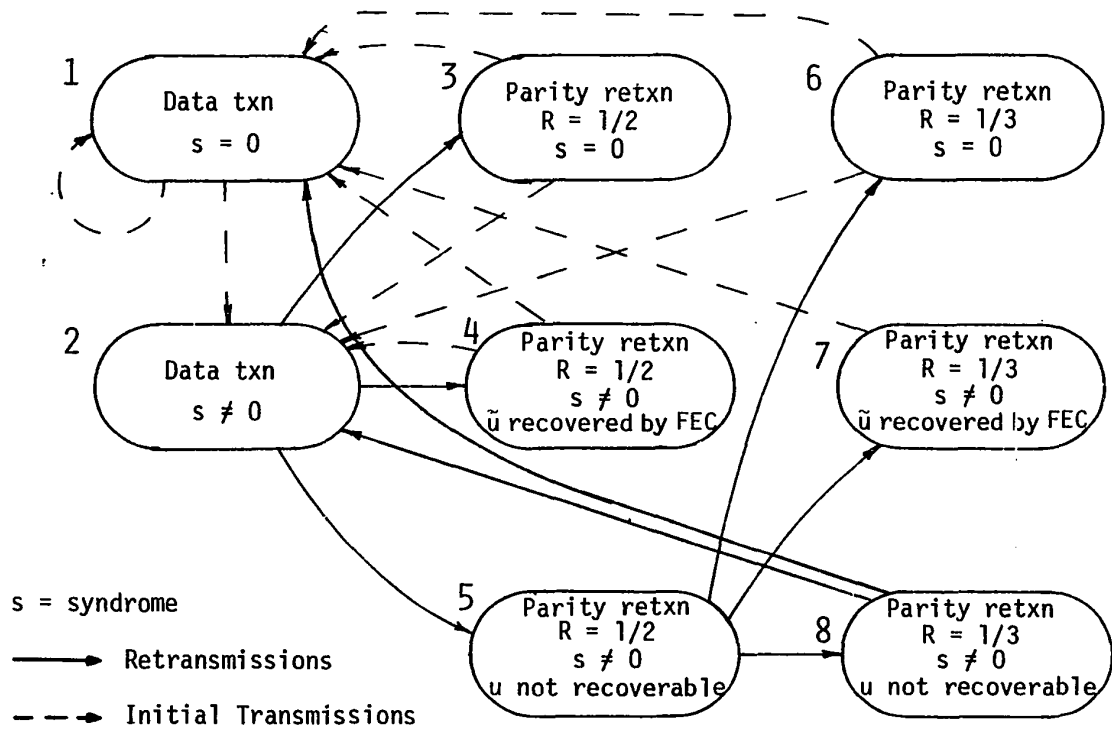


Figure 3.4
 State Diagram for $\frac{1}{2}/\frac{1}{3}$ Rate Hybrid ARQ

fails to recover the data successfully (either by inversion or decoding) then for the next (third) transmission for that block, k parity bits generated by a $1/3$ rate code C_2 are used. For this purpose a $(3, 1)$ Code C_2 and a "related" $(2, 1)$ code C_1 must be available such that C_2 has generator polynomials closely related to that of C_1 . C_1 and C_2 must be systematic codes related in such a way that for each information bit, the first parity bit generated by C_2 is the same as the first parity bit generated by C_1 .

Construction procedures for finding such codes will be discussed in a later section.

If the related half-rate code C_1 and one-third rate code C_2 are used, the error-correction process could be as follows. After the first transmission attempt the receiver computes the syndrome using code C_0 . If the syndrome is zero, the data is assumed error-free and delivered to the user. If not, a retransmission is requested. This will be a "parity" transmission $\{P_1(u), P_0(P_1(u))\}$ where the k parity bits $P_1(u)$ are generated using the systematic convolutional code C_1 . The $n-k$ bits $P_0(P_1(u))$ are derived from code C_0 . At the receiver, the syndrome for the received block is computed using code C_0 . If it is zero, the received parity bits $\hat{P}_1(u)$ are assumed error-free. An inversion circuit is used to recover the data estimate \tilde{u} from $\hat{P}_1(u)$. If not, a decoder based on code C_1 is used to attempt to correct all the errors in the first block data bits \hat{u} using the $2k$ bits \hat{u} and $\hat{P}_1(u)$. If this fails, a retransmission is again requested. For the third transmission, the transmitter uses the one-third rate code C_2 and sends n bits $\{P_2(u), P_0(P_2(u))\}$ where $P_2(u)$ is the set of second parity bits generated for each data bit by the $(3, 1)$ code C_2 . The $n-k$ bits $P_0(P_2(u))$ are

generated using code C_0 . At the receiver, the process is similar to that after the second transmission. If the syndrome using C_0 is zero, the data \tilde{u} is obtained from $\hat{P}_2(u)$ by inversion. If not, error-correction is attempted using a decoder for the (3, 1) code C_2 . The inputs to the decoder are the bits \hat{u} , $\hat{P}_1(u)$ and $\hat{P}_2(u)$ respectively. If the decoder is unable to correct all the errors in \hat{u} , the receiver discards \hat{u} , $\hat{P}_1(u)$ and $\hat{P}_2(u)$ and requests a retransmission. The transmitter sends a new data block of the form $\{u, P_0(u)\}$ as for the first transmission and the process is repeated as for transmissions 1, 2, and 3 until the data estimate \tilde{u} is recovered.

This form of Type-II hybrid ARQ may lead to improved throughput performance when the channel bit-error rate is very high. Furthermore, it will be shown in later sections that related codes C_1 and C_2 can be found such that both are orthogonalizable. This offers the advantages that a simple threshold decoding circuit could be used for both 1/2 and 1/3 rate decoding as will be discussed later.

Table 3.1 describes the parity retransmission procedure.

For this case the transition probability matrix is of the form

$$[\Pi] = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{23} & p_{24} & p_{25} & 0 & 0 & 0 \\ p_{31} & p_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{41} & p_{42} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{56} & p_{57} & p_{58} \\ p_{61} & p_{62} & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{71} & p_{72} & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{81} & p_{82} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.5)$$

Table 3.1
Rate 1/2, 1/3 Type-II Hybrid ARQ

Transmission	Type
1. Initial txn $\{u, P_0(u)\}$	Data plus error detection parity (C_0)
2. Retxn $\{P_1(u), P_0(P_1(u))\}$	1/2 rate C_1 parity plus C_0 detection parity
3. Retxn $\{P_2(u), P_0(P_2(u))\}$	1/3 rate C_2 parity plus C_0 detection parity
4. Retxn $\{u, P_0(u)\}$	Data plus error detection parity (C_0)
5. Retxn $\{P_1(u), P_0(P_1(u))\}$	1/2 rate C_1 parity plus C_0 detection parity
6. Retxn $\{P_2(u), P_0(P_2(u))\}$	1/3 rate C_2 parity plus C_0 detection parity
7. Retxn $\{u, P_0(u)\}$	Data plus error detection parity (C_0)
⋮	⋮

As an alternative error recovery procedure using codes C_1 and C_2 , a cyclic scheme could be used similar to the alternate data-parity scheme of Section 3.3. Consider the event that after three transmission attempts the receiver can still not recover the data using $\hat{u}, \hat{P}_1(u)$ and $\hat{P}_2(u)$. A new data block is then requested. The receiver could then retain $\hat{P}_1(u)$ and $\hat{P}_2(u)$ and combine this with the new version of the data \hat{u} to attempt error correction. If uncorrectable error patterns continue to require further retransmissions of new parity or data blocks, the receiver could use the most recent versions of $\hat{u}, \hat{P}_1(u)$ and $\hat{P}_2(u)$ if necessary to attempt error correction. This alternate error recovery procedure is difficult to analyze. The performance of the scheme described in Equation (3.5) will provide a lower bound on throughput for the alternative scheme.

3.4 Throughput Analysis Procedure

The state diagrams and transition probability matrix representation of the previous section provide an orderly approach to the throughput analysis of a wide range of ARQ schemes including Type-II ARQ. It was shown in Section 2 that for the Go-Back-N scheme or the mixed-mode schemes (SR + ST or SR + GBN) the throughput η can be computed using

$$\eta = \frac{1}{E\{M_i\}}$$

where M_i is a random variable representing the number of block transmission intervals required for any block i to be received successfully. This includes intervals in which block i was sent and also intervals when other blocks were sent but discarded by the receiver in the process of recovering block i . In order to compute η for these ARQ schemes over the random BSC, two steps are required, namely:

Step 1. Error Process. Specify the distribution of the random variable N_i where N_i is the number of transmissions required of block i for the data to be recovered without detectable errors. For example, for the BSC with an error-free feedback channel, N_i is geometrically distributed as

$$P_{N_i}(n_i) = (1-P_B) P_B^{n_i-1} \quad (3.6)$$

where

$$P_B = 1 - (1-p)^n$$

with n the block length and p the bit error rate.

Step 2. ARQ Protocol. Specify the functional relationship between the independent identically distributed M_i 's and the N_i 's. The functional relationship depends only on the particular ARQ protocol proposed. For example, for the SR + ST ARQ scheme with $v = 1$ selective-repeat retransmissions and then subsequent retransmissions being "stutter" repetition of the faulty block,

$$M_i = \begin{cases} N_i & N_i = 1, 2 \\ N_i + S & N_i = 3, 4, \dots \end{cases} \quad (3.7)$$

where S is the round-trip delay.

Once Steps 1 and 2 have been determined, it is a simple procedure to determine $E\{M_i\}$ and hence the throughput η .

In similar manner for a range of hybrid ARQ schemes, it is possible to derive expressions for throughput η . Step 2 remains the same as for conventional ARQ. However, Step 1 must be modified to provide information on the forward-error correction procedures as well as the channel error process.

Step 1. Error Process for Type-II Hybrid ARQ. Let the random variable N_1 be defined as for the previous cases. Consider the alternate data-parity scheme described in Section 3.2 and illustrated by the state diagram in Figure 3.1. We define the following conditional probabilities:

Let q_1 be the conditional probability that a message estimate \tilde{u} without detectable errors is recovered from the first received parity block $\hat{P}_1(u)$ (first retransmission) either by an inversion process (no errors in parity) or by a decoding process based on the code C_1 given that errors are detected in the first received block $\{\hat{u}, \hat{P}_0(u)\}$.

Let q_2 be the conditional probability that a message estimate \tilde{u} is successfully recovered from the second received information block $\{\hat{u}', \hat{P}_0'(u)\}$ (i.e. second retransmission) given that errors were detected in the first received information word $\{\hat{u}, \hat{P}_0(u)\}$ and the subsequently received parity word $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ and that the received parity word failed to recover \tilde{u} by decoding.

In a similar fashion we can define conditional probabilities q_3, q_4, \dots of successfully recovering \tilde{u} without error from the 3rd, 4th, ... retransmissions respectively given that this was not possible after any previous retransmission. In terms of the state-transition probabilities of Equation (1) we can write

$$q_1 = p_{23} + p_{24}$$

$$q_2 = p_{56} + p_{57}$$

and so on for q_3, q_4, q_5, \dots

Now if we had q_2, q_3, q_4, \dots equal then the infinite chain of Figure 3.1 could be collapsed into a finite-state first order Markov

chain. Unfortunately, this is not the case for the error correction/detection procedures described.

We therefore analyze an inferior system as follows:

For the inferior system, the transmission/retransmission sequence is the same. That is, first a data block is sent, then a parity block, if required, then alternate data and parity transmissions until the data estimate \tilde{u} is recovered without detectable errors. The error-control process for the inferior system is different from that of the alternate data-parity scheme of Section 3.2. For the inferior system the receiver operations can be summarized as follows:

After transmission 1, data block $\{u, P_0(u)\}$: Syndrome checking error detection only; if errors detected, request a retransmission.

After transmission 2, parity block $\{P_1(u), P_0(P_1(u))\}$: Check detectable errors in $\hat{P}_1(u)$. If not, invert to obtain \tilde{u} . If necessary, decode $\{\hat{u}, \hat{P}_1(u)\}$ to correct errors in \tilde{u} . If unsuccessful, request a retransmission.

After transmission 3, data block $\{u', P'_0(u)\}$: Check detectable errors in \hat{u}' . If necessary decode $\{\hat{u}', \hat{P}'_1(u)\}$ to correct errors in \hat{u}' . If unsuccessful discard $\hat{u}', \hat{P}'_1(u)$ and \hat{u}' and request a retransmission.

After transmission 4, parity block $\{P'_1(u), P_0(P'_1(u))\}$: Syndrome checking error detection only. If errors detected, request a retransmission.

After transmission 5, data block $\{u'', P''_0(u)\}$: Check detectable errors in \hat{u}'' . If necessary decode $\{\hat{u}'', \hat{P}''_1(u)\}$ to correct errors in \hat{u}'' . If unsuccessful, request a retransmission.

After transmission 6, parity block $\{P_1^u(u), P_0^u(P_1^u(u))\}$: Check detectable errors in $\hat{P}_1^u(u)$. If not, invert to obtain \tilde{u} . If necessary, decode $\{\hat{u}^u, \hat{P}_1^u(u)\}$ to correct errors in \hat{u}^u . If unsuccessful, discard $\hat{P}_1^u(u), \hat{u}^u$ and $\hat{P}_1^u(u)$ and request a retransmission.

The process continues as above for transmissions 7, 8, ... until the estimate of the data \tilde{u} is obtained without detectable errors. Note that the inferior system, if \tilde{u} is not recovered ever three transmission attempts, all three blocks are discarded. The receiver process reverts to error detection only for the next block. It will provide inferior throughput to the alternate data-parity scheme in which error correction is available if required after every retransmission attempt.

Since the event that u is recovered error-free after transmissions 4, 7, 10, ... is independent of the events that u was not recoverable after preceding retransmissions, this inferior system can be modelled as a Markov chain as shown in Figure 3.5. The transition probability matrix is of the form:

$$[\Pi] = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{23} & p_{24} & 0 & 0 \\ p_{31} & p_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{45} & p_{46} \\ p_{51} & p_{52} & 0 & 0 & 0 & 0 \\ p_{61} & p_{62} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.8)$$

where $p_{i1} = 1 - p_B$ and $p_{i2} = p_B$ for $i = 1, 3, 5$ and 6 .
 $p_{23} = q_1$ $p_{24} = 1 - q_1$
 $p_{45} = q_2$ $p_{46} = 1 - q_2$

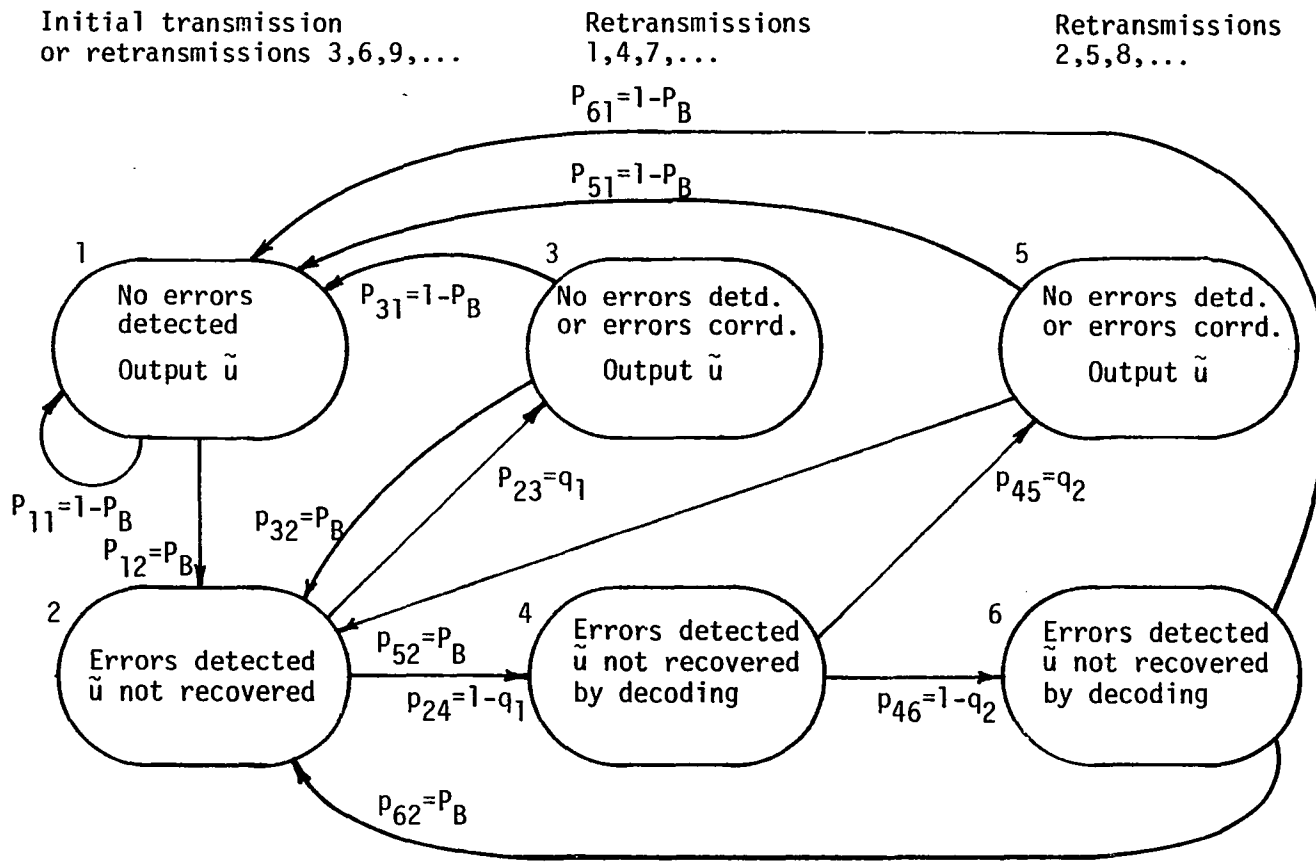


Figure 3.5

State Transition Diagram for Inferior Alternate Data-Parity Hybrid ARQ System (Markov Model)

Expressions for the probabilities q_1 and q_2 will be derived in a subsequent section.

We can now proceed to complete step (1) by finding the probability distribution for the identically distributed random variables N_i , $i = 1, 2, 3, \dots$

Consider those cases where the error generating/correcting mechanisms can be represented by a finite first-order Markov chain. This would include the hybrid schemes represented in Figures 3.4 and 3.5 but not that of Figure 3.1. In the latter case, a lower bound on throughput performance can be found using the inferior system represented in Figure 3.5.

Then the following procedure can be used to determine the probability distribution of the N_i 's taking into account a BSC and also the error-correction performance of the FEC code used in the hybrid scheme. Consider the state diagrams and transition matrices described in Section 3. Given that the system commences at state j , the probabilities of reaching any state k after a specific number of steps r is given by matrix multiplication as

$$[P_{jk}^{(r)}] = [P_D^{(0)}] [\Pi]^r \quad (3.9)$$

where $[P_D^{(0)}]$ is the initial probability matrix given

$$[P_D^{(0)}] = \begin{bmatrix} P(s_1) & 0 & 0 & 0 \\ 0 & P(s_2) & 0 & 0 \\ 0 & 0 & P(s_3) & 0 \\ & & \vdots & \end{bmatrix} \quad (3.10)$$

We now wish to use this approach to find the probability $P_{N_i}(n_i)$ of reaching a final state (recover an estimate of a data block without

detectable errors) in $r = n_i - 1$ steps given that the system commenced in an initial state (first transmission of the block).

Note that $P(s_i)$ is the initial probability of being in state i . To illustrate the method consider the inferior Type-II hybrid ARQ scheme using alternative data/parity retransmission using an SR + ST protocol. This scheme was described in Section 3.3 and illustrated in Figure 3.5.

For this case, the initial probability matrix $[P_D(0)]$ will be of the form

$$[P_D(0)] = \begin{bmatrix} (1-p)^n & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-(1-p)^n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.11)$$

since the first transmission of a block results in state 1 (no errors detected) with probability $(1-p)^n$ or in state 2 (errors detected) with probability $1-(1-p)^n$. Note $P_B = 1-(1-p)^n$.

Using Equations 3.9 and 3.11, we obtain the probabilities $P\{s_k/s_j\}^{(r)}$ of reaching s_k from s_j in r steps for $r = 1, 2, 3, \dots$ as

$$\begin{aligned} P\{s_k/s_j\}^{(1)} &= P\{s_j\}P\{s_k/s_j\} \\ P\{s_k/s_j\}^{(2)} &= P\{s_j\} \sum_{i=1}^6 P\{s_i/s_j\}P\{s_k/s_i\} \\ P\{s_k/s_j\}^{(3)} &= P\{s_j\} \sum_{i=1}^6 \sum_{g=1}^6 P\{s_i/s_j\}P\{s_g/s_i\}P\{s_k/s_g\} \end{aligned} \quad (3.12)$$

and so on.

We can use these expressions to find the probability distribution of the random variables N_i . For the hybrid ARQ schemes described, the N_i 's will be independent and identically distributed. Henceforth, the generic random variable N will be used to denote the number of transmissions for a block to be received without error.

We can find the probability $P_N(n)$ as follows:

$$P_N(n) = \sum_k \sum_j P\{s_k/s_j\}^{(n-1)} \quad (3.13)$$

where the sum is taken over all k for which s_k represents a final state and all j for which s_j represents an initial state. Also in evaluating the terms $P\{s_k/s_j\}^{(r)}$ care must be taken to exclude these sequences of states which include a final state (successful recovery of the information without detected errors) in any part of the sequence except the last. To explain further, consider as an example the problem of finding the probability $P_N(4)$ for the case described by Equations 3.8 and 3.11. This is done as follows

$$P_N(4) = \sum_k \sum_j P\{s_k/s_j\}^{(3)}$$

where

$$P\{s_k/s_j\}^{(3)} = P\{s_j\} \sum_{i=1}^6 \sum_{g=1}^6 P\{s_i/s_j\} P\{s_g/s_i\} P\{s_k/s_g\}$$

$j, i, g \neq$ final state number.

In Markov chain theory, formal methods exist for calculating the probability of first passage from state j to state k in n steps see, for example, reference [50]). However, for the cases treated in

this dissertation, these probabilities can be easily obtained from the state diagrams by inspection.

For this example, referring to Figure 3.5 , it is apparent that the cases where $j=2$ and $k=1$ are the only possible ones for $N=4$ (representing successful transmission of a block after an initial transmission plus three subsequent retransmissions). That is, s_2 and s_1 are the only possible initial and final states, respectively. Also, states s_3 and s_5 are final states not allowed for $N=4$.

Thus, for this example, the only 4 state sequence allowed is

$\{s_2, s_4, s_6, s_1\}$

$$\begin{aligned} P_N(4) &= P\{s_1/s_2\}^{(3)} \\ &= P\{s_2\} P\{s_4/s_2\}P\{s_6/s_4\}P\{s_1/s_6\} \end{aligned}$$

Thus,

$$P_N(4) = P_B(1-q_1)(1-q_2)(1-P_B) \quad (3.14)$$

3.5 Summary

It is possible to conceive of a large number of Type-II hybrid ARQ schemes. In general, such variations represent different tradeoffs between performance and complexity. In Section 3, a unified method of representing different schemes has been presented. It has been demonstrated that it is important to distinguish between two different aspects of any ARQ scheme. The first of these is the channel error and error correction characteristics of the scheme. This includes a description of the channel error process, for example, a random error model. It also should include a description of the forward error correction/detection procedures. These factors when combined together lead to a probability distribution $P_N(n)$ as described in Section 3.4. This process

is facilitated by the use of a state-diagram approach to the representation of each particular error correction/detection scheme.

The second aspect of the ARQ scheme that must be specified is the retransmission protocol. Possible retransmission protocols include the Go-Back-N, Selective-Repeat or the mixed-mode schemes described in Section 2. This second aspect can be characterized independent of the error process and error recovery system. It provides for each protocol a functional relationship between the generic random variables M and N as illustrated by Equation (3.7) for the SR + ST protocol.

The combination of these two aspects of any ARQ scheme (whether it be a hybrid or not) leads to a general unified approach to analysis of a wide variety of different systems. This will now be taken up in subsequent sections.

The use of state diagrams as proposed to represent successive outcomes of various forms of ARQ systems has interesting parallels. For example, directed graphs called Numerical Petri Nets (NPN's) have recently been found useful for representing communication protocols such as the CCITT Recommendation X.25 for packet switching networks [39]. The NPNs provide an orderly means of verification of the logical functioning of protocols, checking for deadlocks or loops and ensuring completeness, that is, the provision of all possible input signals. It is postulated that such state diagrams could be used in conjunction with the methods of this section--to study throughput characteristics of such systems and protocols.

State transmission diagrams similar in concept to those proposed in this section have also been used to study the performance of frame

alignment systems to evaluate the relative performance of different framing procedures (see, for example, reference [40]).

4. ANALYSIS OF TYPE-II HYBRID SCHEMES

In this section expressions for the throughput efficiency are derived for a number of different Type-II hybrid ARQ schemes which were described in Section 3. The results can be applied when either block or convolutional codes are to be used for forward-error-correction.

4.1 Alternate Data Parity Scheme with SR + ST Protocol

The error detection/correction scheme is illustrated in Figure 3.1 and represented in Equation (3.2). Now we combine that error correction procedure with an SR + ST ($v=1$) retransmission protocol as described in Section 2.2. Consider that a data block i $\{u, P_0(u)\}$ is transmitted for the first time. If errors are detected the transmitter stops transmitting new blocks and retransmits block i in the form of a parity block $\{P_1(u), P_0(P_1(u))\}$ in a selective-repeat mode. That is, the parity block is sent immediately following receipt at the transmitter of a NAK for that block. The parity transmission is then followed by new blocks (unless the next ACK/NAK indication received at the transmitter is a NAK). Refer to Section 2.2 for details. If the parity block i also contains errors and if the error correction procedure fails to correct all errors in the data another NAK is sent. The transmitter again stops sending new blocks and retransmits block i in a stutter mode. That is, block i is retransmitted repeatedly until an ACK is received indicating the data in block i has been recovered without detectable errors. The repetitive retransmission of block i is in the form of alternate data blocks of the form $\{u, P_0(u)\}$ or parity blocks

$\{P_1(u), P_0(P_1(u))\}$ respectively. As each retransmission of block i is received, the receiver attempts to recover the data as described in section 3.3. If a syndrome check indicates the received block is error-free the data estimate \tilde{u} is recovered either directly (if it was a "data" transmission) or by inversion (for a "parity" transmission). If the syndrome is non zero error correction is attempted using the most recently received data \hat{u} and parity $\hat{P}_1(u)$ words for block i . If the most recent retransmission of block i was a "data" transmission, the decoder uses the data bits \hat{u} from that block and the parity bits $\hat{P}_1(u)$ from the previous retransmission of block i . If the most recent transmission of block i was a "parity" transmission, the decoder input is the parity $\hat{P}_1(u)$ from that transmission and the data word \hat{u} from the previous transmission attempt of block i . Unfortunately, because this scheme cannot be represented by a finite Markov-chain, it is necessary to analyze an inferior system as illustrated in Figure 3.5 and represented in Equation (3.8). For this system the throughput will not be as high because no error-correction is attempted after certain retransmissions. The receiver recovers the data estimate \tilde{u} from transmissions 1, 4, 7, ... only if a syndrome calculation indicates the block is error-free. No error correction is performed. However for all other transmission attempts of a given block the receiver attempts to recover the data by error-correction if the syndrome is non zero. In the superior scheme error correction is available if required for every retransmission. We now proceed to find the probability distribution for the identically distributed random variables N_i for the inferior system. As explained in chapter 2, N_i is the number of transmissions of block i required to recover the data estimate without detectable errors.

Given that the system initial state is state 1 or state 2, the state sequences and associated probabilities shown in Table 4.1 represent the system operation for different values n_i of transmissions required to reach a final state (s_1, s_3 or s_5) where \tilde{u} is recovered without detectable errors. As explained in chapter 3, q_1 and q_2 are the conditional probabilities that an estimate \tilde{u} of the data is recovered without detectable errors after the second and third transmission attempts respectively given that \tilde{u} could not be recovered after earlier transmission attempts.

Hence we obtain the probability distribution $P_N(n)$ for the generic random variable N for the inferior system assuming a random error forward channel and error-free return ACK/NAK channel:

$$P_N(n) \begin{cases} (1-P_B)x^{(n-1)/3} & n = 1, 4, 7, \dots \\ P_B q_1 x^{(n-2)/3} & n = 2, 5, 8, \dots \\ P_B(1-q_1)q_2 x^{(n-3)/3} & n = 3, 6, 9, \dots \end{cases} \quad (4.1)$$

$$\text{where } x = P_B(1-q_1)(1-q_2)$$

This completes the description of the error mechanism and error correction aspect of the ARQ scheme. Next we combine this with a description of a particular ARQ protocol to find an expression for the throughput η .

For the inferior Type-II hybrid scheme with SR + ST protocol ($v=1$) described in Sections 2 and 3 we have from Equations (3.7) and (4.1).

Table 4.1

Probability Distribution Values for Number of Required Transmissions

n_i	Sequence	$P_{N_i}(n_i)$
1	s_1	$1-P_B$
2	$s_2 s_3$	$P_B q_1$
3	$s_2 s_4 s_5$	$P_B(1-q_1)q_2$
4	$s_2 s_4 s_6 s_1$	$P_B(1-q_1)(1-q_2)(1-P_B)$
5	$s_2 s_4 s_6 s_2 s_3$	$P_B^2(1-q_1)(1-q_2)q_1$
.	.	.
.	.	.
.	.	.

$$\begin{aligned}
 E\{M\} &= P_N(1) + 2P_N(2) + (S+3)P_N(3) + (S+4)P_N(4) + (S+5)P_N(5) + \dots \\
 &= 3(1-x) \sum_{i=0}^{\infty} ix^i + \{(1-x)(S+1) + P_Bq_1 + 2P_B(1-q_1)q_2\} \sum_{i=0}^{\infty} x^i \\
 &\quad - S(1-P_B + P_Bq_1)
 \end{aligned}$$

Since the throughput is the reciprocal of $E\{M\}$, by simplifying the above expressions we obtain the throughput (η) for this hybrid ARQ scheme as

$$\eta = \frac{1 - P_B(1-q_1)(1-q_2)}{1 + P_B + P_B(1+S)(1-q_1) - SP_B^2(1-q_1)^2(1-q_2)} \quad (4.2)$$

Now Equation (4.2) for the inferior scheme provides a lower bound on the throughput of the alternate data/parity hybrid scheme with SR + ST protocol.

It is an interesting check on this result to note that if no forward-error correction is used, the scheme reverts to a simple SR + ST ARQ scheme and it follows readily that

$$q_1 = q_2 = 1 - P_B$$

Substitution into Equation (4.2) leads to

$$\eta = \frac{1 - P_B}{1 + SP_B^2(1 - P_B)} \quad (4.3)$$

which is the result previously obtained for the SR + ST scheme in Equation (2.6).

Note also that this method of deriving an expression for throughput η has the advantage that it applies to a Type-II hybrid scheme with block or convolutional coding as the forward-error correction scheme. The derivation of the conditional probabilities q_1 and q_2 takes into account the type of code and decoding scheme used. This will be dealt with in a subsequent section.

4.2 Alternate Data-Parity Scheme with SR + GBN Protocol

It is also possible to find a lower bound for the throughput for the Alternate Data/Parity Type II hybrid scheme when combined with the SR + GBN protocol. In this scheme, the first retransmission is in a selective-repeat mode. Should the message \tilde{u} still not be recovered, further retransmissions are in the conventional Go-Back-N mode to avoid receiver buffer overflow.

It was shown that this protocol when used in conventional ARQ detection/retransmission scheme has only slightly inferior throughput performance to the SR + ST scheme but requires less receiver buffer store. A similar result could be predicted with Type-II hybrid operation.

For the SR + GBN scheme the number of transmission intervals M_i is equal to the number of transmission attempts N_i if the first or second transmissions are successful. Otherwise M_i is taken to include an additional S intervals for each extra transmission attempt. This is because on those occasions the receiver discards S other blocks each time to avoid buffer overflow. Hence, we have

$$M_i = \begin{cases} N_i & , \quad N_i = 1, 2 \\ N_i + S(N_i - 2) & , \quad N_i = 3, 4, \dots \end{cases} \quad (4.4)$$

As before, combining this with the hybrid scheme $P_N(n)$ distribution given by (4.1) we obtain a lower bound for the throughput η of the SR + GBN protocol with the alternate data/parity retransmission scheme as

$$\eta \geq \frac{1 - P_B(1-q_1)(1-q_2)}{1 + P_B + P_B(1+S)(1-q_1) + SP_B(1+P_B)(1-q_1)(1-q_2)} \quad (4.5)$$

As a check, let

$$q_1 = q_2 = 1 - P_B$$

which is then equivalent to the conventional SR + GBN ARQ without forward-error correction.

Then

$$\eta = \frac{1 - P_B}{1 + SP_B^2}$$

as given previously in Equation (2.11) for the conventional SR + GBN protocol.

4.3 Alternate Data-Parity Scheme with Selective-Repeat Protocol (Infinite Buffer)

In each of the retransmission protocols (SR + ST and SR + GBN) analyzed in the previous sections it was assumed that only the first retransmission could be in the selective-repeat mode (i.e. $v=1$). Other retransmissions if required, were assumed to be in the Stutter mode or Go-Back-N mode respectively. However as pointed out in section 2.4, the number v of allowable selective-repeat types of retransmissions can be increased providing appropriate extra buffer store is available at the receiver. As v is increased so will the throughput performance at high bit-error rates.

In order to examine the limits to which the throughput might increase, consider the calculation of throughput for a Type-II hybrid scheme using a purely selective-repeat protocol with infinite receiver buffer to prevent overflow. The error correction scheme analyzed here is the "inferior" alternate data-parity scheme for error correction used in the previous two cases and represented in Equation (3.8). Combining this with the fact that for the selective-repeat protocol, the number of transmission intervals M_i is equal to the number of transmission attempts N_i .

We obtain the throughput as

$$\eta = \frac{1 - P_B(1-q_1)(1-q_2)}{1 + P_B + P_B(1-q_1)} \quad (4.6)$$

As a check, letting $q_1=q_2=1-P_B$, we obtain the selective-repeat ARQ scheme without FEC and Equation (4.6) becomes

$$\eta = 1 - P_B$$

as obtained in section 2 for the selective-repeat ARQ scheme.

4.4 Lin and Yu Hybrid ARQ Scheme

It is of interest to compare the throughput performance of the previous schemes with the Type-II hybrid scheme proposed by Lin and Yu [37]. This scheme is more complex than the schemes presented so far. By more efficient use of the storage capacity in the receiver buffer, the scheme can provide a higher throughput.

In essence, the scheme works as follows. The retransmissions consist of alternate repetitions of an "information block" $\{u, P_0(u)\}$ and "parity block" $\{P_1(u), P_0(P_1(u))\}$ as for the alternate data parity scheme above. However this retransmission scheme is combined with a selective-repeat type of ARQ protocol proposed by Yu and Lin [34]. This protocol is similar to the SR + GBN protocol described in section 2.3 except that it makes better use of the receiver buffer store by a more complex set of retransmission protocols. Blocks are numbered with sequence numbers modulo- $3N$ where N is the number of blocks round-trip delay. Also the receiver buffer is capable of storing N blocks. The protocol provides that when errors occur on the channel, this receiver buffer is used to a maximum extent to store those blocks that are successfully received without error but which have higher sequence numbers than other blocks previously NAK'ed but not yet received correctly. The transmitter keeps a running check on the state of the receiver buffer and uses under/overflow information to determine the retransmission mode for NAK'ed blocks.

When a NAK is received for a block j (or no ACK is received at the expected time after one round-trip delay) the transmitter computes the forward index f_T of the block as follows. Let x_0 be the sequence number of the earliest NAK'ed or unACK'ed information block in the retransmission buffer (where all blocks are stored after transmission until ACK'ed). The

forward index f_T of an information block with sequence number x in the retransmission buffer (or in the input queue) with respect to the earliest NAK'ed or unACK'ed information word in the retransmission buffer is defined as

$$f_T = x - x_0 \pmod{3N}$$

Then for the NAK'ed or unACK'ed block j (known as the current time-out word) the transmitter proceeds as follows.

- If block j has a forward index f_T less than N , a selective-repeat type of retransmission of block j is initiated (either "parity block" or "information block" type of retransmission depending on whether the previous attempt was of "information" or "parity" type respectively). All the information words in the retransmission buffer with forward indices equal to or greater than N are moved back to the input queue for retransmission at a later time. Those words have been transmitted but when they were received the receiver buffer was full so buffer overflow occurred and the blocks were discarded at the receiver.
- If block j has a forward index f_T greater than or equal to N , then a type of Go-Back- N protocol is initiated. The first information word in the input queue is the next transmitted, this word having been moved back to the input queue from the retransmission buffer due to receiver buffer overflow.

Further details are given in Reference [37].

Although, the authors considered only the possibility of using a block code for forward-error-correction, the methods of this section can be used for a convolutional code application also. The methods outlined

above are generally applicable to either form of forward-error correction codes; the features of the particular coding scheme used must be taken into account when computing the conditional probabilities q_1 and q_2 defined above. The derivation of expressions for q_1 and q_2 for different coding arrangements will be discussed in subsequent sections.

For the Lin and Yu scheme, it has been shown that for a receiver buffer equal to one round trip delay $N = S+1$ blocks, the throughput is lower bounded as follows [37]:

$$\eta \geq \frac{E\{M\}_{\text{Inf}}}{E\{V\}_{\text{Inf}}} \quad (4.7)$$

$$\text{where } E\{M\}_{\text{Inf}} = \frac{\gamma[P_B + (1-P_B) - \beta]\beta^S}{(1-\beta)P_B}$$

$$\begin{aligned} \text{and } E\{V\}_{\text{Inf}} = E\{M\}_{\text{Inf}} &+ q_1\alpha^{S-1}[1-P_B+P_Bq_1(S+1)] \\ &+ q_2\beta^{S-1}[\alpha-q_1+\gamma(1-\alpha)(S+1)] \\ &+ (S-1)[2-\gamma\beta^S - q_1\alpha^S] \end{aligned}$$

$$\text{where } \alpha = 1 - P_B(1-q_1)$$

$$\beta = 1 - P_B(1-q_1)(1-q_2)$$

$$\text{and } \gamma = q_1 + q_2 - q_1q_2$$

The parameters q_1 and q_2 are defined as outlined earlier in this section. It can be noted that for this scheme, a rather complicated lower bound only is given compared to previous cases. This is because the particular construction of the Lin and Yu scheme does not permit the use of Equation (2.4) so that the lower bound has to be found by other ways. Details are given in Reference [37].

4.5 Rate 1/2 and 1/3 Parity Retransmission Scheme

The scheme represented in Figure 3.4 and described in section 3.3 utilizes parity retransmissions based on related 1/2 and 1/3 rate convolutional codes. In order to determine the throughput for this case, consider states 3 and 4 in Figure 3.4 being combined into one state. Also combine states 6 and 7. As a result the state diagram becomes of the same form as that shown in Figure 3.5. Hence the throughput expressions given by Equations (4.2) and (4.5) apply respectively for the SR + ST and SR + GBN protocols used with the 1/2|1/3 parity retransmission scheme.

The conditional probabilities q_1 and q_2 will of course differ from the values for the alternate data-parity scheme. These parameters reflect the effectiveness of the particular forward-error correction procedures used. Expressions for them will be discussed in a later section.

4.6 Alternate Data-Parity with ST + GBN Protocol

An ARQ retransmission protocol suggested by Weldon [44] has some similarity to the mixed-mode schemes described in section 2. When a NAK is received after the first transmission of a block the transmitter stops transmitting new blocks and re-sends the faulty block (stutter mode) a specified number (n_1) of times. Then the transmitter returns to transmitting other new blocks waiting in the transmit buffer. Should all n_1 transmission attempts of the faulty block be unsuccessful the transmitter again stops transmitting new blocks and retransmits the faulty block n_2 times in stutter mode. As before the transmitter returns to continuous transmission of new blocks. The process continues with the transmission of n_i copies of a faulty block if all n_{i-1} copies fail to be received without error. The intention is to choose the values of n_1, n_2, \dots to "match" the

channel, that is to provide maximum throughput performance for given error rate and delay.

It is necessary to provide receiver buffer with capacity sufficient to hold blocks received error-free with higher sequence numbers than the faulty block. Let the receiver buffer size be qS where S is the round-trip delay as defined in section 2 and q may be 1, 2, 3,.... The transmitter must revert to a Go-Back-N type of mode if receiver buffer overflow occurs. Accordingly we can for convenience refer to this protocol as an ST + GBN protocol. The following summary of the protocol explains the sequence of events:

Error free state. An ACK is received at the transmitter. A new block is sent.

Stutter mode. If a NAK is received for a given block, the transmitter next repeats the block n_1 times, then continues with new blocks. If all n_1 repeats are received in error (n_1 NAK's received) the transmitter repeats the block n_2 times, then continues with new blocks.

Go-Back-N mode. If all n_q repeats are received in error, buffer overflows will occur. The block is repeated n_q times and then the transmitter repeats other blocks transmitted following the previous stutter-mode transmission. The system stays in this mode until the faulty block is communicated.

Weldon [44] has derived an expression for throughput for this scheme. For the case where a receiver buffer of just one round-trip delay is provided ($q=1$) the throughput becomes

$$n = \frac{1 - P_B^{n_1}}{(1+n_1 P_B)(1-P_B^{n_1})+(n_1+S)P_B^{n_1+1}} \quad (4.8)$$

where P_B is the block error rate for a BSC. Note that if $n_1=1$ the scheme becomes the SR + GBN scheme of section 2.4 and the throughput is given by Equation (2.11) with $v=1$.

Consider now the incorporation of this ST + GBN protocol scheme into a Type-II hybrid ARQ system of the alternate data-parity type as in section 4.1. Following the same procedure, we can determine a lower bound on throughput by considering the inferior system (refer section 3.4) in which forward-error-correction is only attempted after transmission attempts 2, 3, 5, 6, 8, 9, ... of a given block. After transmission attempts 1, 4, 7, ... error detection only is carried out and all previously received data or parity transmissions of the faulty block are discarded.

For the ST + GBN protocol we have the following relationships between the generic random variables N (number of transmission attempts required for success) and M (number of equivalent transmission slots occupied including that occupied by any discarded blocks).

$$M = \begin{cases} 1 & , N = 1 \\ 1 + n_1 & , N = 2, 3, \dots, n_1+1 \\ 1 + 2n_1 + S & , N = n_1+2, n_1+3, \dots, 2n_1+1 \\ 1 + 3n_1 + 2S & , N = 2n_1+2, 2n_1+3, \dots, 3n_1+1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{cases} \quad (4.9)$$

We can now find the average number of time slots required per block using Equations (4.9) and (4.1). The expressions become somewhat unwieldy unless n_1 is assumed some fixed value. Examination of Equation (4.8) indicates the $n_1=3$ might be a reasonable choice. For this case we obtain:

$$\begin{aligned}
E\{M\} = & (1-P_B) + 4P_Bq_1 + 4P_B(1-q_1)q_2 + 4x(1-P_B) \\
& + (7+S) \times P_Bq_1 + (7+S) \times P_B(1-q_1)q_2 + (7+S) x^2(1-P_B) \\
& + (10+2S) x^2P_Bq_1 + (10+2S) x^2P_B(1-q_1)q_2 + (10+2S) x^3(1-P_B) \\
& + \dots
\end{aligned}$$

where $x = P_B(1-q_1)(1-q_2)$.

Taking the reciprocal after appropriate collection of terms we obtain the throughput as

$$\eta = \frac{1-P_B(1-q_1)(1-q_2)}{1+3P_B+P_B(1-q_1)(1-q_2)(SP_B-1)} \quad (4.10)$$

4.7 Summary

In Chapter 4 a unified procedure has been used to obtain expressions for the throughput of a number of hybrid ARQ schemes involving

- different retransmission protocols
- different error detection/correction schemes

The results were presented in terms of conditional probabilities q_1 and q_2 which were defined in Chapter 3. The values of q_1 and q_2 depend on the forward error correction/detection process being used, the properties of the codes and the decoding procedures. In the next chapters, expressions for these parameters will be given for convolutional codes and block codes.

In order to determine q_1 and q_2 for schemes using convolutional codes, it will be shown in Chapter 7 that it is useful to first compute the block error probability P_B' after decoding. In Chapter 5, methods for computing P_B' will be given. Upper bounds are derived for certain classes of convolutional decoders. This will then permit a comparative assessment of the different results of Chapter 4. The results of this chapter are not exhaustive in that expressions for throughput were determined only for

selected combinations of retransmission protocols and detection/correction schemes. However, it should be apparent that the analysis procedure can be readily applied to other related Type-II hybrid ARQ systems.

5. BLOCK ERROR RATE FOR CONVOLUTIONAL CODES

5.1 Introduction

We wish to determine the (average) probability P_B^i of incorrectly decoding the set of information symbols in a block. For a Type-II hybrid ARQ scheme with alternative transmissions of data, (2,1) parity, data, parity, etc... the probability P_B^i can be defined using the following:

Consider a "data information" transmission block (v) n bits long where

$$v = \{u, P_0(u)\}$$

The k bits u are followed by $(n-k)$ error detection parity bits $P_0(u)$ associated with an (n,k) error detection block code C_0 as illustrated in Figure 5.1(a).

Consider also the "parity" transmission block

$$v^* = \{P_1(u), P_0(P_1(u))\}$$

where the first k digits $P_1(u)$ are the parity bits generated by encoding u with a (2,1) convolutional code C_1 . The last $(n-k)$ digits $P_0(P_1(u))$ are the parity bits resulting from encoding $P_1(u)$ with block error detection code C_0 . This is illustrated in Figure 5.1(b).

Note that for $P_1(u)$ to provide a complete set of parity bits for u it is necessary to provide a "tail" of m zeros terminating u . That is the total number of information bits in the transmission u is $(k-m)$. Normally $m \ll k$ so this represents only a slight reduction in throughput efficiency.

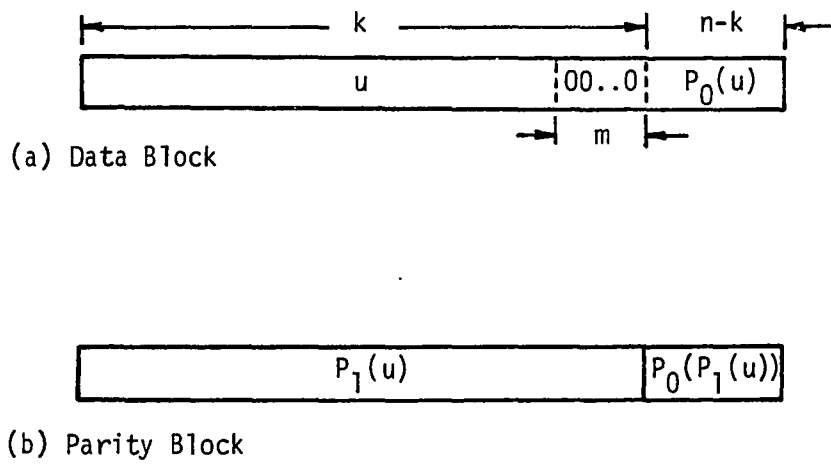


Figure 5.1

Alternate Data and Parity Transmission Blocks

We observe that other parity retransmission schemes are possible. For example, the first transmission u could consist of k information bits with no zero tail as illustrated in Figure 5.2. The parity retransmission could consist of n symbols all of which are parity digits generated by a convolutional code C_1 designed for error correction and detection. This can be achieved providing that for the code C_1 , $m \leq n-k$. However, in the following we consider the first parity retransmission scheme.

Let the k data bits \hat{u} of the received "data" block v be represented

$$\hat{u} = (u_0, u_1, \dots, u_{k-1}) .$$

Let the k parity bits $\hat{P}_1(u)$ of the next received "parity" block \hat{v}^* be represented

$$\hat{P}_1(u) = (p_0, p_1, \dots, p_{k-1}) .$$

Both \hat{u} and $\hat{P}_1(u)$ may contain errors. The decoder attempts to correct the errors and recover u . This may be preceded by a multiplexing of \hat{u} and $\hat{P}_1(u)$ to form a sequence

$$(u_0, p_0, u_1, p_1, \dots, u_{k-1}, p_{k-1})$$

which is $N_B = 2k$ bits in length.

We define the block error probability (P_B^i) as:

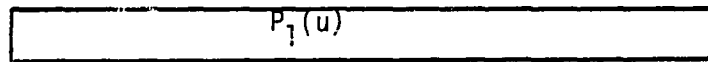
$$P_B^i = \text{Pr}\{\text{one or more errors remain after decoding the } N_B \text{ bits}\} .$$

Obviously P_B^i will depend on the

- channel error pattern before decoding
- the convolutional code used
- the decoding procedure



(a) Data Block



(b) Parity Block

Figure 5.2

Another Possible Alternate Data/Parity Scheme

We assume a randomly distributed error pattern with bit-error-rate p . Further we assume that for the 1/2 rate convolutional code, the constraint length (n_A bits) or memory order (m) and the minimum free distance (d_{free}) are known.

5.2 Decoder Characteristics

The choice of the error-correction decoder to be used in the hybrid ARQ system would in practice depend on a compromise between performance and complexity. Possible classes of decoders include the following

- Viterbi algorithm decoders
- Sequential decoders
- Sliding block feedback decoders including Threshold Decoders and Table look-up.

Viterbi decoders and Sequential decoders can be classed in the high performance, more complex end of the scale. The Viterbi algorithm is a maximum likelihood decoding scheme. Its major drawback is that while error probability decreases exponentially with constraint length, the corresponding decoder complexity also grows exponentially.

Sequential decoders offer asymptotically the same error probability as Viterbi decoders. The decoder is also reasonably complex but suffers an additional disadvantage for ARQ usage in that it is necessary to incorporate a considerable decoder buffer (typically of the order of 10^4 - 10^5 bits [38]) which results in undesirable delay. This is necessary because the number of operations performed for each input bit to the Sequential decoder is a random variable and a buffer must be provided to avoid loss of data.

The Feedback decoder class of decoding algorithms is more attractive for hybrid ARQ purposes particularly because of simplicity. While some degradation of performance is inevitable the availability of a retransmission procedure whenever decoding fails still ensures high reliability. The following analysis therefore determines the block error rate performance of two types of (syndrome) feedback decoders: threshold and table look-up (ROM) decoders.

To obtain reasonable performance but with minimum decoder complexity, attention is given to 1/2 rate convolutional codes with moderate error correcting ability. For this purpose, codes with maximal d_{free} for reasonably small values of constraint length are of particular interest. The following tables list some 1/2 rate codes considered from Lin and Costello [2]. In Table 5.1 optimal codes are listed and their minimum free distance (d_{free}), constraint length (n_A) are shown. In Table 5.2 orthogonalizable codes suitable for threshold decoding are shown in terms of their t-error correcting ability and constraint length.

Table 5.1
Codes with Maximal d_{free}

d_{free}	Systematic Codes	Nonsystematic Codes	Complementary Codes
	n_A	n_A	n_A
3	4	4	-
5	10	6	6
7	16	10	10
9	-	-	14
10	-	14	-

Table 5.2
Orthogonalizable Codes (1/2 rate)

t	n_A
1	4
2	12
3	24
4	44

As pointed out in Viterbi and Omura [38] the class of Feedback decoders operates essentially as "sliding block decoders". The decoder can correct any t -error pattern in a "sliding window" of length n_A bits providing d_{free} is greater than or equal to $2t+1$.

In the following sections, methods for computing the bound-distance block-error probability performance after decoding are given. The results provide upper bounds on P_B^i for all the above decoders being tightest for the threshold and table look-up respectively. Two methods are presented, namely

- Trellis algorithm approach
- Combinatorial pattern enumeration method.

5.3 Trellis Algorithm for P_B^i Evaluation for Convolutional Codes

Consider the first received data block u of k bits

$$\text{where } \hat{u} = (u_0, u_1, \dots, u_{k-1})$$

and also the k received associated (2,1) convolutional code parity bits

$$\hat{P}_1(u)$$

$$\text{where } \hat{P}_1(u) = (p_0, p_1, \dots, p_{k-1}).$$

After interleaving we have the reconstituted decoder input vector

$$(u_0, p_0, u_1, p_1, \dots, u_{k-1}, p_{k-1})$$

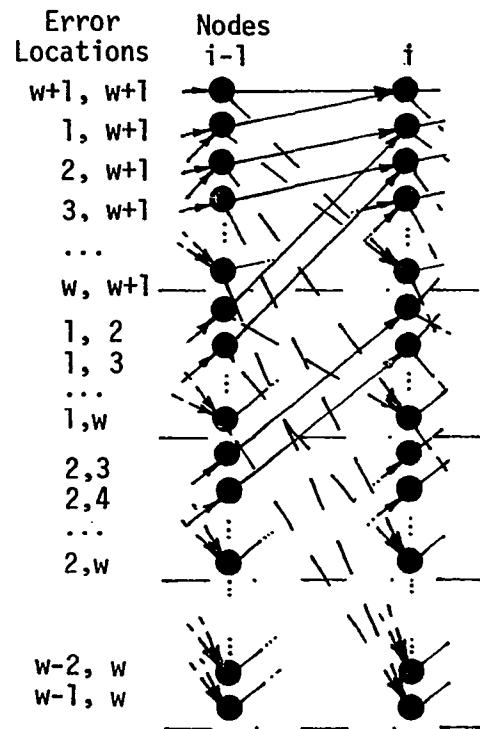
of length $2k=N_B$ bits. Given a random BSC with bit error rate p , a computer program can in principle be written to enumerate all error patterns and their associated probabilities such that each pattern is correctable by the decoder, that is, all patterns which contain no more than t errors in a constraint length "window" of n_A bits anywhere in the input vector. However, for typical N_B values of the order of $N_B=2000$ bits, the checking and enumeration for all 2^{N_B} patterns would be hopelessly impractical.

However a new algorithm based on a code trellis diagram has been developed to reduce the number of computations to within reasonable bounds, at least for values of t and n_A not too large. The algorithm may be seen to bear a resemblance to the Viterbi decoding procedure. Consider the first n_A bits in the N_B -tuple at the decoder input (i.e. the constraint length "window" located over the first n_A bits in the decoder input). For ease of typing, let $\omega=n_A$.

We construct a trellis diagram such that the first set of nodes in the trellis (placed in the first column) represent each of the possible error patterns of bits for which the total number of errors is less than t . Figure 5.3 illustrates the method for a $t=2$ error correcting code. Note that although there are a total of 2^ω possible error patterns, the number (C') of correctable error patterns (number of first nodes) is only

$$C' = \sum_{i=0}^t \binom{N_B}{i}$$

where i is the Hamming weight of the error pattern.



$$P_i(w+1, w+1) = (1-p) \{P_{i-1}(w+1, w+1) + P_{i-1}(1, w+1)\}$$

$$P_i(j, w+1) = (1-p) \{P_{i-1}(j+1, w+1) + P_{i-1}(1, j+1)\}$$

$$j=1, 2, \dots, w-1.$$

⋮

Figure 5.3

Trellis for Sliding Block Decoding ($t=2$) Performance Calculation

For each of the first nodes (correctable error patterns), we compute and store the probability of the associated error pattern e occurring using

$$P_1(e) = p^i(1-p)^{\omega-i}, \quad \text{where } p = \text{BER} .$$

Now consider the constraint length "window" moved one bit into the block (i.e. the first bit is excluded but the $(\omega+1)^{\text{th}}$ bit is included). Again we have C' correctable error patterns represented by the second column of nodes in the trellis illustrated in Figure 5.3. For each of the C' nodes, it is a simple matter to compute the "running probability" of reaching the node commencing at the start of the trellis. For example, in Figure 5.3 the second node "null error" pattern $(000\dots 0)$ n_A bits long has "running probability" of

$$P_2(000\dots 0) = (1-p)\{P_1(000\dots 0) + P_1(100\dots 0)\}$$

where $P_1(000\dots 0)$ and $P_1(100\dots 0)$ are the probabilities computed at the first node set for the two error patterns $(000\dots 0)$ and $(100\dots 0)$ respectively. Note that as the "window" is moved from the first to the second position, both these patterns have a 0 added and their first error bits removed to obtain $(000\dots 0)$ at the second stage. For each of the C' correctable pattern nodes at the second stage the associated "running probabilities" can be computed similarly using no more than two probability values from the previous set. The required calculations are listed below. The notation used to number each node is based on a t dimensional array of the form $P(x_1, x_2, \dots, x_t)$ where x_1, x_2, \dots denotes the positions of the 1st, 2nd ... errors in the window pattern respectively and $x_1 < x_2 < \dots$

For example for $t=2$

$$P(1,2) \equiv (1100\dots 0)$$

denotes an error pattern with errors in the first and second positions.

Also

$$P(1,3) \equiv (10100\dots 0)$$

a pattern with errors in the 1st and 3rd positions, and so on. Conventional computer programming array notation (as used in Fortran for example) prohibits array indexes of zero. Therefore, in order to describe error patterns with $i < t$ errors, the artificial array integer $\omega+1$ is used so that the node designation includes $(t-i)$ dimensioning integers $(\omega+1)$. For example, for $t=2$ the null error pattern $(000\dots 0)$ is represented as having probability $P(\omega+1, \omega+1)$. Likewise the error pattern $(0100\dots 0)$ is represented by $P(2, \omega+1)$.

In summary, for $t=2$ error-correcting codes the iteration procedure is as follows:

Compute

$$P_i(\omega+1, \omega+1) = (1-p)\{P_{i-1}(\omega+1, \omega+1) + P_{i-1}(1, \omega+1)\}$$

$$P_i(j, \omega+1) = (1-p)\{P_{i-1}(j+1, \omega+1) + P_{i-1}(1, j+1)\} \text{ for } j = 1, 2, \dots, \omega-1$$

$$P_i(j, k) = (1-p)P_{i-1}(j+1, k+1) \text{ for } j = 1, 2, \dots, \omega-2$$

$$k = 2, 3, \dots, \omega-1$$

and $j < k$.

$$P_i(j, \omega) = p\{P_{i-1}(j+1, \omega+1) + P_{i-1}(1, j+1)\} \text{ for } j = 1, 2, \dots, \omega-1$$

$$P_i(\omega, \omega+1) = p\{P_{i-1}(\omega+1, \omega+1) + P_{i-1}(1, \omega+1)\} \quad (5.1)$$

with initial conditions

$$P_{n_A}(\omega+1, \omega+1) = (1-p)^{N_B}$$

$$P_{n_A}(j, \omega+1) = p(1-p)^{N_B-1} \quad \text{for } j = 1, 2, \dots, \omega$$

$$P_{n_A}(j, k) = p^2(1-p)^{N_B-2} \quad \text{for } j = 1, 2, \dots, \omega-1 \\ k = 2, 3, \dots, \omega \\ j < k$$

After the algorithm computes the running probabilities $P_2(x_1, x_2, \dots)$ for all C' correctable error patterns in the second column of nodes, all the C' stored probabilities $P_1(x_1, x_2, \dots)$ for the first column of nodes can be wiped from memory as they are no longer required. If required, the second stage probabilities can be added together to provide the probabilities $P_C(\omega+1)$ where

$$P_C(\omega+1) = \sum_{C'} P_2(x_1, x_2, \dots)$$

which is the probability that a received block of length $(\omega+1)$ bits will be decoded correctly. The algorithm now proceeds to complete the above process by computing probability values for the third column of nodes (third window position) and so on up to the $(N_B - \omega+1)$ th column of nodes representing the last window position in the block. Then the sum of these values produces the block error probability P_B' that a block of N_B bits will not be decoded correctly using

$$P_B' = 1 - \sum_{C'} P_{N_B-\omega+1}(x_1, x_2, \dots) \quad (5.2)$$

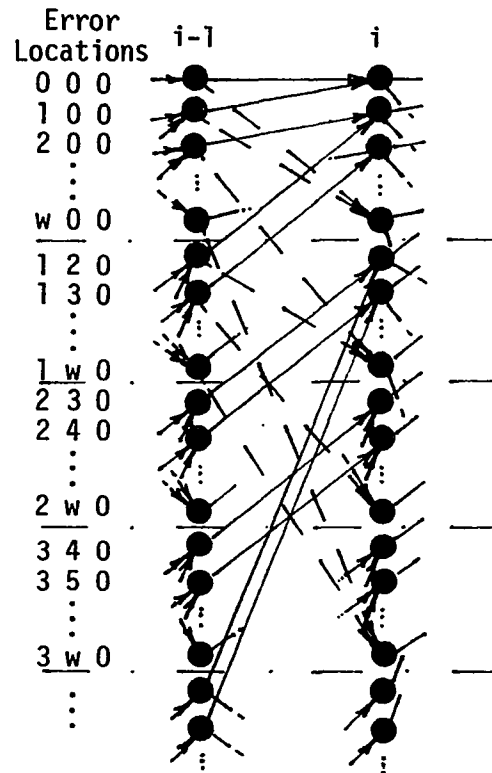
This provides a relatively simple method of computing exactly the bound distance decoding block error probability P'_B for any convolutional code and associated decoder capable of correcting up to t errors in any constraint length of n_A bits.

Figures 5.4 and 5.5 illustrate the trellis and associated calculations for $t=3$ and $t=4$ correcting codes. A typical Fortran program used to compute block-error probability (P'_B) using the algorithm is listed in Appendix B. In general, the computation requires two memory arrays of maximum size $(n_A)^t$ and approximately $C'(N_B-n_A+1)$ computations.

In section 5.6 some computed results are provided for typical convolutional codes. The above method appears to provide a general decoding bound with wider application than the ARQ system analysis under study. It also should provide an interesting means for making direct comparisons between the decoding performance of block codes and convolutional codes over the random BSC.

5.4 Combinatorial Approach to P'_B Calculation

This section describes an alternative approach to computing a bound-distance decoder block error probability for convolutional codes. The results provide a check on the trellis algorithm method of section 5.4. Also they permit the estimation of block error probability for parameter values of the code and its decoder where the trellis algorithm may prove too time consuming. Methods for obtaining closed form expressions for enumerating the number $N(j)$ of correctable error patterns of j errors in a block of N_B bits received at the input to an (n,k) convolutional code decoder are presented. For each value of j , a difference equation



$$P_i(0,0,0) = (1-p) \{P_{i-1}(0,0,0) + P_{i-1}(1,0,0)\}$$

$$P_i(j,0,0) = (1-p) \{P_{i-1}(j+1,0,0) + P_{i-1}(1,j+1,0)\}$$

$$j=1,2,\dots,w-1.$$

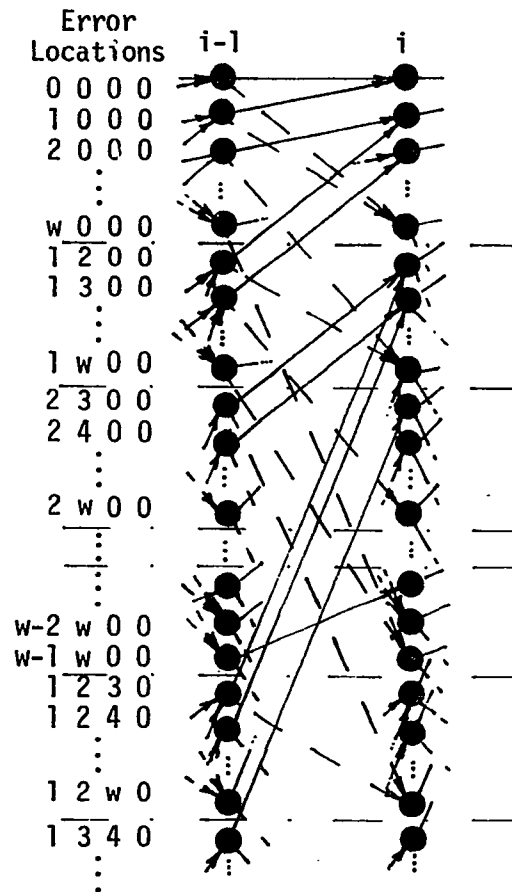
$$P_i(w,0,0) = p \{P_{i-1}(0,0,0) + P_{i-1}(1,0,0)\}$$

⋮

For computer program $w+1 \equiv 0$

Figure 5.4

Trellis Diagram for $t=3$ Decoding



$$P_i(0,0,0,0) = (1-p) \{P_{i-1}(0,0,0,0) + P_{i-1}(1,0,0,0)\}$$

$$P_i(j,0,0,0) = (1-p) \{P_{i-1}(j+1,0,0,0) + P_{i-1}(1,j+1,0,0)\}$$

$$j=1,2,\dots,w-1$$

⋮

Note: $0 \equiv w+1$ for computer program

Figure 5.5

Trellis Diagram for t=4 Decoding

approach is used to find closed form expressions for $N(j)$ for various values of N_B , n_A , t and j .

In order to obtain values for block error probability P'_B for a convolutional code used in Type-II hybrid ARQ it is necessary to assume the decoding type is known. As discussed previously, if we assume a Majority logic decoding circuit will be used, then for a J -orthogonalizable code, any pattern of $t=\lfloor J/2 \rfloor$ or fewer errors in a span of $n_A=n(m+1)$ positions can be corrected.

If we consider the $2k$ bit sequence made up of the information u and C_1 parity $P_1(u)$ then for a BSC with transmission error probability p

$$P'_B = 1 - \sum_{j=0}^{2k} N(j) p^j (1-p)^{2k-j} \quad (5.3)$$

where $N(j)$ is the number of patterns of j bits in the $2k$ bit sequence which can be corrected by C_1 . In general, $N(j)$ will be a function of the following

- the block length ($2k$)
- the code constraint length $n_A=n(m+1)$
- the code error correction ability t .

Note that a majority logic decoder with feedback decoding may correct some patterns with greater than t errors in a constraint length span so Equation (5.3) provides a lower bound on P'_B .

The approach used to evaluate $N(j)$ is based on combinatorial enumeration techniques.

$t = 1$ Error Correcting Codes

Theorem 5.1. For single error $t=1$ correcting codes, the number $N(j)$ of correctable patterns in an N_B digit block is

$$\begin{aligned}
 N(j) &= \binom{N_B - (j-1)(n_A-1)}{j} \\
 &= \frac{[N_B - (j-1)(n_A-1)]!}{j! [N_B - (j-1)n_A - 1]!}
 \end{aligned} \tag{5.4}$$

$$\text{and } P'_B = 1 - \sum_{j=0}^{\left\lceil \frac{N_B}{n_A} \right\rceil} N(j) p^j (1-p)^{N_B-j} \tag{5.5}$$

Proof. Consider j errors arranged in correctable patterns into the N_B digit block (represented by crosses in Figure 5.6(a)).

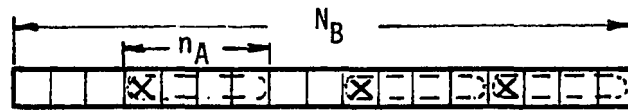
After each of the first $j-1$ errors there must be at least n_A-1 error free digits (positions in which error patterns are not allowed). There are a total of $(j-1)(n_A-1)$ such positions. Therefore $N(j)$ is the number of ways in which j errors can be arranged to occur in $N_B - (j-1)(n_A-1)$ bits.

The maximum number of errors that can conform to a correctable pattern in the block is determined by considering how many constraint length spans of length n_A bits (each commencing with an error) can be accommodated in the block of N_B bits. As shown in the Figure 5.6(b) it is necessary to count those subblocks which span N_B bits commencing at the first position.

The last block to be counted is the one that begins within the N_B bit block and ends with the N_B^{th} bit or beyond it. That is

$$j_{\max} = \left\lceil \frac{N_B}{n_A} \right\rceil$$

the least integer greater than or equal to N_B/n_A . As a result P'_B for $t=1$ is given by Equations (5.4) and (5.5). Note that the performance of the



(b) Typical Correctable Error Pattern

X represents an error

--- represents decoding constraint length (n_A bits)



(a) Illustration of Procedure for Counting
the Maximum Number of Correctable Errors

Figure 5.6

Correctable Error Patterns for $t=1$ Decoding

code is independent of the actual decoding operation used providing the decoder achieves the t -error correcting ability of the code. In that sense it is a bound distance decoding performance result.

$t=2$ Error-Correcting Codes

Consider the transmission of a binary data block of length N_B bits consisting of a segment of the output from a $(2,1)$ convolutional encoder using a code and associated decoder capable of correcting 2 errors in any constraint length of n_A bits.

Consider that the block has been transmitted through a random binary symmetric channel with bit error rate p . Then the block error probability P_B' after decoding can be determined by

$$P_B' = 1 - \sum_{j=0}^{\infty} N(j)p^j(1-p)^{N_B-j} \quad (5.6)$$

$$j = 0, 1, \dots$$

where $N(j)$ = the no. of patterns of j errors in the N_B -tuple which can be corrected by the decoder.

Previously an expression for $N(j)$ was found for a single error ($t=1$) correcting code. The enumeration of $N(j)$ for $t=2$ is a much more difficult task. In general, $N(j)$ can be represented by a polynomial in N_B of degree j .

The following enumerations of $N(j)$ have been proven using difference equations.

Theorem 5.2. For double error ($t=2$) correcting codes the number $N(j)$ of correctable error patterns of j errors in N_B bits is given as follows:

$$j=0: N(j) = 1 \quad (5.7)$$

$$j=1: N(j) = N_B \quad (5.8)$$

$$j=2: N(j) = \binom{N_B}{2} \quad (5.9)$$

$$j=3: N(j) = \binom{N_B - n_A + 2}{3} + \binom{n_A - 2}{1} \binom{N_B - n_A + 1}{2} \quad (5.10)$$

$$j=4: N(j) = \binom{N_B - n_A + 2}{4} + \binom{N_B - n_A + 1}{4} - \binom{N_B - 2n_A + 3}{4} \quad (5.11)$$

$$j=5: N(j) = \binom{N_B - 2n_A + 2}{5} + 2 \binom{n_A - 1}{1} \binom{N_B - 2n_A + 2}{4} + \binom{n_A}{2} \binom{N_B - 2n_A + 2}{3} \quad (5.12)$$

$$j=6: N(j) = 5 \binom{N_B - 2n_A + 3}{6} - 4 \binom{N_B - 3n_A + 4}{6} - \binom{2n_A - 1}{1} \binom{N_B - 3n_A + 4}{5} + \binom{N_B - 2n_A + 2}{4} \quad (5.13)$$

$$j=7: N(j) = \sum_{i=1}^{n_A - 1} \binom{i+2}{3} \binom{n_A - 2}{i-1} \binom{N_B - 4n_A + 8}{8-i} + \sum_{i=1}^{n_A - 2} \binom{i}{1} \binom{n_A - 1}{i+1} \binom{N_B - 4n_A + 6}{6-i} \quad (5.14)$$

The above expressions for $N(j)$ for $j = 0, 1$ and 2 are obvious since all patterns are correctable for those values of j . The derivations of the other expressions are discussed in Appendix C. A general expression for all j values has not been found. However, the expressions found so far provide a useful upper bound on P_B^j for typical values of block length (N_B) and channel bit-error-rate (p). Also they have been found in practice to provide an exact check on P_B^j values computed by the trellis algorithm of section 5.4 for parameter values such that $N(j)=0$ for $j>7$. This is true for relatively small block lengths. Further details of computations are given in section 5.6. The expressions given in Theorem 5.2 can be checked using Newton's backward differences formula as follows. Given a function $f(x)$ the collocation polynomial can be expressed in terms of the argument x at values x_0, x_1, \dots as

$$f(x) = f_0 + \left(\frac{x-x_0}{x_1-x_0} \right) \Delta f_0 + \frac{(x-x_0)(x-x_1)}{(x_1-x_0)^2!} \Delta^2 f_0 + \dots \quad (5.15)$$

where $f_i = f(x_i)$

$$f_0 = f_1 - f_0$$

and $\Delta^2 f_0 = \Delta f_1 - \Delta f_0$ with $\Delta f_1 = f_2 - f_1$.

For a polynomial $f(x)$ of degree j there are $j+1$ non-zero terms in the equation requiring specification of $f(x)$ at $j+1$ points x_0, x_1, \dots, x_j .

Once these values are specified, the polynomial can be uniquely found from (5.15). This can be applied to the expressions in Theorem 5.2 after using a counting procedure to find the first $j+1$ values for $N(j)$ as shown in Table 5.3; in all cases the results are consistent.

$N(j)$ values for $j \geq 8$, $t=2$

Exact expressions have been found as outlined above for the number $N(j)$ of correctable error patterns of j errors in N_B bits given that a $t=2$ sliding block decoder is used. For values of $j \geq 8$, the following lower bound on $N(j)$ is useful

$$N(j) \geq \binom{N_B - (j-1)(n_A - 1)}{j} \quad (5.16)$$

This follows from Theorem 5.1.

$N(j)$ values for $t > 2$ codes

For t error correcting codes of constraint length n_A , the number $N(j)$ of correctable error patterns of j errors in a block of length N_B is lower bounded as follows:

$$\begin{aligned} N(j) &= \binom{N_B}{j} \quad \text{for } j = 0, 1, \dots, t \\ N(j) &> N(j)|_{t=2} \quad j = t+1, t+2, \dots, 7 \\ N(j) &> \binom{N_B - (j-1)(n_A - 1)}{j} \quad j = 8, 9, \dots \end{aligned} \quad (5.17)$$

where $N(j)|_{t=2}$ are the values for $N(j)$ given by Theorem 5.2 for $t=2$ codes and the bound for $j \geq 8$ follows from Equation (5.4).

Table 5.3

Number of Correctable Error Patterns $N(j, N_B)$ of j Errors in N_B Bits as a Function of Block Length N_B and Constraint Length n_A ($t=2$)

$j=3$

n_A	N_B										
	4	5	6	7	8	9	10	11	12	...	
3	2	7	16	30	50	77	112		...		
4		3	10	22	40	65	98	140			
5			4	13	28	50	80	119			
6				5	16	34	60	95	140		
⋮											

$N(3, N_B)$ values

$j=4$

n_A	N_B													
	4	5	6	7	8	9	10	11	12	13	14	...		
3	0	1	6	19	45	90	266			...				
4			1	6	20	49	100	181	301					
5				1	6	20	50	104	191	321				
6					1	6	20	50	105	195	331			
⋮														

$N(4, N_B)$ values

$j=5$

n_A	N_B																	
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...		
3			0	3	16	51	126	266	504									
4					0	6	30	91	216	441	812							
5							0	10	48	141	326	65	1176					
6									0	15	70	201	456	896	1596			
⋮																		

$N(5, N_B)$ values

Table 5.3 (continued)

j=6

n_A	4	5	6	7	8	9	10	11	12	N_B	13	14	15	16	17	18	19	20	...
3				0	1	10	45	141	357	784	1554				...				
4						0	1	10	50	168	444	100	2016						
5								0	1	10	50	176	481	1118	2303				
6										0	1	10	50	175	490	1165	2450		
⋮															⋮				

$N(6, N_B)$ values

j=7

n_A	4	5	6	7	8	9	10	11	12	N_B	13	14	15	16	17	18	19	20	21	21...
3						0	4	30	126	393	1016	2304	4740							
4									0	10	70	280	840	210	4628	9276				
5											0	20	133	512	1486	3610	7750	15184		
⋮																				

$N(7, N_B)$ values

5.5 Computed Block Error Probability (P_B') values

Fortran programs based on the trellis algorithm of section 5.4 have been used to compute exact values for the block error probability (P_B') for bound-distance (sliding block or syndrome-feedback) decoding. Table 5.4 lists P_B' values versus bit-error-rate before decoding for selected 1/2 rate codes referred to in section 5.3. Values are given in the table for "block" lengths N_B of 1024 and 2000 bits respectively. As pointed out in section 2.1 it is assumed that the last m information bits are zero to clear the encoder for the last m parity bits in the "block" (where m is the memory order of the code). Figure 5.7 shows plots of P_B' versus BER for selected codes for "block" sizes $N_B = 2000$ bits.

It should be noted that Table 5.4 can be used to find P_B' values for other values of N_B with good accuracy for $N_B > 100$ bits. This can be found using the $P(N_i/N_{i-1})$ values given in the Table. Recall that the iteration procedure computes P_B' for a given block length, say N_{i-1} bits. Then the constraint length decoding window is moved forward one bit and a new value of P_B' is computed for block length of N_i bits and so on. Let

$$P_\infty(N_i - N_{i-1}) = \lim_{N_B \rightarrow \infty} \frac{P_B'(N_i) - P_B'(N_{i-1})}{1 - P_B'(N_{i-1})}$$

where $P_B'(N_i)$ = the P_B' value for block length of N_i

and $P_B'(N_{i-1})$ = the P_B' value for block length of N_{i-1} .

Now $P(N_i/N_{i-1}) = P_r$ {the block is not correctly decodable at length N_i given that it was decodable at length N_{i-1} } i.e.

$$P_B'(N_i) = P_B'(N_{i-1}) + \{1 - P_B'(N_{i-1})\} p(N_i/N_{i-1}) \quad (5.18)$$

It has been found from the computations performed so far that, for the codes examined, the ratio

Table 5.4

Block Error Rate Values versus BER for Selected Convolutional Codes

t=1 $n_A=4$ BER	Block Error Rate P_B			
	$N_B=100$	$N_B=1024$	$N_B=2000$	$P(N_i/N_{i-1})$
.1	.8911	1.0000	1.0000	.022056
.03	.2136	.91618	.9930	.0024006
.01	$.2819 \times 10^{-1}$.25569	.44616	$.2886 \times 10^{-3}$
.003	$.2639 \times 10^{-2}$.02693	$.5315 \times 10^{-1}$	$.26681 \times 10^{-4}$
.001	$.2958 \times 10^{-3}$.00305	$.6098 \times 10^{-2}$	$.2988 \times 10^{-5}$
.0003	$.2670 \times 10^{-4}$	$.2758 \times 10^{-3}$	$.5519 \times 10^{-3}$	$.26968 \times 10^{-6}$
.0001	$.2969 \times 10^{-5}$	$.3068 \times 10^{-4}$	$.6138 \times 10^{-4}$	$.29988 \times 10^{-7}$
.00001	$.2970 \times 10^{-7}$	$.3069 \times 10^{-6}$	$.6141 \times 10^{-6}$	$.29999 \times 10^{-9}$
.000001	$.2970 \times 10^{-9}$	$.3069 \times 10^{-8}$	$.6141 \times 10^{-8}$	$.3 \times 10^{-11}$

t=2 $n_A=10$ BER	Block Error Rate P_B			
	$N_B=100$	$N_B=1024$	$N_B=2000$	$P(N_i/N_{i-1})$
.1	.7462	1.0000		
.03	$.6497 \times 10^{-1}$.5167		
.01	$.3032 \times 10^{-2}$	$.3248 \times 10^{-1}$	$.6268 \times 10^{-1}$	$.3245 \times 10^{-4}$
.003	$.8801 \times 10^{-4}$	$.9582 \times 10^{-3}$	$.1877 \times 10^{-2}$	$.9427 \times 10^{-6}$
.001	$.3326 \times 10^{-5}$		$.7106 \times 10^{-4}$	$.3563 \times 10^{-7}$
.0003	$.9045 \times 10^{-7}$		$.1932 \times 10^{-5}$	$.9690 \times 10^{-9}$
.0001	$.3357 \times 10^{-8}$		$.7172 \times 10^{-7}$	$.3596 \times 10^{-10}$
.00001	$.3688 \times 10^{-11}$			
.000001				

Table 5.4 (continued)

t=2 $n_A=6$ BER	P_B Values	
	$N_B = 2000$	$P(N_i/N_{i-1})$
.1		
.03		
.01	$.1881 \times 10^{-1}$	$.9510 \times 10^{-5}$
.003	$.5310 \times 10^{-3}$	$.2660 \times 10^{-6}$
.001	$.1987 \times 10^{-4}$	$.9950 \times 10^{-8}$
.0003	$.5384 \times 10^{-6}$	$.2696 \times 10^{-9}$
.0001	$.1996 \times 10^{-7}$	$.9995 \times 10^{-1}$

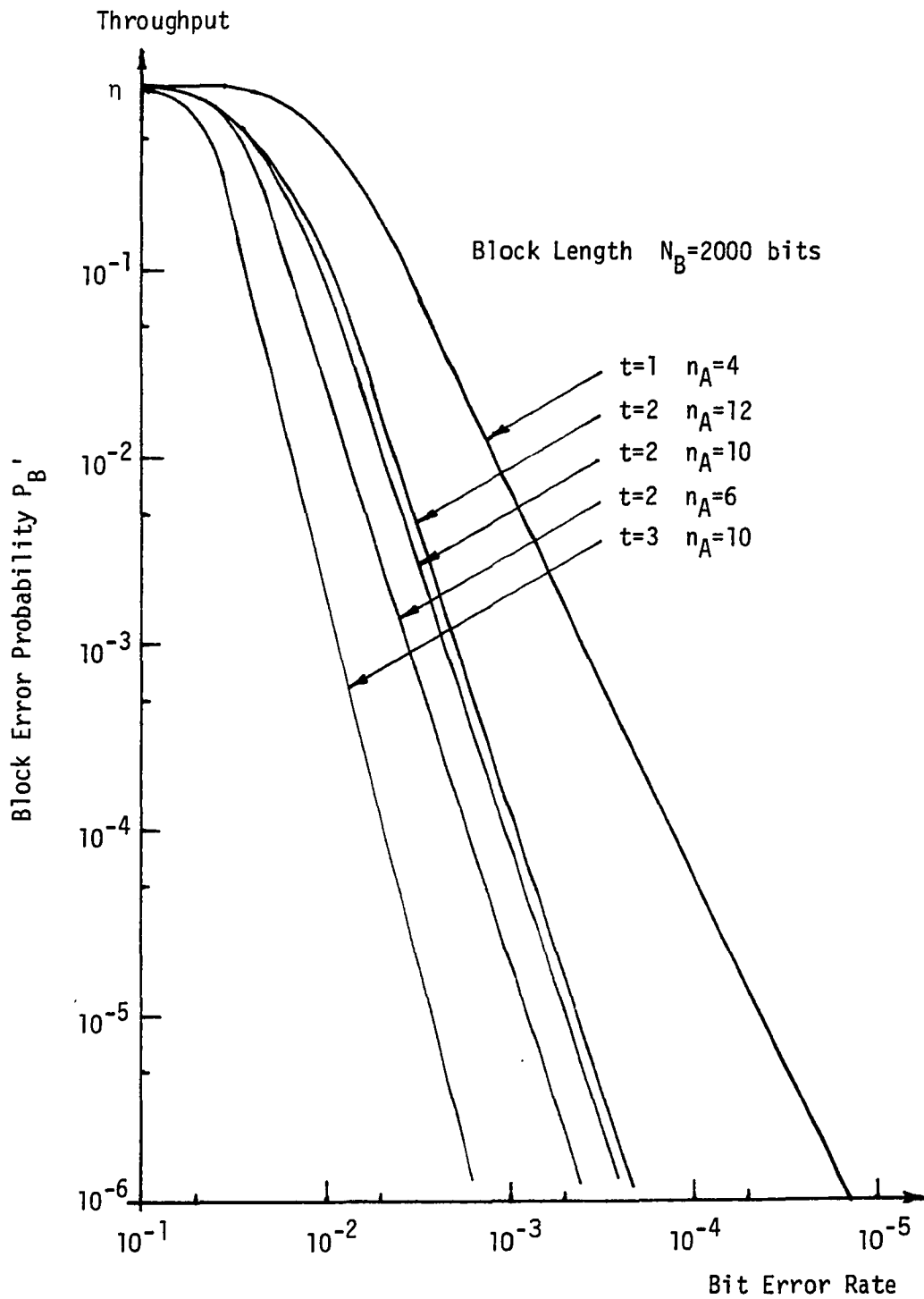


Figure 5.7

Block Error Rate After Sliding Block
Decoding for Various Convolutional Codes

$$P(N_i/N_{i-1}) = \frac{P_B(N_i) - P_B(N_{i-1})}{1 - P_B(N_{i-1})}$$

becomes constant to within approximately 12 significant figures for N_B values greater than 100 bits. Therefore it is possible to use the $P(N_i/N_{i-1})$ values from Table 5.4 to find P_B for any block length greater than 100 bits using a simple iteration procedure. This can be used to considerably reduce the computation time required to find P_B for large block lengths.

The "block" error rate values for selected convolutional codes shown in Figure 5.7 indicate the performance expected for a sliding block decoding scheme such as Threshold Decoding using Orthogonalizable codes. The best $t=2$ orthogonalizable code has $n_A=12$ (refer section 5.3). Such a decoding procedure is simple to implement. However the performance is inferior to either the Viterbi decoder or a Sequential Decoder so the results in Figure 5.7 provide upper bounds on the "block" error probability for these more efficient decoders.

It is believed that "block" error rates after decoding have not previously been available for any of the above classes of decoders.

5.6 Comparisons between Convolutional and Block Code Performance

The calculations of block-error-rate (P_B) after decoding for selected convolutional codes as described in the previous section seem to offer a new means of comparing convolutional code and block code performance for rate 1/2 and 1/3 codes with comparable block sizes on the random BSC.

For (n,k) block codes on the random BSC, an upper bound on block error probability P_B is given by (Viterbi and Omura [38])

$$P_B \leq \begin{cases} \sum_{j=(d_{\min}+1)/2}^n \binom{n}{j} p^j (1-p)^{n-j}, & d_{\min} \text{ odd} \\ \sum_{j=d_{\min}/2}^n \binom{n}{j} p^j (1-p)^{n-j}, & d_{\min} \text{ even} \end{cases} \quad (5.19)$$

For "perfect" codes, such as the Hamming codes or the Golay code, the above expressions are exact.

A comparison of block code versus convolutional code performance is indicated in Figure 5.8 for $N_B=2000$ bits. Values of $t=5$ and 10 are shown for block codes using Equation (5.25) with

$$d_{\min} = 2t + 1$$

The results show that for bit-error rates greater than 10^{-3} before decoding, the orthogonalizable convolutional code with threshold decoding could be expected to provide superior performance. It will be shown in Section 7 that this might provide superior throughput performance in a Type-II hybrid ARQ system over the results obtained using block codes for forward error correction, assuming comparable decoder complexity.

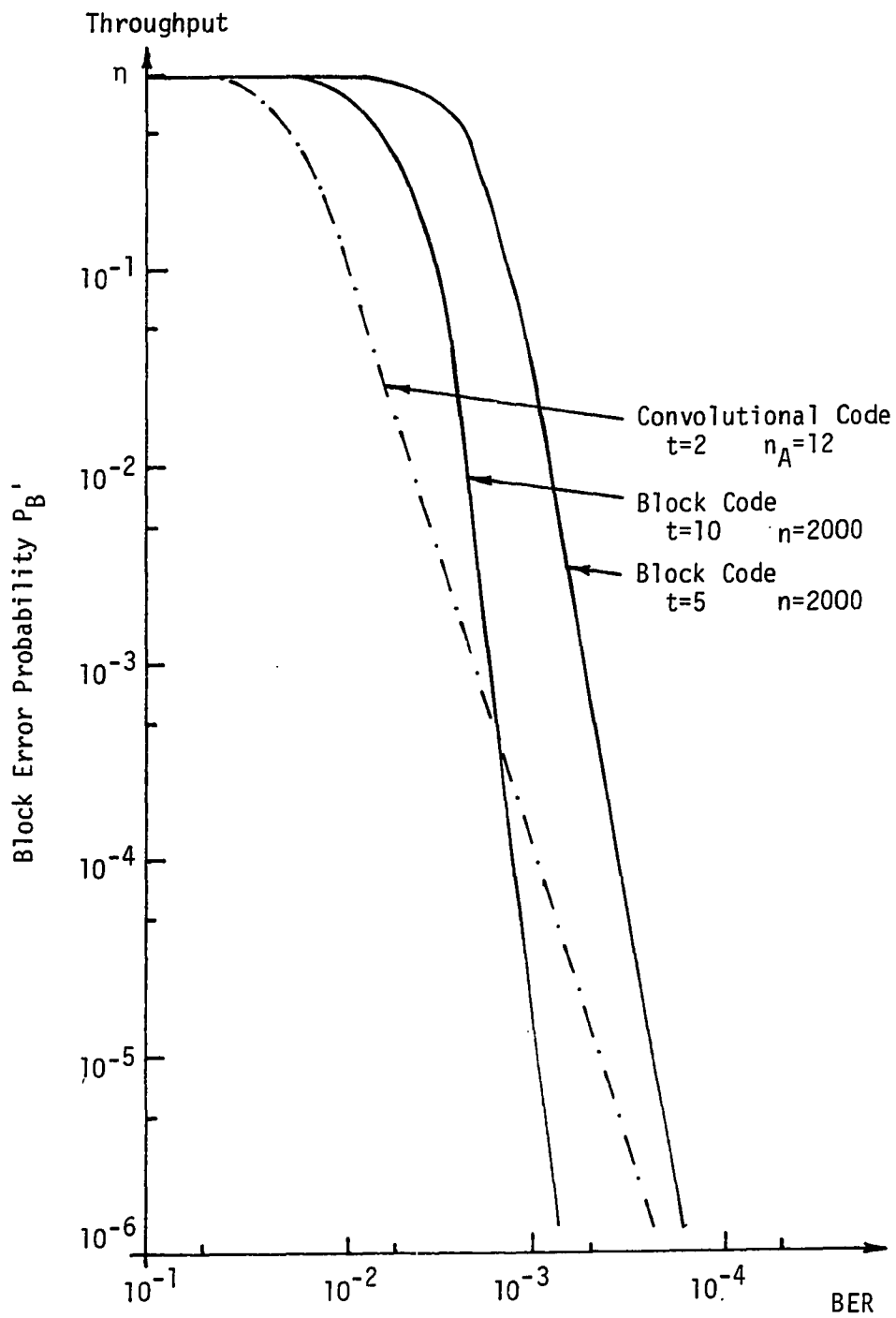


Figure 5.8

One-Half Rate Block Codes Versus Convolutional Codes
 - Block Error Rate Versus Precoder BER

6. RELATED $R = 1/2$ and $R = 1/3$ ORTHOGONALIZABLE CODES

6.1 Introduction

The alternate data-parity hybrid ARQ scheme described in section 3.2 can be seen to perform in an error correction/retransmission mode which in a sense is adaptive. That is, when the channel is error-free a high rate code C_0 is used for error detection only. When channel errors occur, a half-rate code C_1 is introduced by way of a parity block retransmission to attempt to correct the errors. If this is successful then the number of retransmissions of blocks may be drastically reduced so improving the throughput and minimizing delays.

It is also interesting to conjecture whether the adaptive characteristics of the above retransmission scheme could be improved upon. Consider the case when the channel error characteristics are so bad that the information cannot be recovered even after the first two transmission attempts (data and parity types of transmissions respectively) and error-correction procedures based on a half-rate code have failed. An even more powerful error correcting code namely a $(3, 1)$ code may possibly be applied to this situation by sending k extra parity bits in the third transmission attempt so that the receiver can use the k information bits and $2k$ parity bits to attempt to correct all errors. This procedure will only be feasible if codes with appropriate structure can be found.

As described in section 4.5, one possibility might be to look for a class of related code pairs such that any related pair of codes C_1 and C_2 have the following properties:

- C_1 is a $R = 1/2$ convolutional code with generator polynomials $g^{(1)}(x) = 1$ and $g^{(2)}(x)$.
- and
- C_2 is a $R = 1/3$ convolutional code with generator polynomials $g^{(1)}(x) = 1$, $g^{(2)}(x)$ and $g^{(3)}(x)$ where $g^{(1)}(x)$ and $g^{(2)}(x)$ generate C_1 .

In this chapter it is shown that new convolutional codes can be found which satisfy those requirements. These codes provide related pairs of (2, 1) and (3, 1) codes which are orthogonalizable. Since threshold decoding is therefore possible, these codes may have desirable benefits for use in type-II hybrid ARQ where decoder simplicity seems highly desirable.

A retransmission scheme using related (2, 1) and (3, 1) codes is described in Section 4.5. The sequence of transmissions of a given block assuming the information is not recoverable without error after two transmissions is as follows:

- Initial transmission (information) $\{u, P_0(u)\}$
- Second transmission (2, 1) parity $\{P_1(u), P_0(P_1(u))\}$
- Third transmission (3, 1) parity $\{P_2(u), P_0(P_2(u))\}$

The k bits $P_1(u)$ are found by encoding the information u with the (2, 1) code C_1 . Let

$$u = (u_0 \ u_1 \ \dots \ u_{k-1})$$

Then assuming that u is encoded with a systematic (2, 1) code C_1 we obtain the code word

$$y_2 = (y_0^{(1)} \ y_0^{(2)} \ , \ y_1^{(1)} \ y_1^{(2)} \ , \ y_2^{(1)} \ y_2^{(2)} \ , \ \dots \) \quad (6.1)$$

where the information bits are

$$u = (y_0^{(1)} \quad y_1^{(1)} \quad y_2^{(1)} \quad \dots)$$

and the parity bits sent in the second transmission attempt are

$$P_1(u) = (y_0^{(2)} \quad y_1^{(2)} \quad y_2^{(2)} \quad \dots) \quad (6.2)$$

Likewise after encoding the information with a (3, 1) systematic code C_2 we obtain the code word

$$y_3 = (y_0^{(1)} \quad y_0^{(2)} \quad y_0^{(3)} \quad , \quad y_1^{(1)} \quad y_1^{(2)} \quad y_1^{(3)} \quad , \quad y_2^{(1)} \quad y_2^{(2)} \quad y_2^{(3)} \quad \dots) \quad (6.3)$$

from which the parity bits $P_2(u)$ for the third transmission attempt are

$$P_2(u) = (y_0^{(3)} \quad y_1^{(3)} \quad y_2^{(3)} \quad \dots) \quad (6.4)$$

Now the codes C_1 and C_2 must be related in such a way that the parity bits $P_1(u)$ from equation (6.2) are the same as the parity bits

$y_0^{(2)} \quad y_1^{(2)} \quad y_2^{(2)} \quad \dots$ from equation (6.3). Providing such codes can

be found then after the third transmission attempt, the receiver can

combine the $3k$ bits u , $P_1(u)$ and $P_2(u)$ in a form similar to equation

(6.3) for decoding in an attempt to correct all errors. (It should be

noted that although for clarity it has been assumed in the above that

C_1 and C_2 are systematic, this restriction is not strictly necessary).

A description of procedures we have developed for designing appropriate codes follows and related code pairs are tabulated for various degrees of error correcting capability based on best known orthogonalizable codes found by Massey [42]. At this stage procedures for finding related (2, 1)(3, 1) code pairs for convolutional code classes other than the orthogonalizable codes have not been found.

Practical aspects of encoder and decoder hardware are also discussed to demonstrate that the proposed (2, 1)(3, 1) parity retransmission scheme is not excessively complex to implement.

6.2. New orthogonalizable code design

6.2.1 Notation

In order to describe the procedures used to find related (2, 1) (3, 1) code pairs, it is necessary to first summarize the notation used. Typical encoders for (2, 1) and (3, 1) convolutional codes respectively are shown in figure 6.1. In each case the information sequence enters the encoder one bit at a time. For the (2, 1) encoder the output sequences are

$$y^{(1)} = (y_0^{(1)}, y_1^{(1)}, y_2^{(1)}, \dots) \text{ and } y^{(2)} = (y_0^{(2)}, y_1^{(2)}, y_2^{(2)}, \dots)$$

obtained as the convolution of the input sequence u with the two encoder impulse responses $g^{(1)}$ and $g^{(2)}$ more commonly known as the code generator sequences. In polynomial form the encoding equations are conventionally written [2]

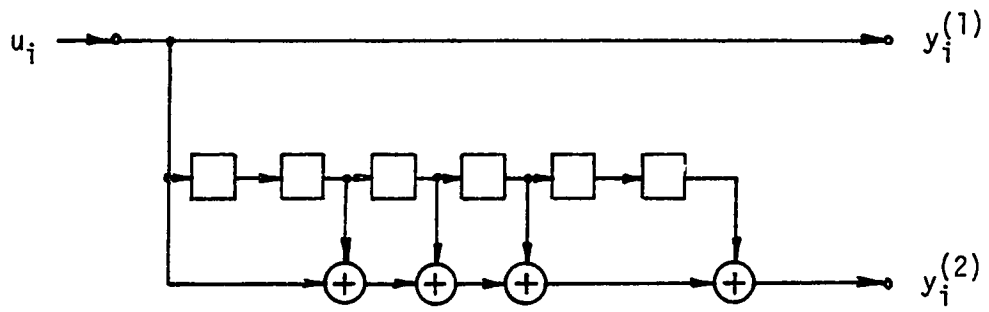
$$\begin{aligned} y^{(1)}(x) &= u(x) g^{(1)}(x) \\ y^{(2)}(x) &= u(x) g^{(2)}(x) \end{aligned}$$

For the (2, 1) encoder in figure 6.1 we have

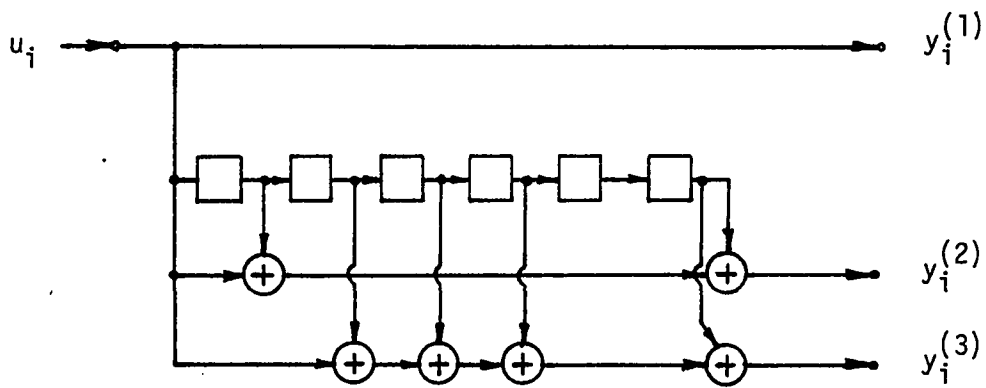
$$g^{(1)}(x) = 1 \text{ and } g^{(2)}(x) = 1 + x^2 + x^3 + x^4 + x^6$$

where the memory order (m) of the code is the highest degree of the $g^{(2)}(x)$ polynomial.

It is convenient to describe convolutional codes in a more abbreviated form by expressing the generator sequences in octal notation. For the above (2, 1) code example, the generator sequences



a. (2, 1) encoder



b. (3, 1) encoder

Figure 6.1

Typical Convolutional Encoders

$$g^{(i)} = (g_0^{(i)} \ g_1^{(i)} \ g_2^{(i)} \ \dots \ g_m^{(i)}) \text{ are}$$

$$g^{(1)} = (1000000) \text{ and } g^{(2)} = (1011101)$$

When expressed in octal form $g^{(2)}$ becomes $(135)^2$. An alternative notation used by Massey [42] for threshold decodable codes denotes the generator polynomials in terms of the positions of the non-zero elements. In Massey's notation the above $(2, 1)$ code would be represented by $(02346)^2$. In similar manner for a $(3, 1)$ systematic code, the generators $g^{(2)}$ and $g^{(3)}$ specify the code. For example, the code represented in figure 6.1.(b) is described using Massey notation in the form $(0, 1, 7)^2 (0 2 3 4 6)^3$. It is also noted that one advantage of systematic codes is that they are always non-catastrophic; actually the majority of non-systematic $(n, 1)$ codes are also non-catastrophic so it is not difficult to find codes for which error propagation can be avoided.

The minimum free distance measure d_{free} for convolutional codes is a most important distance measure [2] defined

$$d_{\text{free}} = \min \{d(y', y'') : u' \neq u''\} \quad (6.5)$$

where u' and u'' are the information sequences corresponding to the code words y' and y'' respectively. Tables of $(2, 1)$ and $(3, 1)$ codes with maximum d_{free} have been found for different memory orders m by computer search and are available in reference [2]. It should be noted that the systematic code d_{free} values are less than or equal to those of non-systematic codes with a given rate and constraint length. For a $(2, 1)$ $(3, 1)$ parity retransmission ARQ application, non-systematic codes could possibly be used and provide better error-correction performances. However an inverting circuit would be required at the receiver to recover the input data from the first received sequence if recovered without detectable error.

Orthogonal codes and orthogonalizable codes are important subclasses of convolutional codes. Their primary importance for a hybrid ARQ lies in the fact that simple majority logic or threshold decoding can be used. Apart from the relative simplicity of the decoder, another important advantage in an ARQ application is that there are minimum decoding delays contributed to the overall round-trip delay. As shown in sections 2 and 4, such delays can very significantly deteriorate the throughput in an ARQ system. More powerful decoding procedures such as Viterbi decoding or sequential decoding involve greater decoding effort and may result in long delays. Since in a hybrid ARQ system there is always provision for retransmission if certain error patterns cannot be corrected, it is postulated in what follows that threshold decodable codes therefore offer the most acceptable practical compromise between performance and complexity. In threshold decoding the final decision is made on only one constraint length (n_A) of received blocks where the encoding constraint length is defined for an (n, k) code of memory order m as

$$n_A = n(m + 1) \quad (6.6)$$

Threshold decoding is based on the concept of orthogonal parity check sums being any syndrome bit or any sum of syndrome bits. Given a channel error sequence for the k information bits of the form

$$e^{(1)} = (e_0^{(1)}, e_1^{(1)}, e_2^{(1)}, \dots)$$

we obtain the combined error sequence for a $(2, 1)$ code as

$$e = (e_0^{(1)} e_0^{(2)}, e_1^{(1)} e_1^{(2)}, e_2^{(1)} e_2^{(2)}, \dots)$$

6.2.2. Design of related code pairs

Orthogonalizable codes have been found by Massey [42] using a trial and error approach. A set of J check sums orthogonal on an error bit e_j is required such that

- each check sum checks e_j , but
- no other error bit is checked by more than one check sum.

In the case of orthogonalizable codes, some of the orthogonal parity checks are formed from sums of syndrome bits. (By contrast, for self-orthogonal codes, the parity checks used are individual syndrome bit checks.) A t -error correcting code is one for which the error bits checked by the J orthogonal check sums contain t or fewer errors with $t = \lfloor J/2 \rfloor$.

Consider a $(2, 1)$ code of memory order m_2 generated by

$$g^{(2)} = (g_0, g_1, \dots, g_{m_2})$$

Consider also a $(3, 1)$ code of memory order m_3 with $m_3 \geq m_2$ generated by

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, g_2^{(2)} \dots)$$

and

$$g^{(3)} = (g_0^{(3)}, g_1^{(3)}, g_2^{(3)} \dots)$$

then we seek related $(2, 1)$ and $(3, 1)$ code pairs for hybrid ARQ for which

$$g_0 = g_0^{(i)} \quad g_1 = g_1^{(i)} \quad \dots \quad g_{m_2} = g_{m_2}^{(i)} \quad (6.10)$$

for $i = 2$ or 3 .

Both codes must be self-orthogonal or orthogonalizable so that a common majority logic decoder circuit can be used. New related orthogonalizable code pairs have been found by considering optimum (3, 1) orthogonalizable codes from Massey [42] and finding related orthogonalizable (2, 1) codes for each. It has not been found possible to find related pairs of self-orthogonal codes using parity-triangle design procedures [2] although these were investigated.

The procedure used to find orthogonalizable related (2, 1) codes was based on an exhaustive trial-and-error analysis of the parity check matrix for each optimum (3, 1) code. This has been achieved for (3, 1) codes with t-error correcting ability of $t = 1, 2, 3, \dots, 8$. To illustrate the method, consider the optimum (3, 1) orthogonalizable code with $t = 4$. It is the code described above by generators given in Massey notation by

$$(0\ 1\ 7)^2 \equiv (1\ 1\ 0\ 0\ 0\ 0\ 0\ 1)$$

$$\text{and } (0\ 2\ 3\ 4\ 6)^3 \equiv (1\ 0\ 1\ 1\ 1\ 0\ 1\ 0)$$

for which the encoder is shown in figure 6.1. The parity check matrix for this code follows from (6.9) as

respectively providing a $t=2$ error correcting $(2, 1)$ code with effective constraint length $n_E = 18$ bits. That is the $(2, 1)$ code is capable of correcting up to 2 errors in an effective constraint length of 18 bits.

The check sums follow as

$$\begin{aligned}
 s_0 &= e_0^{(1)} + e_0^{(2)} \\
 s_2 &= e_0^{(1)} + e_2^{(1)} + e_2^{(2)} \\
 s_3 &= e_0^{(1)} + e_1^{(1)} + e_3^{(1)} + e_3^{(2)} \\
 s_1 + s_4 + s_8 &= e_0^{(1)} + e_1^{(2)} + e_4^{(2)} \\
 &\quad + e_5^{(1)} + e_6^{(1)} + e_7^{(1)} + e_8^{(1)} + e_8^{(2)}
 \end{aligned}
 \tag{6.13}$$

Similar procedures were used for all other optimum orthogonalizable $(3, 1)$ codes for $t=1$ to 8. The results are given in Table 6.1 which lists the generator for the $(2, 1)$ codes found by the above procedure as well as the rules for forming their orthogonal checks.

Table 6.1

Related Orthogonalizable Code Pairs

New Orthogonalizable--Pair (2,1) Codes				Best Orthogonalizable (2,1) Code for Comparison		Massey's Orthogonalizable (3,1) Codes		
t	$\frac{n_E}{n}$	Gen ^r Sequence	Rules for Forming Check Sums	t	$\frac{n_E}{n}$	t	$\frac{n_E}{n}$	Generator Sequences
1	2	$(0,1)^2$	(0)(1)	1	2	2	3	$(0,1)^2(0,2)^3$
2	8	$(0,2,3,4)^2$	(0)(2)(1,3)(4,5,7)	2	6	3	5	$(0,1)^2(0,2,3,4)^3$
2	9	$(0,2,3,4,6)^2$	(0)(2)(3)(1,4,8)	2	6	4	8	$(0,1,7)^2(0,2,3,4,6)^3$
2	15	$(0,1,2,3,5,8,9)^2$	(0)(1)(5)(2,3,8,14)	2	6	5	11	$(0,1,9)^2(0,1,2,3,5,8,9)^3$
2	7	$(0,4,5,6)^2$	(0)(4)(5)(1,6)			6	18	$(0,4,5,6,7,9,12,13,16)^2$ $(0,1,14,15,16)^3$
	or			2	6			
3	28	$(0,4,5,6,7,9,12,13,16)^2$	(0)(4)(5)(16) (2,9,20,21)(7,10,11,27)	3	12			
2	7	$(0,4,5,6)^2$	(0)(4)(5)(1,6)	2	6	7	23	$(0,4,5,6,7,9,12,13,16,19,20,21)^2$ $(0,1,20,22)^3$
2	7	$(0,4,5,6)^2$	(0)(4)(5)(1,6)	2	6	8	36	$(0,4,5,6,7,9,12,16,17,30,31)^2$ $(0,1,22,25,35)^3$
	or							
3	22	$(0,4,5,6,7,9,12,16,17)^2$	(0)(4)(5)(1,6) (2,9,21)(7,10,11)	3	12			

Also shown in Table 6.1 for comparison are the optimum orthogonalizable (2,1) codes from Massey [49]. It can be seen that the new (2,1) codes found for related pairs are not as powerful as the optimum (2,1) codes. However they do provide a means for implementing the adaptive (2,1)(3,1) parity retransmission hybrid ARQ scheme.

6.3 Encoding and Decoding

We consider now typical encoder and decoder configurations for the related half-rate and one-third rate codes. A typical encoder was illustrated in Fig. 6.1(b). It can be used to encode the related orthogonalizable code pair given in Table 6.1 as follows

$$\begin{array}{ll} (2,1) \text{ code} & \text{Generator } (0,2,3,4,6)^2 \\ (3,1) \text{ code} & \text{Generator } (0,1,7)^2 (0,2,3,4,6)^3 \end{array}$$

Recall from section 6.1 that both codes are systematic. In a hybrid ARQ system using these codes, the first block transmitted is a data block $\{u, P_0(u)\}$. The k bits u are available at the output terminal $y_i^{(1)}$ in Fig. 6.1(a) and (b). The last m bits are taken to be zeros (m being the code memory order). Then u must be encoded with a high-rate block code C_0 to produce the $n-k$ parity bits $P_0(u)$ for error detection.

The receiver computes the syndrome for the received data block $\{\hat{u}, \hat{P}_0(u)\}$. If it is non-zero, a NAK is sent back to the transmitter. The transmitter then sends the parity block $\{P_1(u), P_0(P_1(u))\}$. The k bits $P_1(u)$ are obtained at the output terminal marked $y_i^{(3)}$, having been held in the transmitter buffer until required. As before the $k-n$ parity bits $P_0(P_1(u))$ are obtained by encoding $P_1(u)$ with the high rate block code C_0 . The $2k$ bits u and $P_1(u)$ constitute a half-rate code word.

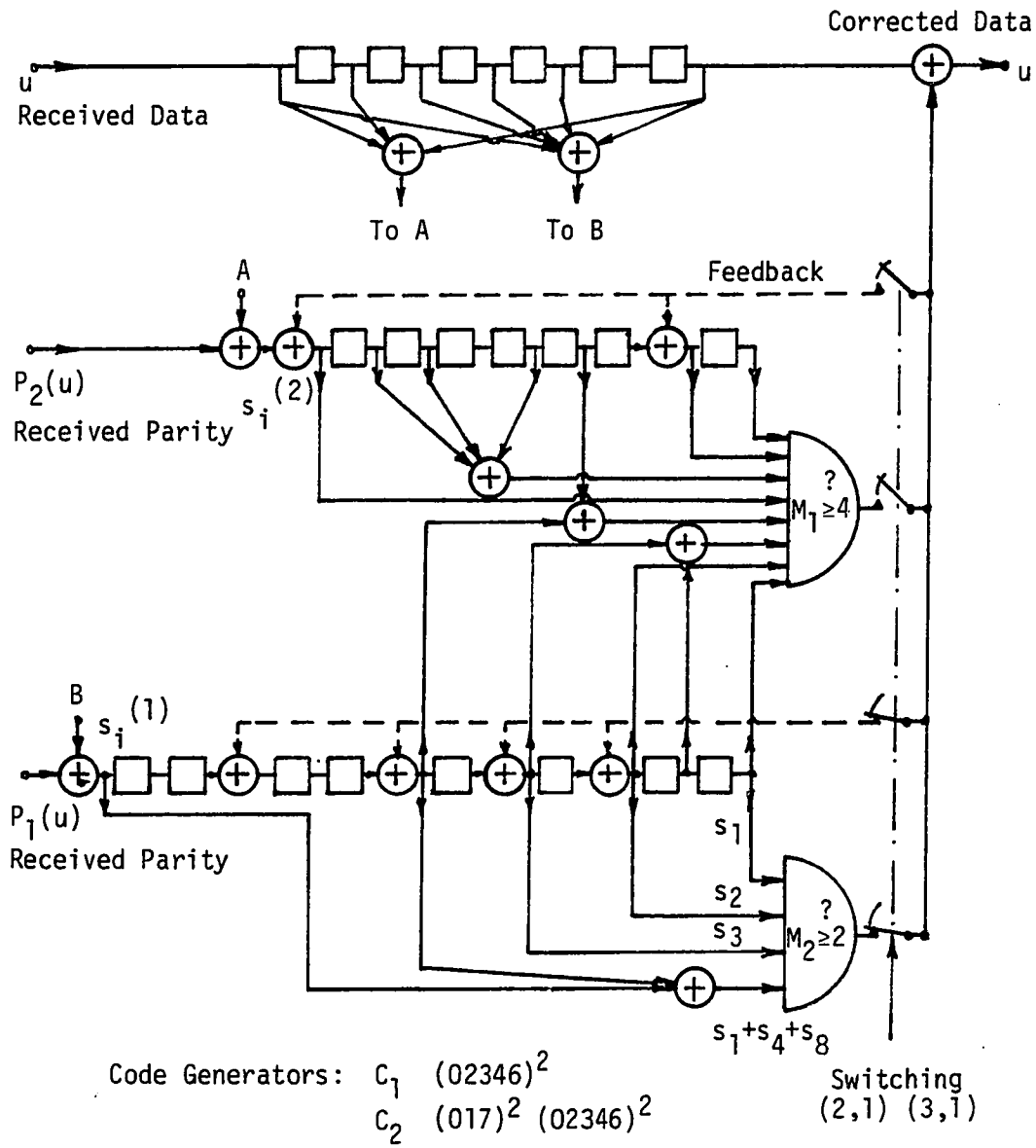


Figure 6.2

Threshold Decoder for (2,1) Code C_1
or Related (3,1) Code C_2

The receiver attempts to recover the data error-free from the received data and parity blocks as discussed in section 3.3. If this is unsuccessful a NAK is again sent back to the transmitter. The transmitter next sends the parity block $\{P_2(u)P_0(P_2(u))\}$ where $P_2(u)$ is obtained from the terminal marked $y_i^{(2)}$ in Fig. 6.1(b). The 3k bits $u, P_1(u)$ and $P_2(u)$ constitute a one-third-rate code word with greater error correcting capability. Should the receiver fail to recover the data estimate \tilde{u} without detectable errors, the transmitter repeats the data block $\{u, P_0(u)\}$. When $\{\hat{u}, \hat{P}_0(u)\}$ is received, it is used together with $P_1(u)$ and $P_2(u)$ to recover \tilde{u} . The process continues until \tilde{u} is recovered. When this has occurred the transmitter clears its encoder and prepares for the next data block to be encoded.

It is also possible to use a single decoder circuit to decode the half-rate and one-third-rate code words respectively. Figure 6.2 shows a decoder suitable for decoding the (2,1) and (3,1) codes just described. In order to explain its operation we consider first a conventional threshold decoder for an (n,1) code of memory order m. The conventional threshold feedback decoder (see for example reference [2]) first computes the syndrome. This is done by passing the first m received data bits \hat{u} through an encoder circuit. The parity bits thus generated are added modulo-2 to the received parity bits to produce the n-1 syndrome sequences, one corresponding to each parity error sequence. Each syndrome sequence corresponding to one constraint length consists of m+1 bits. Next the decoder forms a set of orthogonal check sums on the first information error bit. The set of check sums is formed from the syndrome bits calculated above. Table 6.1 lists the check sums required for each of

the new (2,1) codes. The set of check sums is fed into a threshold gate which produces an output of "1" if and only if more than one half of its inputs are "1's". If this occurs the first data bit is assumed incorrect. It is corrected by adding the output of the threshold gate to the first received bit. The output of each threshold gate is also fed back and subtracted from each syndrome it affects. The first estimated information bit is then shifted out of the encoder. The syndrome registers are shifted one to the right. The next block of n received bits is shifted into the decoder, and the process of computing the next set of $n-1$ syndrome bits is continued as explained above. They are then shifted into the $n-1$ left most stages of the syndrome registers which then hold modified syndromes along with the new set of syndrome bits. The decoder again uses the threshold decision procedures to estimate whether an error has occurred in the second information bit. The process continues until all information bits are decoded.

We now consider how the above procedure can be executed for either the (2,1) code or the (3,1) code using the common circuit of Fig. 6.2. If the receiver wishes to decode \hat{u} using the (2,1) code bits \hat{u} and $\hat{P}_1(u)$ the switches in Fig. 6.2 are left as shown. For the code with generator $(0,2,3,4,6)^2$, the check sums to be formed were specified in equation (6.13). These four check sums are s_0 , s_2 , s_3 and $(s_1+s_4+s_8)$ respectively. They are shown in Fig. 6.2. A given information bit is assumed in error if the threshold gate finds 3 or 4 of these are "1's" and decoding proceeds as described above. This code can correct up to 2 errors in a constraint length of 18 bits. Methods for detecting whether \tilde{u} is decoded without detectable errors are described in chapter 7.

If the receiver does not recover \tilde{u} successfully from \hat{u} and $\hat{P}_1(u)$, the transmitter sends the additional parity bits $P_2(u)$. As described, $\hat{u}, \hat{P}_1(u)$ and $\hat{P}_2(u)$ constitute a (3,1) code word. The receiver is assumed to hold \hat{u} and $\hat{P}_1(u)$ in a buffer until the data estimate \tilde{u} is recovered successfully. The decoder in Fig. 6.2 is then cleared and the switches changed over to the opposite positions to those shown. The new check sums required for decoding were specified by Massey [49]. They are $s_0^{(1)}, s_0^{(2)}, s_1^{(1)}, s_2^{(2)}, (s_1^{(2)} + s_3^{(2)}), (s_2^{(1)} + s_4^{(2)}), s_7^{(1)}, (s_3^{(1)} + s_5^{(1)} + s_6^{(1)} + s_6^{(2)})$. This code can correct up to 4 errors in a constraint length of 24 bits. As can be seen in Fig. 6.2 these check sums are applied to the threshold gate. An error is assumed detected if more than 4 are found to be "1's". The decoder proceeds as described above.

The decoder illustrated in Fig. 6.2 is a relatively simple circuit. It clearly illustrates the ease with which threshold decoding can be accomplished. The decoder illustrated in Fig. 6.2 is a feedback decoder. Past estimates of information bits are fed back and used to modify the syndrome. If feedback is not used, the process is referred to as 'definite' decoding. Feedback decoders can give rise to an error propagation effect [2]. This may be undesirable in FEC systems without ARQ retransmission facilities. However in an ARQ system, retransmission is requested if one or more decoding errors occur. Therefore error propagation is not a disadvantage. On the other hand definite decoding eliminates error propagation but weakens the error-correction properties of the code. The effective constraint length is always increased [2]. Also for codes which are not self-orthogonal, the majority-logic error correcting capability t of the code may be reduced. In a Type-II hybrid ARQ system definite decoding would reduce the throughput and is therefore undesirable.

7. PERFORMANCE OF HYBRID ARQ SCHEMES

7.1 Introduction

Expressions giving throughput efficiency for various ARQ protocols in conjunction with hybrid schemes were derived in Chapter 4. Throughput was shown to be a function of channel bit error rate (p) and round-trip delay (S). For hybrid ARQ schemes, an additional effect on throughput was expressed in terms of probabilities (q_1, q_2, \dots) that indicate the effectiveness of the forward-error correction procedures being used.

For convenience, these probabilities are redefined here:

- q_1 is the conditional probability that a message u is recovered successfully after the first retransmission given that errors were detected in the initial transmission. (For a Type-II alternate data/parity hybrid scheme as described in section 3.2, q_1 expresses the probability that u will be recovered either by an inversion process that is, no errors in the parity retransmission, or by error correction decoding using a half-rate error correction code.)
- q_2 is the conditional probability that a message u is successfully recovered from the third transmission (second retransmission) given that u was not recovered without error after the first and second transmission attempt of a given message (or its 1/2 rate parity equivalent.)

For some hybrid schemes, it may be useful to define further probabilities q_3, q_4, \dots in similar mode to the above. This would

apply for example to the alternate data/parity scheme referred to in section 3.2 for which

$$q_1 \neq q_2 \neq q_3 \dots$$

Let $P(C_i/E_1E_2\dots E_{i-1})$ represent the probability of recovering the message u after the i^{th} transmission attempt (including data and parity transmissions) given that uncorrectable errors remained after transmissions 1, 2, ..., $i-1$.

Then this can be related to the various error detection/correction ARQ schemes described in sections 3 and 4 as follows:

- Alternate data/parity scheme (section 3.2)

$$P(C_i/E_1E_2\dots E_{i-1}) = \begin{cases} 1 - P_B & i = 1 \\ q_i & i = 2, 3, \dots \end{cases} \quad (7.1)$$

- Inferior data/parity scheme (section 3.4)

$$P(C_i/E_1E_2\dots E_{i-1}) = \begin{cases} 1 - P_B & i = 1, 4, 7, \dots \\ q_1 & i = 2, 5, 8, \dots \\ q_2 & i = 3, 6, 9, \dots \end{cases} \quad (7.2)$$

- 1/2, 1/3 parity retransmission scheme (section 3.3)

$$P(C_i/E_1E_2\dots E_{i-1}) = \begin{cases} (1 - P_B) & i = 1, 4, 7, \dots \\ q_1 & i = 2, 5, 8, \dots \\ q_2 & i = 3, 6, 9, \dots \end{cases} \quad (7.3)$$

We now proceed to examine methods for evaluating the conditional probabilities q_1 and q_2 for different cases. The evaluation of q_3 , q_4 etc. appears too complex for hybrid ARQ schemes of interest to warrant the extensive work involved.

The parameters q_1 and q_2 will depend on the nature of the forward-error correction coding procedures being used. Evaluations for block codes will be compared with those using convolutional codes. This then will permit evaluation of throughput efficiency for a large class of ARQ systems. It has been explained previously that for convenient performance analysis, any ARQ system description should comprise a particular combination of selected

- ARQ retransmission protocol
(e.g. Go-Back-N, SR + ST, SR + GBN, etc.)
- error detection/recovery scheme
(e.g. alternate data/parity retransmission, 1/2 1/3 parity retransmission etc.)
- forward-error correction coding/decoding procedure
(e.g. (2000,1000) block code with 10 error correcting decoder, (2,1) convolutional code with sliding block decoding, etc.)

7.2 Conditional Probabilities q_1 and q_2 for block codes

Consider first a Type-II hybrid scheme in which a half-rate block code is used. The first transmission (the "information transmission") $\{u, P_0(u)\}$ consists of k information bits plus $n-k$ parity bits based on an (n, k) high rate block code C_0 to be used for error detection only. If errors are detected in the first received block, then the second transmission attempt (the "parity transmission") $\{P_1(u), P_0(P_1(u))\}$ consists of n parity bits. The first k bits $P_1(u)$ are the parity bits generated by a half-rate block code C_1 designed for error correction. The other $n-k$ bits $P_0(P_1(u))$ are generated by encoding $P_1(u)$ with code C_0 for error detection.

On receipt of the parity transmission the receiver computes the syndrome using code C_0 . If the syndrome is non-zero, errors are detected. The receiver then attempts to correct the errors in the combined $2k$ bit vector $\{\hat{u}, \hat{P}_1(u)\}$ where \hat{u} and $\hat{P}_1(u)$ are the received information and code C_1 parity respectively. It is assumed that a t -error correction decoder is used. It should be noted here that practical considerations in respect of decoder complexity may dictate that the decoder t parameter may be considerably less than the maximum error-correcting capability of the code. For example a $(1000,500)$ shortened BCH code is available which is capable of correcting up to 50 errors. However the following analysis takes into account the likelihood that a decoder with t -error correcting ability of the order of 10 or less would be used. The parameter q_1 represents the conditional probability of recovering the data \tilde{u} at this stage either by inversion of $\hat{P}_1(u)$ if error-free or by correcting errors in $\{\hat{u}, \hat{P}_1(u)\}$.

Should the decoder fail to correct the errors, then the next transmission of that block would be a repeated "information transmission". The receiver carries out the same procedure as described following the "parity transmission". The parameter q_2 represents the conditional probability of successfully recovering \tilde{u} at this stage.

Consider therefore an error correction procedure such that for any data/parity transmission pair, the decoder is capable of correcting up to t errors. Then q_1 can be found (Lin [37]) as follows

$$q_1 = (1-P_B) + \frac{1}{P_B} \left\{ \sum_{i=0}^t \binom{2k}{i} p^i (1-p)^{2k-i} + (1-p)^{2n} - 2(1-p)^n \left[(1-p)^k + \sum_{i=1}^t \binom{k}{i} p^i (1-p)^{k-i} \right] \right\} \quad (7.4)$$

where $P_B = 1 - (1-p)^n$, $p = \text{BER}$ and $n = \text{block size}$.

The first term in Equation (7.4) represents the probability that the first received block $\{\hat{u}, \hat{P}_0(u)\}$ contains detectable errors. The second term represents the probability that after transmission 2, the received vector $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ contains detectable errors but the error patterns in transmissions 1 and 2 were correctable, given that the first transmission was received with errors detectable.

Also, for block codes, q_2 can be evaluated using the following lower bound [37]

$$q_2 \geq (1-P_B) + \frac{1}{P_B(1-q_1)} \left\{ \sum_{i=1}^{t-1} \Delta_i S_{t-i} (1-\Delta_0 - S_{t-i}) \right\} \quad (7.5)$$

$$\text{where } \Delta_i = \binom{k}{i} p^i (1-p)^{k-i}$$

$$\text{and } S_j = \sum_{\ell=1}^j \Delta_\ell$$

7.3 q_1 and q_2 for Convolutional Codes

Consider now the evaluation of the conditional probabilities q_1 and q_2 for the case where a convolutional code is used for forward-error correction using a "sliding block" decoding scheme such as threshold decoding. A data-parity-data retransmission sequence is assumed as in section 7.2. Likewise the receiver operation is also assumed to be as discussed in 7.2 with error-correction decoder capable of correcting up to t errors in any constraint length of n_A bits.

Using the definition of q_1 and following the method of Lin and Yu [37] used to derive (7.4) and (7.5), we can write

$$q_1 = P_r\{(C_2 \cup D_2 \cup F_2)/D_1\} \quad (7.6)$$

where the events C_2 , D_1 , D_2 and F_2 are defined as follows:

C_2 is the event {the received parity block $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ at the second transmission contains no detectable errors} D_i is the event {the block received at the i^{th} transmission/retransmission contains detectable errors}

i.e. D_1 is the event $\{\{\hat{u}, \hat{P}_0(u)\}$ contains detectable errors} and D_2 is the event $\{\{\hat{P}_1(u), \hat{P}_0(u)\}$ contains detectable errors} F_2 is the event {the error patterns in the received data/parity pair $\{\hat{u}, \hat{P}_1(u)\}$ are correctable by decoding}.

Note that • \hat{u} is the received k -bit message block,

• $\hat{P}_0(u)$ is the set of $(n-k)$ parity detection bits computed for u using a high rate block code C_0 ,

• $\hat{P}_1(u)$ represents the k parity bits generated by the $1/2$ rate convolutional code C_1 for the message bits u .

and

• $\hat{P}_0(P_1(u))$ represents the $n-k$ parity detection bits generated by code C_0 for the bits $P_1(u)$.

(In each case, the symbol \hat{x} represents a received vector x .) Now since D_1 and C_2 are independent events

$$\begin{aligned} q_1 &= P_r\{C_2\} + \frac{P_r\{D_1 D_2 F_2\}}{P_r\{D_1\}} \\ &= (1-P_B) + \frac{P_r\{D_1 D_2 F_2\}}{P_B} \end{aligned} \quad (7.7)$$

Appendix D shows how the joint probability $P_r\{D_1 D_2 F_2\}$ can be calculated. As a result we obtain for a convolutional code C_1 capable of correcting $\leq t$ errors in any constraint length of n_A bits:

$$\begin{aligned}
q_1 = (1-P_B) + \frac{1}{P_B} \left\{ \sum_{j=2}^{\infty} N(j) p^j (1-p)^{2k-j} \right. \\
- 2(1-p)^k \sum_{j=2}^{\infty} N^1(j) p^j (1-p)^{k-j} \\
+ 2(1-p)^k [1-(1-p)^{n-k}] \sum_{j=1}^{\infty} N^1(j) p^j (1-p)^{k-j} \\
\left. + (1-p)^{2k} [1-(1-p)^{n-k}]^2 \right\} \quad (7.8)
\end{aligned}$$

where

- $N(j)$ is the number of correctable patterns of j errors in a block of $2k$ bits such that no more than t errors occur in any span of n_A bits in the block
- $N^1(j)$ is the number of correctable patterns of j errors in a block of k bits such that no more than t errors occur in any span of $n_A/2$ bits in the block

and as before

- $P_B = 1-(1-p)^n$

The similarities in form between Equations (7.8) and (7.4) for convolutional and block codes should be noted. Methods for computing $N(j)$ and $N^1(j)$ were discussed in Section 5.

Let $P_B(N_B, n_A)$ represent the block error probability after decoding N_B bits with a decoder capable of correcting up to t errors in any "sliding window" span n_A . Note that the symbol $P_B(N_B, n_A)$ has the same meaning as the symbol P_B' used to denote block error probability in section 5 but with the arguments N_B and n_A stated explicitly. Values for $P_B(N_B, n_A)$ can therefore be computed using either the trellis algorithm of section 5.3 or the Combinatorial approach of section 5.4. In practice

it was found convenient to combine elements of both methods. For example considering the terms in Equation (7.8), use

$$\begin{aligned} \sum_{j=2}^{\infty} N(j) p^j (1-p)^{2k-j} &= \left\{ 1 - P_B(2k, n_A) \right\} - \sum_{j=0,1} N(j) p^j (1-p)^{2k-j} \\ &= \left\{ 1 - P_B(2k, n_A) \right\} - (1-p)^{2k} - 2k p (1-p)^{2k-1} \end{aligned} \quad (7.9)$$

where the term in brackets on the right hand side of (7.9) can be computed using the trellis algorithm. Likewise,

$$\sum_{j=1}^{\infty} N^1(j) p^j (1-p)^{k-j} = \left\{ 1 - P_B(k, \frac{n_A}{2}) \right\} - (1-p)^k \quad (7.10)$$

Consider now the evaluation of the conditional probability q_2 defined in section 7.1. As shown in Appendix D, for the case where a 1/2 rate convolutional code is used in a forward-error correction mode, we use

$$\begin{aligned} q_2 \geq (1 - P_B) + \frac{1}{P_B(1 - q_1)} \left\{ \sum_{i \geq 1} \Delta_i \sum_{j=1}^{\infty} N''(j) p^j (1-p)^{k - i n_A / 2 - j} \right. \\ \left. [1 - \sum_{j=0}^{\infty} N^1(j) p^j (1-p)^{k-j}] \right\} \end{aligned} \quad (7.11)$$

where $\Delta_i = \binom{k}{i} p^i (1-p)^{k-i}$ as before in relation to (7.5) and $N^1(j)$ is defined as explained above in relation to Equation (7.8) for q_1 . $N''(j)$ is defined in the same way except that it refers to the number of correctable error patterns in a block of length $(k - i n_A / 2)$ bits as indicated by the power of the associated $(1-p)$ term in Equation (7.11).

Details are given in Appendix D.

7.4 Throughput for Type-II Hybrid ARQ Schemes

The computation of throughput can now be completed for various combinations of the following:

- ARQ retransmission protocols (see sections 2 and 4)
- hybrid error correction scheme (see sections 3.2, 3.3 and 4)

and

- convolutional or block codes (as per sections 7.2 and 7.3).

Fortran programs have been used to evaluate the throughput for a number of different hybrid ARQ combinations and the results are depicted in Figures 7.1 to 7.4. In particular, the following schemes were analyzed:

- | | |
|--------------|-------------------------------------------------------------------------------------------------------------|
| Scheme (i) | Type-II hybrid scheme with |
| | - SR+ST retransmission protocol |
| | - alternate data/parity hybrid scheme |
| | with - various half-rate convolutional codes. |
| Scheme (ii) | As above but using SR+GBN retransmission protocol |
| Scheme (iii) | As above but using Lin and Yu retransmission protocol |
| Scheme (iv) | Type-II hybrid scheme |
| | as above but using the ideal selective-repeat retransmission protocol (requiring infinite receiver buffer). |

For scheme (i), the throughput is given by Equation (4.2) combined with Equations (7.8) and (7.11).

For scheme (ii), Equation (4.5) replaces (4.2).

For scheme (iii), Equation (4.7) is used.

For scheme (iv), Equation (4.6) is used.

Figure 7.1 shows throughput plotted versus BER for a random error channel for a round-trip delay $S=1024$ blocks. Throughput plots are shown for the following FEC codes:

Code (a) - a convolutional code with minimum free distance $d_{\text{free}}=5$ and constraint length $n_A=6$ bits is assumed which implies a

nonsystematic code as set out in Table 5.1. For such a code, ROM table lookup decoding would be desirable. There is no orthogonalizable code with such parameters so threshold decoding is not possible. Since a nonsystematic code must be used, an inversion circuit will be necessary to recover the message \tilde{u} after the first transmission of any block if no errors are detected in the first transmission attempt.

Code (b) - a convolutional code with $d_{\text{free}}=5$ and $n_A=12$. As shown in Table 5.1, a systematic orthogonalizable code exists with these parameters. As a result, no inversion circuit is required at the receive end to recover the data \tilde{u} if no errors are detected in the first received transmission attempt. A very simple threshold decoder can be used.

Throughput values are given in Figure 7.1 for $v=1$ selective-repeat retransmissions. As discussed in section 2, this means that after the second attempt at transmission of a given data block, subsequent retransmissions are made in the "stutter" mode (for the SR+ST scheme) or in the Go-Back-N mode (for the SR+GBN scheme). This ensures that receiver buffer overflow cannot occur providing that a receiver buffer store S_R is provided as follows:

$$S_R = \begin{cases} S(v+1) \text{ blocks for the SR+ST scheme} \\ v(S+1) \text{ blocks for the SR+GBN scheme} \end{cases} \quad (7.12)$$

Figure 7.1 shows that the SR+ST scheme throughput is greater than that of the SR+GBN for BER rates greater than $p=10^{-3}$. This shows the benefit to be gained at the expense of increased buffer store. Figure 7.1 also shows the benefit to be gained by use of a nonsystematic code with

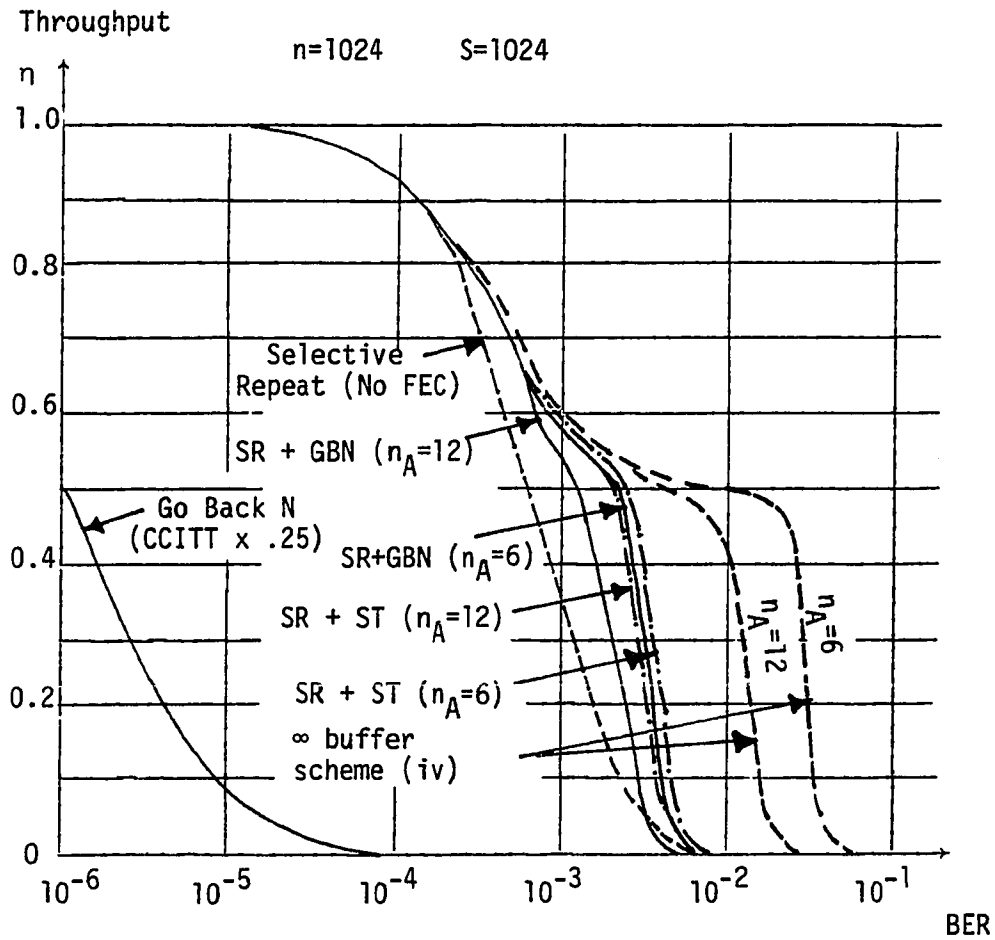


Figure 7.1

Throughput of Alternate Data/Parity Hybrid ARQ with SR + ST and SR + GBN Protocols ($v=1$)

FEC Codes Used: $n_A=6$ $d_{free}=5$ (nonsystematic non-orthogonalizable)
 $n_A=12$ $d_{free}=5$ (systematic orthogonalizable)

appropriate inverter and decoder versus an orthogonalizable code with very simple decoder circuit.

Figure 7.2 shows how throughput varies with BER for scheme (iii) (i.e. Lin and Yu retransmission protocol). Results for both convolutional and block codes are shown for comparison. As expected from the results in Section 5, the simple convolutional codes out perform the $t=10$ block code shown and also the "ideal" selective-repeat ARQ scheme with infinite receiver buffer but no forward-error correction.

Both Figures 7.1 and 7.2 show how throughput varies for the infinite receiver buffer hybrid selective-repeat scheme (iv) for a convolutional code with $d_{\text{free}}=5$ and $n_A=6$. This result also provides the throughput values (with one percent accuracy) for a terrestrial circuit with say $S=1$ round-trip delay for either scheme (i) or (ii).

The use of a more powerful FEC convolutional code is illustrated in Figure 7.3. Here the FEC code used is assumed to have $d_{\text{free}}=7$ with $n_A=10$ (refer Table 5.1). The results are shown for the alternate data/parity schemes (i) and (ii).

Figure 7.4 illustrates the performance of the ST+GBN scheme described in section 4.6. Throughput values are shown for cases with and without forward-error correction. For comparison, plots of throughput are given for the SR+GBN protocol and the ST+GBN scheme. For the ST+GBN scheme we have used $n_1=3$ in Equation (4.8). The parameter n_1 specifies the number of "stutter" retransmissions sent following a NAK being received at the transmitter (refer section 4.6). When no forward-error correction is used, the optimum value of n_1 is a function of the round-trip delay S and block-error rate P_B . Figure 7.4 is drawn for $S=1024$ blocks. For this case, $n_1=3$ is a reasonable choice. However, it is not the best

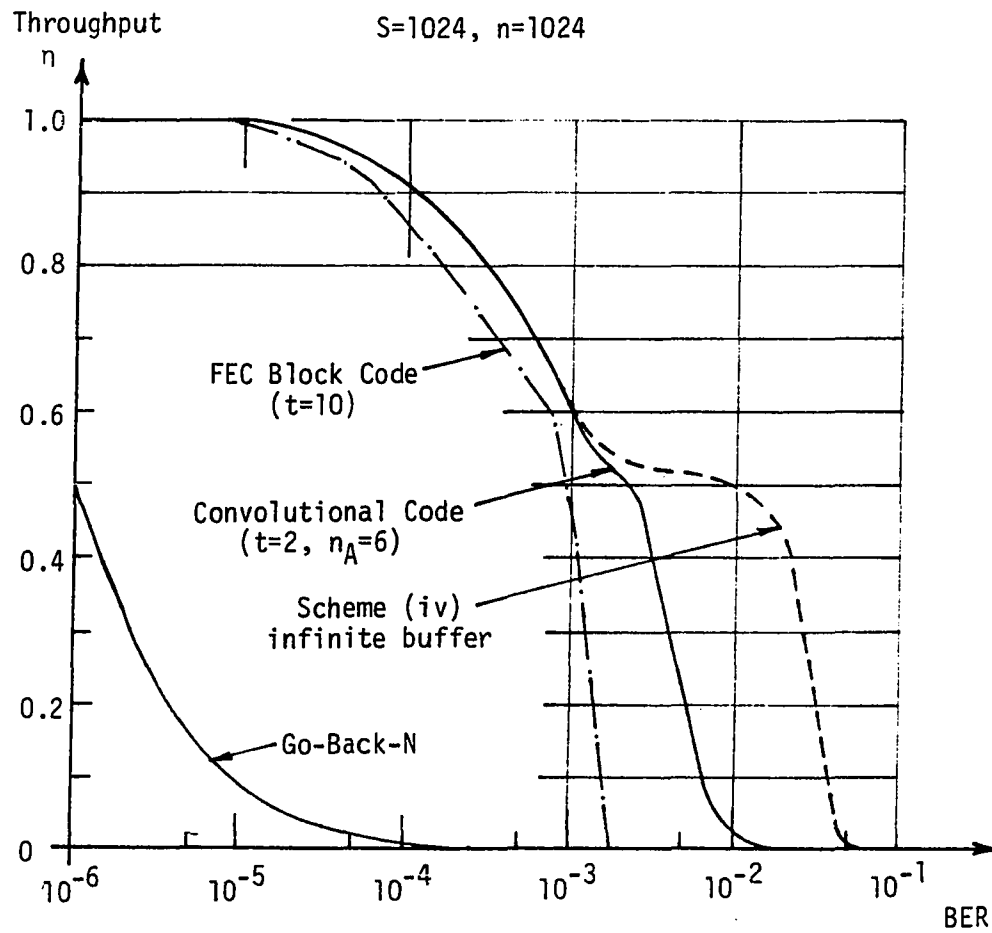
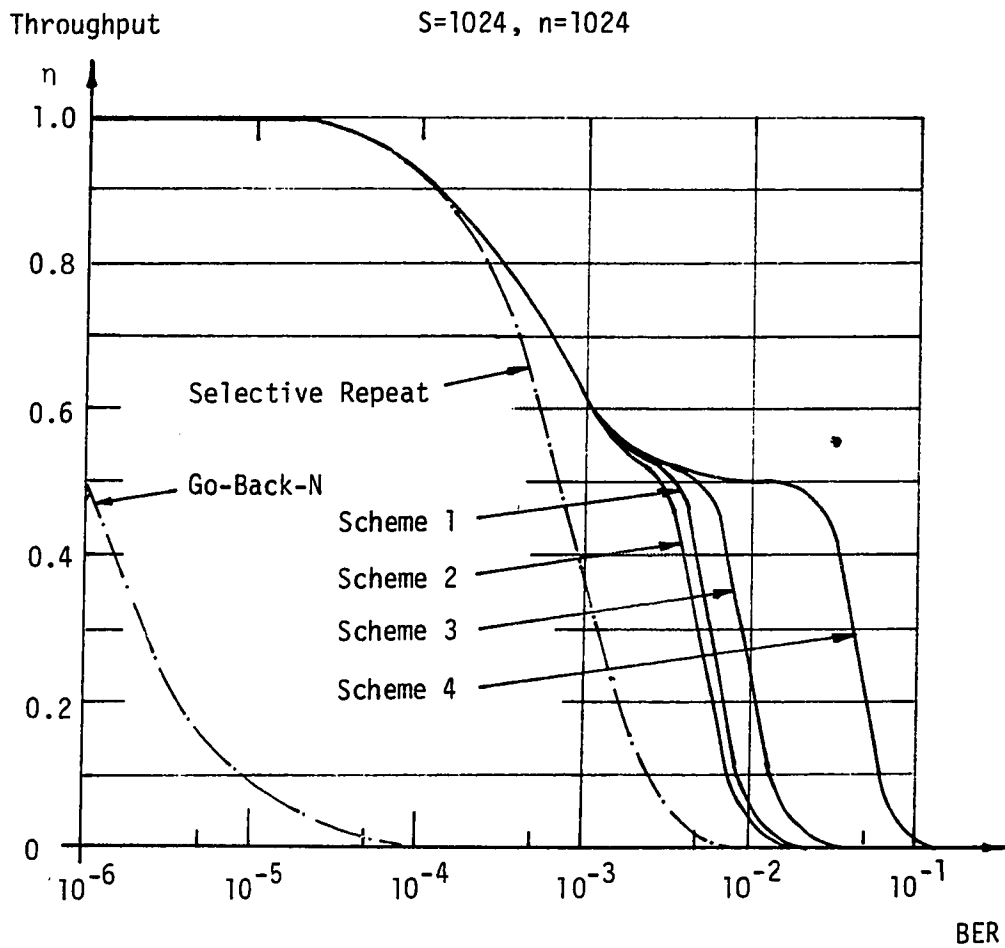


Figure 7.2

Comparison of Performance of Block Codes and Convolutional Codes
Lin and Yu Retransmission Protocol (Scheme iii)



- Scheme 1 - SR + ST protocol, alternate data/parity
 Scheme 2 - SR + GBN protocol, alternate data/parity
 Scheme 3 - Lin & Yu protocol, alternate data/parity,
 R_x buffer = S blocks
 Scheme 4 - Lin & Yu protocol, alternate data/parity,
 R_x buffer = ∞
 (Scheme 4 is equivalent to result for round trip delay S=1)

Figure 7.3

Type-II Hybrid ARQ Throughput for Schemes
 Using Convolutional Code With $d_{free}=7$, $n_A=10$

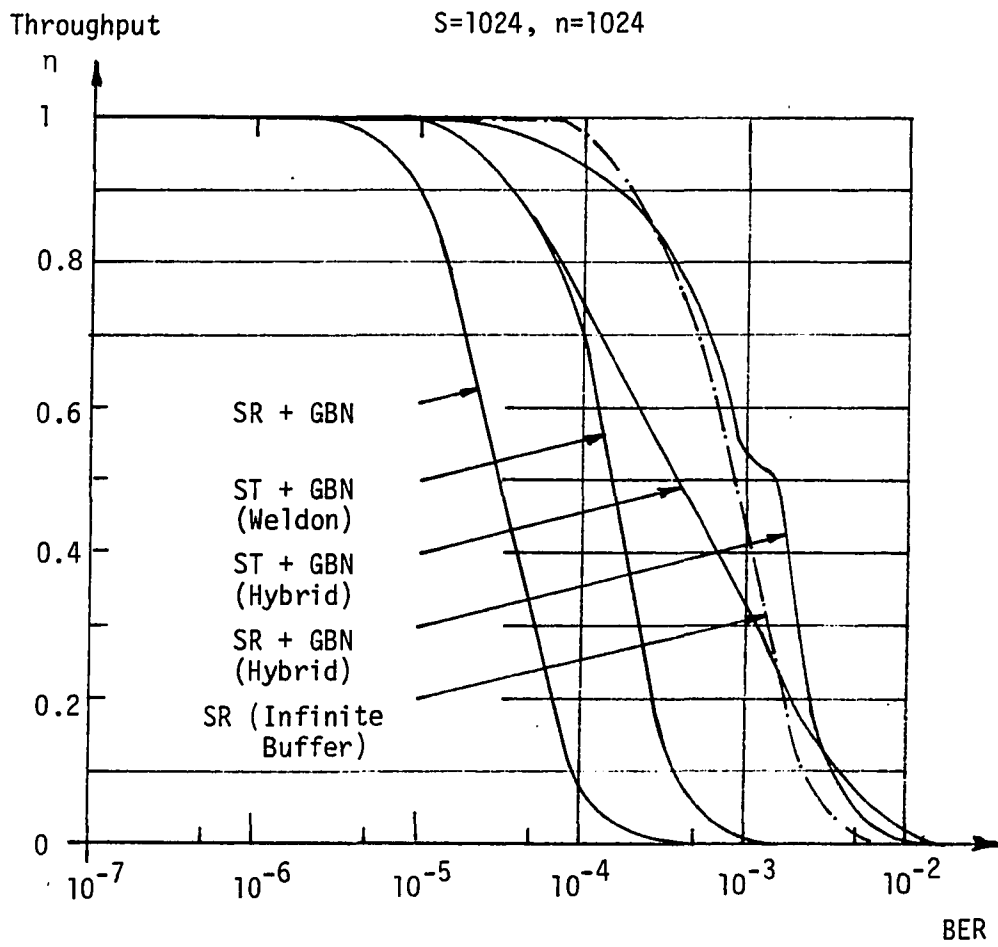


Figure 7.4

Performance of SR + GBN and ST + GBN Compared

choice for use in a Type-II hybrid scheme. In Figure 7.4 it can be seen that the ST+GBN hybrid scheme ($n_1=3$) has significantly worse throughput than the SR+GBN hybrid scheme except for $BER \leq 2 \times 10^{-3}$. In a properly designed hybrid scheme there should be a high probability that the data is recovered without errors after the second transmission. In that case the optimum choice of n_1 would be 1. For this value, the ST+GBN protocol becomes the same as the SR+GBN.

7.5 Reliability

In evaluating the performance of ARQ schemes, two parameters of primary importance are throughput and reliability. The previous sections have been concerned with the throughput performance of various ARQ and hybrid-ARQ procedures. In this section, consideration is given to reliability, that is the probability that blocks will be error-free when they are delivered to the user. If the event E is the event that the receiver accepts a received vector with undetected errors the reliability R of the system is given as

$$R = 1 - \Pr\{E\}$$

In conventional ARQ procedures such as the Go-Back-N a high rate block code C_0 is used for error detection. The transmitter encodes each message block of k bits into a code word $\{u, P_0(u)\}$ of n bits which is then transmitted. In practice, the $n-k$ parity bits $P_0(u)$ typically represent a small fraction of the total n bits in the block. For example, the CCITT Rec. X.25 packet switching link-level protocol recommends the use of 16 parity bits based on a cyclic code. The information vector u is typically 1024 bits in length. The receiver checks whether the

received block $\{\hat{u}, \hat{P}_0(u)\}$ has detectable errors by computing the syndrome. If this is zero the block is assumed error-free.

For the mixed-mode ARQ systems discussed in section 2, the same error detection procedures would be appropriate. Accordingly the same reliability performance can be expected. In the throughput analysis of section 2 it was assumed that the code C_0 was chosen in such a way that the probability of undetected errors is negligible.

Let E_i^d be the event that, after receiving the i^{th} transmission of a block, the receiver recovers the message u without errors.

Let E_i^e be the event that, after receiving the i^{th} transmission the receiver makes a decoding error by accepting the message containing undetected errors. Then it follows that for any ARQ scheme, the reliability is given by

$$R = 1 - \Pr\{E_1^e\} - \Pr\{E_1^d E_2^e\} - \Pr\{E_1^d E_2^d E_3^e\} - \Pr\{E_1^d E_2^d E_3^d E_4^e\} - \dots \quad (7.13)$$

For any ARQ system without forward-error correction over the BSC it follows that E_1^e, E_2^e, \dots are independent and that

$$E_i^d = p_d$$

$$E_i^e = p_e$$

The reliability therefore becomes

$$R = \frac{p_c}{p_c + p_e} \quad (7.14)$$

where $p_c = 1 - p_d - p_e$
 $= (1-p)^n$ with p the BER.

Also if the error detection code C_0 is chosen properly, then an upper bound on p_e is given [2] by

$$p_e \leq 2^{-(n-k)} [1 - (1-p)^k] \quad (7.15)$$

For Type-II hybrid ARQ schemes the above error detection procedure (syndrome check) can be used after the initial transmission of a given information block. For all the hybrid schemes discussed in section 3, whenever errors are detected in the initial transmission the second transmission is a "parity" block represented by $\{P_1(u), P_0(P_1(u))\}$. The k bits $P_1(u)$ are the parity bits generated by encoding the information bits u with a half-rate systematic code C_1 . Also the transmitter encodes the k -tuple $P_1(u)$ with the high rate code C_0 and produces $n-k$ parity bits $P_0(P_1(u))$.

On receipt of the n bits $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ the receiver first computes the syndrome based on C_0 to determine whether errors are detectable in the received parity block. This procedure is the same as that for error detection for the initial transmission. If errors are detected in the parity block the receiver uses the code C_1 to attempt to correct the errors in the $2k$ bits $\{\hat{u}, \hat{P}_1(u)\}$. If C_1 is a block code (e.g. a (2000,1000) shortened cyclic code) then the code can be used for error detection as well as error correction. As discussed in section 5, decoder complexity considerations would probably limit the choice of decoder error correction capability (t) to a maximum of about 10 errors. There exist half-rate codes of block size of the order of 2000 bits for which the minimum distance d_{\min} far exceeds that required for $t=10$ error correction. For example if C_1 is a (1000,500) shortened BCH code with $d_{\min}=111$ the code can be used to correct 10 errors and also to detect up to 100 errors. Such an error detection capability ensures that the Type-II hybrid ARQ schemes using block codes maintain very high reliability even for high channel error rate.

Lin and Yu [34] have shown that for this case the reliability is lower bounded by

$$R \geq \frac{p_c - \sigma}{p_c + p_e} \quad (7.16)$$

where for a $(2k, k)$ linear code C_1 which is capable of correcting up to t errors and simultaneously detecting up to d errors ($d > t$)

$$\sigma = \sum_{\ell > d}^{2k} \binom{2k}{\ell} p^\ell (1-p)^{2k-\ell} \quad (7.17)$$

If d is reasonably large, σ can be made very small and $R \approx \frac{p_c}{p_c + p_e}$.

7.6 Error Detection Procedures

In a Type-II hybrid scheme using convolutional codes it is necessary to use a different procedure for error detection to that used for schemes using block codes. In the block code hybrid scheme, not all the error correction capabilities of the code are utilized for error correction. This is to avoid the requirement for an excessively complex decoder. As a result the code can also be utilized for error detection.

For the hybrid schemes considered in section 7 it was assumed that the convolutional code C_1 was utilized for error correction alone. As expected the throughput performance results were superior to those for the block code case when channel BER is high.

Consider the alternate data-parity scheme (see section 3.3) using a convolutional code C_1 for error correction. If errors are detected in the first received data block $\{\hat{u}, \hat{P}_0(u)\}$ and in the next received parity retransmission $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ then an attempt is made to correct the errors in $\{\hat{u}, \hat{P}_1(u)\}$ using the half-rate convolutional code C_1 . Sliding block decoding is envisaged either by threshold decoding or by syndrome

look-up techniques. In order to determine whether errors still remain after decoding, the decoded message block \tilde{u} could be combined with the $n-k$ parity bits $\hat{P}_0(u)$ received in the first transmission. Then the syndrome can be computed using code C_0 to determine whether $\{\tilde{u}, \hat{P}_0(u)\}$ is a code word. If the syndrome is zero \tilde{u} is assumed error-free. If the syndrome is non-zero, a NAK is sent back to the transmitter to initiate a retransmission of the n -tuple $\{u, P_0(u)\}$. If errors are detected in this received block, error correction is again attempted using the $2k$ bits $\hat{P}_1(u)$ and \hat{u} received in the second and third transmissions respectively. The resultant information bits \tilde{u} are then combined with the $n-k$ parity bits $\hat{P}_0(u)$ received in the third transmission attempt. The syndrome check then proceeds as before and the process continues.

With this error detection procedure it may happen that even though there are no errors remaining in the decoded information bits \tilde{u} it may nevertheless be rejected. This would occur if a detectable pattern of errors occurred in the $n-k$ parity check bits $P_0(u)$. We next consider alternative error detection procedures. They are more complex but offer some advantages over the procedure described above. As before we assume the alternate data-parity hybrid scheme is used with a convolutional code C_1 for forward-error correction. After the initial transmission $\{u, P_0(u)\}$ of a given block, the high rate block code C_0 is used to compute the syndrome. If it is zero, the data is assumed error-free. If the syndrome is not zero, a "parity" retransmission $\{P_1(u), P_0(P_1(u))\}$ is sent. Again at the receiver the code C_0 is used to check whether the syndrome is zero. If it is zero, the received bits $\hat{P}_1(u)$ are assumed error-free and the data \tilde{u} is recovered from $\hat{P}_1(u)$ by an inversion procedure. If the syndrome is not zero, a convolutional decoder attempts to decode $\{\hat{u}, \hat{P}_1(u)\}$

to correct all the errors in the received data bits \hat{u} . To this point the procedure is identical to that described for the first error-detection procedure.

Now consider two alternative procedures for checking whether errors remain in the data \tilde{u} after decoding. For convenience we refer to them as error detection procedures 2 and 3 respectively.

Error detection procedure 2. In this procedure the decoder output \tilde{u} is checked by two steps. First the syndrome is computed for $\{\tilde{u}, \hat{P}_0(u)\}$ using the $n-k$ parity bits $\hat{P}_0(u)$ received in the first transmission. If this syndrome is not zero \tilde{u} cannot be considered error-free. Next \tilde{u} is fed into an encoding circuit identical to the encoder for the $(2,1)$ convolutional code C_1 as used at the transmitter. Let the input \tilde{u} produce the output parity bits $\tilde{P}_1(u)$. Now the block code C_0 is used to compute the syndrome for the n bits $\{\tilde{P}_1(u), \hat{P}_0(P_1(u))\}$. If the resultant syndrome is zero the data \tilde{u} can be considered error-free. The $n-k$ bits $\hat{P}_0(u)$ can be assumed to contain at least one error, the $n-k$ bits $\hat{P}_1(u)$ to be error-free. It is easy to see how this procedure can be used after any subsequent retransmissions if earlier transmissions fail.

Error detection procedure 3. In this alternative procedure, the convolutional code C_1 is used to correct errors in the received parity $\hat{P}_0(u)$ as well as the received data \hat{u} . If the initial transmission of a block is unsuccessful, the transmitter generates a "parity" retransmission block as follows. The original $k-m$ data bits u and the $n-k$ parity bits $P_0(u)$ are fed into the convolutional encoder followed by m encoder clearing zeros. The encoder output P_1 will consist of n bits. Let $I(X)$ represent in polynomial form the "data" transmission consisting of the $k-m$ data bits, the $n-k$ parity bits generated by C_0 and m zeros. Let $P_1(X)$

represent the n -bit "parity" transmission generated by the $(2,1)$ code C_1 with generators $g^{(1)}(X)=1$ and $g^{(2)}(X)$. Then

$$P_1(X) = I(X)g^{(2)}(X) .$$

On receipt of the "parity" transmission $\hat{P}_1(X)$ the receiver first divides $\hat{P}_1(X)$ by $g^{(2)}(X)$. Let $\tilde{I}_1(X)$ and $\tilde{R}_1(X)$ be the quotient and remainder respectively. The receiver first checks whether $\tilde{R}_1(X)$ is zero. If not a NAK is sent. If $\tilde{R}_1(X)=0$ the receiver uses code C_0 for error detection in $\tilde{I}_1(X)$. If no errors have occurred in transmission, $\tilde{I}_1(X)$ will consist of the original k - m data bits and n - k parity bits generated by C_0 . The receiver computes the syndrome of $\tilde{I}_1(X)$ using code C_0 . If the syndrome is zero the receiver recovers the data estimate \tilde{u} and sends an ACK. Otherwise a NAK is sent requesting another transmission.

It is readily observed that, of the three error detection schemes discussed, the third is superior and likely to provide a slightly higher block throughput. The probability of recovering the data \tilde{u} without errors after second and subsequent transmissions will be marginally higher than for the other two error detection schemes.

In chapter 4 expressions for throughput for hybrid schemes were derived (see for example Equations (4.2) and (4.5)). These assumed that there was a negligible probability that the error-detection procedures would fail to detect errors. The throughput expressions were functions of conditional probabilities q_1 and q_2 which depended on the forward-error correction procedures to be used. In chapter 7, methods for computing q_1 and q_2 were given for convolutional and block codes. For hybrid ARQ using convolutional codes, q_1 and q_2 were expressed (Equations (7.9) and (7.11)) in terms of the block error probability P_B^1 after

decoding the $2k$ bits $\{\hat{u}, \hat{P}_1(u)\}$. In computing values for q_1 and q_2 it is a simple matter to take into account the slight reduction in block throughput resulting from use of error detection procedure 3. In evaluating the block error rate P'_B a block length of $N_B=2n$ bits should be used, not $N_B=2k$ as discussed in section 7.3. Examination of Table 5.4 and use of Equation (5.24) shows that for $n \gg n-k$ the reduction in throughput will be very small. For example if $n=1024$ and $n-k=16$ we can compare values of P'_B for length $2n=2048$ and length $2k=2016$ bits respectively. To illustrate, we compute P'_B values for $BER=10^{-3}$ and using a convolutional code capable of correcting up to 2 errors in a constraint length of 10 bits.

$$\begin{array}{ll} \text{For length 2048 bits} & P'_B = 0.7263 \times 10^{-4} \\ \text{For length 2016 bits} & P'_B = 0.7163 \times 10^{-4} \end{array}$$

It is clear that there will not be significant differences in throughput values for these two cases.

The reliability of each of these error-detection schemes will be very high providing the code C_0 is properly chosen. This is because conventional syndrome checking is used in each case as in conventional ARQ systems. However it appears difficult to analyze the reliability performance for these three error-detection procedures. In order to do so the joint probabilities in Equation (7.13) would have to be determined and this appears excessively complex. When the error-correction decoder makes an error, the resultant error pattern will no longer be randomly distributed. As discussed in section 6.3, if feedback decoding is used, error propagation may occur. This means that Equation (7.15) can no longer be used to bound the syndrome check procedure for error detection.

8. SUMMARY AND CONCLUSIONS

8.1 Summary of Research

The key results of this dissertation can be summarized as follows:

ARQ Retransmission Schemes: (Chapter 2)

- (i) A class of mixed-mode ARQ schemes has been found which are selective-repeat types. They operate with finite receiver buffer without overflow.
- (ii) It is useful to conceptualize these ARQ schemes as operating in one of two modes. The primary mode is selective-repeat (SR). The secondary mode can be either a "stutter" repetition of a faculty block (ST) or a Go-Back-N procedure (GBN). The purpose of the secondary mode is to prevent receiver buffer overflow.
- (iii) The mixed-mode schemes are amenable to exact throughput analysis in terms of block length (n), round-trip delay (S) and bit-error rate (p) for a BSC. The analysis results are summarized in Table 2.1 for ARQ schemes labelled SR+ST scheme 1, SR+ST scheme 2 and SR+GBN respectively. Figures 2.9 and 2.10 show typical values of throughput versus bit-error rate for selected block sizes and round-trip delay.
- (iv) The results show that for best ARQ performance, at least the first retransmission of a block following an error should be in the selective-repeat mode. Providing this is done, markedly superior performance is achieved over conventional Go-Back-N schemes for channels with high bit rates and large round-trip delays.

- (v) For related SR+ST and SR+GBN schemes (that is, those with the same number v of SR retransmissions), the throughput is very nearly the same. This indicates that the choice of the secondary mode does not have a significant bearing on the throughput but does have a bearing on complexity.
- (vi) The results also illustrate the dependence of throughput on receiver buffer size. Table 2.1 shows how the throughput and receiver buffer requirements vary with the number (v) of selective-repeat retransmissions allowed for any given block.
- (vii) The mixed-mode schemes offer considerably better throughput than most ARQ schemes previously proposed (refer Figure 2.11). A selective-repeat scheme by Yu and Lin provides superior throughput for some bit-error rates (refer Figure 2.12) but has not been found amenable to exact analysis.

Type-II Hybrid ARQ: (Chapters 3 and 4)

- (viii) Hybrid ARQ schemes are available in which retransmission of parity bits is used rather than the original data. The parity block can be generated using half-rate block or convolutional codes. By this means, forward-error detection is made available at the receiver if required. In this sense the Type-II hybrid scheme is "adaptive".
- (ix) In order to completely specify any Type-II hybrid ARQ scheme it is important to separately describe two aspects, namely
 - the retransmission protocol (refer Chapter 2)
 - and • the error mechanism/correction procedures.

The separate consideration of these two aspects of any ARQ scheme leads to a unified throughput analysis procedure applicable to a wide variety of different schemes.

- (x) State diagrams are of assistance in representing the error mechanism/correction procedures. Figures 3.1-3.5 illustrate different schemes. Such representation is applicable to either convolutional or block FEC codes.
- (xi) Expressions for throughput have been derived for a selection of different Type-II hybrid ARQ schemes (see Equations (4.2), (4.5), (4.6) and (4.10)). They are presented as functions of block error rate (P_B), round-trip delay (S) and conditional probabilities q_1 and q_2 . The latter two parameters depend on the nature of the FEC code and decoding procedures to be used. They are a function of the block error rate achievable after decoding.

Block-Error Rates for Convolutional Codes: (Chapter 5)

- (xii) Modestly powerful half-rate convolutional codes with sliding block decoders are proposed for forward-error correction in the Type-II hybrid ARQ schemes. The decoder may be either a threshold decoder or table-lookup decoder with feedback. This is suggested as a reasonable compromise between complexity and performance.
- (xiii) A "trellis" algorithm is proposed as a means for computing the block error rate (P_B') after decoding a convolutional code. Table 5.4 and Figure 5.9 present typical values for selected codes. A method is outlined (Equation (5.24)) for utilizing the table to obtain P_B' values for other block sizes.
- (xiv) An alternative approach to finding block error probability is outlined. It is based on upper bounds on P_B' found using combinatorial techniques. A method is presented for enumerating the number of correctable patterns $N(j)$ in a block of N_B bits (information plus parity) for any single or double error correcting

convolutional code. The decoder effective constraint length must be specified. The results are presented in Equations (5.4) to (5.14). The P_b values computed using these results are consistent with those using the trellis algorithm. The combinatorial approach can be used when code constraint length or error correcting ability are so large that the computing algorithm becomes impractical.

- (xv) Comparisons are made between the block error rate performance of convolutional and block codes. Results are presented (Figure 5.10) for half-rate codes with block size of 2000 bits. Decoders of approximately comparable complexity are assumed. A 10-error correcting decoder is assumed for the block code, a sliding-block decoder for the convolutional codes. (The former does not utilize the whole error-correcting capability of the block code.) The results show the advantage of using convolutional codes under high bit-error rate conditions.

The Design of Related Codes with Rates 1/2 and 1/3: (Chapter 6)

- (xvi) Codes are investigated for use in a parity transmission scheme described in section 4.5. After the first retransmission, a half-rate code is available at the receiver for error correction. After the second retransmission a one-third rate code word is to be available for more powerful error correction. Common decoder circuitry is envisaged. The properties required of the "related" rate 1/2 and rate 1/3 codes are specified.
- (xvii) Optimum orthogonalizable rate 1/3 codes are well-known. An exhaustive trial-and-error method is outlined for constructing related rate 1/2 codes. New (2,1) codes were found as pairs to

all the optimum orthogonalizable (3,1) codes for error-correcting capability $t=1$ to 8. The new orthogonalizable codes are shown in Table 6.1. Also shown are the rules for forming parity check sums for threshold decoding.

- (xviii) The circuit schematic of a typical threshold decoder capable of decoding either the (2,1) code or the (3,1) code in a related pair is described. Alternatively a read-only-memory (ROM) circuit could be used for table-lookup decoding. In either case, the decoder is relatively simple.

Performance of Type-II Hybrid Schemes: (Chapter 7)

- (xix) Methods for calculating the conditional probabilities q_1 and q_2 (refer xi above) are presented for the alternate data-parity schemes using convolutional codes (Equations (7.4) and (7.5)). Values for block codes were previously available. For one of the hybrid schemes (the rate 1/2, 1/3 scheme) no method has been obtained for calculating q_2 .
- (xx) Throughput is plotted versus bit-error rate for selected combinations of ARQ retransmission protocols, hybrid error-correction schemes and forward-error-correcting codes (Figs. 7.1-7.4). Values of block sizes of 1024 and round-trip delay of 1024 blocks are used. The results show that Type-II hybrid schemes can maintain in excess of 50% throughput even when bit error rates exceed 3×10^{-3} (for blocks of 1024 bits this represents a block error rate of approximately 0.95). Throughput values are computed for hybrid ARQ schemes with infinite buffer. These results indicate that the use of ARQ protocols with more than one selective-repeat retransmission might enable hybrid ARQ systems

to have acceptable throughput up to bit-error rates as high as 10^{-2} .

- (xxi) The convolutional code schemes provide generally better performance than the block code schemes. One obvious reason for this is that the whole error-correcting properties of the convolutional codes are used. Decoder complexity prevents utilization of more than a small part of the potential error correction capability of the block codes.
- (xxii) In Chapter 7 methods for ensuring high reliability are discussed for hybrid schemes using convolutional codes. These use the parity bits generated by a high-rate block code used for error detection only. No procedure for analysis of reliability has been obtained for the hybrid schemes using convolutional codes.
- (xxiii) Type-II hybrid schemes can only be effective if a selective-repeat type of retransmission protocol is used.
- (xxiv) For throughput values between 0.5 and 1, the throughput versus bit error rate curves of all hybrid schemes studied are relatively insensitive of the forward-error-correcting ability of the codes. Therefore in any practical scheme, more powerful codes can only be of benefit for throughput values less than 0.5. More efficient decoding schemes such as maximum likelihood or probabilistic schemes may not justify the significant increase in complexity.

8.2 Conclusions

ARQ schemes seem likely to remain the most popular approach to error control for some time to come. This is because of their high reliability.

The use of digital transmission is rapidly increasing, particularly in circuit-switched and packet-switched networks. As these networks expand, it is to be expected that increasing utilization will be made of satellite and long terrestrial circuits with significant transmission delays. In these circumstances it is to be hoped that this research will contribute towards future design of error-control procedures.

The study of ARQ retransmission procedures in Chapter 2 demonstrates that the throughput performance of currently used ARQ protocols may be considerably improved for circuits with delay and high bit-error rates. The provision of receiver buffers for storing error-free blocks seems desirable. This buffer size should be chosen in accordance with the requirements of a selective-repeat ARQ scheme such as one of the "mixed-mode" protocols. The results of Chapter 2 indicate the anticipated trade-offs between performance and complexity.

Further benefits in improved throughput can be achieved by an appropriately designed hybrid ARQ scheme. This research has shown that convolutional codes can provide an effective means of forward-error-control in an ARQ system. We have directed our attention to codes of modest error correcting ability for which the encoder and decoder circuits are not excessively complex. The results indicate that considerable benefits are possible if such codes are combined with a Type-II hybrid ARQ scheme using parity-retransmission. Type-II hybrid schemes can only be effective if a selective-repeat type of retransmission protocol is used.

Of the mixed-mode schemes studied the SR+GBN protocol requires the least receiver buffer provision, yet provides good throughput efficiency performance. Error-correction using sliding-block decoding based on read-only-memory (ROM) syndrome look-up procedures also appears a reasonable

choice from the decoding procedures discussed. Such a decoder would avoid restricting the choice of codes to orthogonal or orthogonalizable codes. Current developments in large-scale integration of multiple functions on a microcircuit suggest that the economic viability of hybrid ARQ schemes may be well worth further investigation.

Certain simplifying assumptions have been made in this research to facilitate the analytical procedures. Data has been assumed transmitted in equal length blocks. Errors have been assumed randomly distributed. Feedback channels were considered error-free. No consideration has been given to the problems of timeouts, lost blocks or failure of frame synchronization. There are therefore a number of these practical matters that would need to be considered before a working system design could be considered complete. Nevertheless the results of this dissertation seem to provide a useful information base for the ARQ system designers in the future.

Appendix A

A NOTE ON THE DERIVATION OF EQUATION 2.7

Consider the number of combinations of the $2S$ distinct ACK/NAK indications described in Section 2.3.3. We wish to find the probability that at least one pair of NAK's will occur spaced $S + 1$ block intervals apart.

Let $N(k,j)$ denote the number of events such that

- There are $k + 2$ NAK's and $2S - k - 2$ ACK's
- There are $j + 1$ pairs of NAK's such that for each pair the NAK's are spaced $S + 1$ block intervals apart. For convenience we denote such pairs 'double-NAK pairs'.

Case (a). Let $j = 0$. $N(k,0)$ is the number of ways of arranging $k+2$ NAK's and $2S - k - 2$ ACK's so that there is at least one double-NAK pair. It follows that

$$N(k,0) = \binom{S}{1} \binom{2S - 2}{2S - 2 - k} \quad k = 0,1,2, \dots, (2S - 2).$$

Case (b). Let $j = 1$. The number of ways of arranging $k+2$ NAK's so that there are at least two double-NAK pairs is

$$N(k,1) = \binom{S}{2} \binom{2S - 4}{2S - 2 - k}$$

In general,

$$N(k,j) = \binom{S}{j+1} \binom{2S - 2 - 2j}{2S - 2 - k}$$

where the non-zero k, j terms can be illustrated in the following array

$-j =$	0	1	2	3	4 ...	$S-1$
k						
0	$N(0,0)$	0	0	...		
1	$N(1,0)$	0	0	...		
2	$N(2,0)$	$N(2,1)$	0	...		
3	$N(3,0)$	$N(3,1)$	0	...		
4	$N(4,0)$	$N(4,1)$	$N(4,2)$	0	...	
5	$N(5,0)$	$N(5,1)$	$N(5,2)$	0	...	
6	$N(6,0)$	$N(6,1)$	$N(6,2)$	$N(6,3)$	0 ...	
\vdots	\vdots					
$2S-2$	$N(2S-2,0)$	$N(2S-2, S-1)$

The event $\{k+2$ NAK's and $2S - k - 2$ ACK's $\}$ occurs with probability $P_B^{k+2} (1-P_B)^{2S-2-k}$ with P_B defined in Section 3 above.

Following the principle of Inclusion and Exclusion the number of such events is

$$N(k) = \sum_{j=0}^{S-1} (-1)^j \binom{S}{j+1} \binom{2S-2-2j}{2S-2-k}$$

Note that $N(k,0)$ includes counts of 1,2, ... double-NAK pairs and $N(k,1)$ must be subtracted. Then the events denoted $N(k,2)$ have been subtracted twice and must be restored and so on. Finally by examination of the range of non-zero values in the array shown above we obtain

$$P_{NN} = \sum_{k=2j}^{2S-2} \sum_{j=0}^{S-1} (-1)^j \binom{S}{j+1} \binom{2S-2-2j}{2S-2-k} P_B^{2+k} (1-P_B)^{2S-2-k}$$

Appendix B

TRELLIS ALGORITHM COMPUTER PRINTOUT

```

00100 PROGRAM ABMM
00110 INTEGER W
00120 REAL AP (40,40), BP(40,40)
00130C THIS PROG COMPUTES BLOCK ERROR PROB BOUND FOR T=2
00140 PRINT 150
00150 150 FORMAT ('INPUT NB,NA,BER')
00160 READ *,NB,NA,BER
00170 W=NA
00180 PRINT 190,NA,BER
00190 190 FORMAT ('CONSTRAINT LENGTH=',I4,'          BER=',G10.5)
00210 PRINT 220
00220 220 FORMAT ('BLOCK LENGTH          BLOCK ERROR PROB')
00230C
00240C INITIALIZE STATE PROBSFOR BL LENGTH=W
00250C
00252 DO 259 J=1,W+1
00254 DO 259 K=1,W+1 *
00255 BP(J,K)=0
00256 259 AP(J,K)=0
00260 AP(W+1,W+1)=(1-BER)**W
00270 DO 260 I=1,W
00280 AP(I,W+1)=BER*((1-BER)**(W-1))
00290 AP(W+1,I)=BER*((1-BER)**(W-1))
00300 260 CONTINUE
00310 DO 300 J=1,W
00320 DO 300 K=1,W
00330 300 AP(J,K)=(BER**2)*((1-BER)**(W-2))
00340C
00350C COMPUTE NEXT STATE PROBS
00360C
00370 DO 400 L=W+1,NB
00380 BP(W+1,W+1)=(1-BER)*(AP(W+1,W+1)+AP(1,W+1))
00390 DO 310 J=2,W
00400 310 BP(J-1,W+1)=(1-BER)*(AP(J,W+1)+AP(1,J))
00410 BP(W,W+1)=BER*(AP(W+1,W+1)+AP(1,W+1))
00420 DO 360 KI=1,W-2
00430 DO 350 K=KI+1,W-1
00440 350 BP(KI,K)=(1-BER)*(AP(KI+1,K+1))
00450 360 CONTINUE
00460 DO 370 J=1,W-1
00470 370 BP(J,W)=BER*(AP(J+1,W+1)+AP(1,J+1))
00480 SUMPC=0
00510 DO 390 J=1,W+1
00520 DO 380 K=1,W+1
00530 AP(J,K)=BP(J,K)
00540 380 SUMPC=SUMPC+AP(J,K)
00550 390 CONTINUE

```

```
00555  SUMPE=1-SUMPC
00560  400 CONTINUE
00565  PRINT 450,L-1,SUMPE
00570  450 FORMAT(I6,T20,G20.10)
00580  STOP
00590  END
```

Appendix C

COMBINATORIAL EVALUATION OF CORRECTABLE ERROR PATTERNS

EQUATIONS 5.10 AND 5.11

Proof for $j=3$. Figure C.1 illustrates the method used to enumerate the number of correctable patterns of 3 errors in N_B bits.

In Figure C.1 the constraint length (n_A) is shown as 6 bits for illustration. Correctable error patterns are enumerated commencing with the 3 errors (represented by X) located in the left-most positions such that no more than 2 errors occur in the first n_A bits. It is easy to see from Figure C.1 that the number $N_1(3)$ of correctable patterns with an error in the first position of the N_B -bit block is given by

$$N_1(3) = \binom{n_A-2}{1} \binom{N_B-n_A}{1} + \binom{N_B-n_A+1}{2}$$

in which the last term counts those patterns where the first error location does not restrict permissible locations of the last two errors i.e. all possible combinations of 2 errors in (N_B-n_A+1) bits are counted.

Let $N(j,x)$ represent the number of correctable patterns of j errors in a block x bits long (subject, of course, to the constraint that no more than $t=2$ errors should occur in any n_A bit constraint length).

Then $N(3,N_B)$ is the total number of correctable 3-error patterns in N_B bits. In Figure C.1 the correctable patterns indicated below the line AA' are given by $N(3,N_B-1)$, the number of correctable patterns in the last N_B-1 bits. Clearly $N(3,N_B)$ can be expressed in terms of the difference equation

$$N(3,N_B) = \binom{n_A-2}{1} \binom{N_B-n_A}{1} + \binom{N_B-n_A+1}{2} + N(3,N_B-1) \quad (C.1)$$

Example. Consider the case where $S = 4$ illustrated in Figure 2.6. There are 8 ACK/NAK signals to be considered, the 4 before block i is first transmitted and the 4 following. We begin with $N(0,0)$, that is the number of ways of arranging 6 ACK's with 2 NAK's spaced 5 intervals apart to form a double-NAK pair. There are 4 ways this can occur. Next we find $N(1,0)$, that is the number of ways of arranging 3 NAK's and 5 ACK's so that there is one double-NAK pair. There are 24 such arrangements. Likewise we can find $N(2,0)=60$ and so on as illustrated in the following table.

N(k,j) values for S = 4					
	j = 0	1	2	3	N(k)
k					
0	4				4
1	24				24
2	60	6			54
3	80	24			56
4	60	36	4		28
5	24	24	8		8
6	4	6	4	1	1

Now consider the $N(2,0)$ value, say, counting possible arrangements of 4 NAK's and 4 ACK's. The total number of arrangements $N(2,0)$ of double-NAK pairs is 60. But not all of these events are mutually exclusive since $N(2,0)$ includes 6 cases where the 4 NAK's form a pattern of two concurrent double-NAK pairs. These $N(2,1)$ cases must be subtracted from $N(2,0)$ to find the number of mutually exclusive arrangements for $k = 2$. The remainder of the $N(k,j)$ array should be self explanatory.

A convenient method to solve this equation is to note that because the enumeration proceeds as illustrated in Figure C.1 $N(3, N_B)$ will be of the form

$$\begin{aligned}
 N(3, N_B) &= \binom{n_A-2}{1} \binom{N_B-n_A}{1} + \binom{N_B-n_A+1}{2} \\
 &+ \binom{n_A-2}{1} \binom{N_B-n_A-1}{1} + \binom{N_B-n_A}{2} \\
 &+ \dots \\
 &+ \binom{n_A-2}{1} \binom{1}{1} + \binom{2}{2} \\
 &= \sum_{i=1}^{N_B-n_A} \left\{ \binom{n_A-2}{1} \binom{i}{1} + \binom{i+1}{2} \right\}
 \end{aligned}$$

Hence

$$N(3, N_B) = \binom{n_A-2}{1} \binom{N_B-n_A+1}{2} + \binom{N_B-n_A+2}{3}$$

as given in Theorem 3.2, Equation (5.10).

Now, combinatorial identities expressed in terms of binomial coefficients have notoriously a myriad of equivalent solutions. Other convenient equivalent forms for $N(3, N_B)$ based on combinatorial identities including the Vandermonde convolutional formula (Riordan [41])

$$\binom{n}{m} = \sum_{i=0}^m \binom{n-p}{m-i} \binom{p}{i} \quad (C.2)$$

or its variant

$$\binom{n-p}{m} = \sum_k (-1)^k \binom{n}{m-k} \binom{p+k-1}{k} \quad (C.3)$$

are as follows

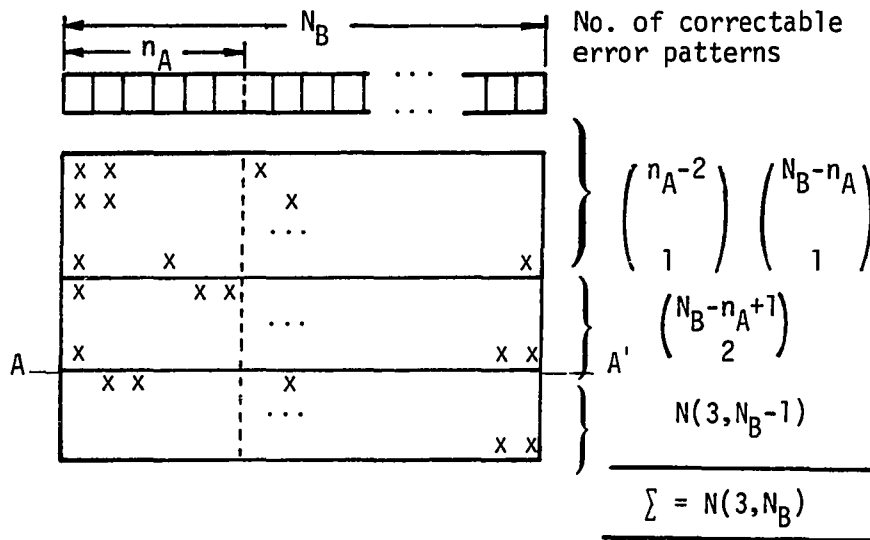


Figure C.1

Correctable Error Patterns $N(3, N_B)$
of 3 Errors in N_B Bits $t=2$

$$\begin{aligned}
N(3, N_B) &= \binom{N_B - n_A + 1}{3} + \binom{n_A - 1}{1} \binom{N_B - n_A + 1}{2} \\
&= \binom{N_B - n_A + 2}{3} + \binom{n_A - 2}{1} \sum_{i=0}^{\infty} (-1)^i \binom{N_B - n_A + 2}{2-i} \\
&= \sum_{i=1}^{\min(n_A - 1, 4)} \binom{i}{1} \binom{n_A - 2}{i-1} \binom{N_B - 2n_A + 4}{4-i} \tag{C.4}
\end{aligned}$$

Proof for $j=4$. Let $N(4, N_B)$ denote the number of correctable patterns of 4 errors in a block of length N_B . Figure C.2 illustrates the enumeration method used to obtain a difference equation for $N(4, N_B)$.

Hence the difference equation is

$$\begin{aligned}
N(4, N_B) &= \sum_{i=0}^{n_A - 3} \binom{N_B - n_A - i}{2} + \sum_{i=1}^{n_A - 3} \binom{i}{1} \binom{N_B - n_A - 1 - i}{1} \\
&\quad + N(3, N_B - n_A + 1) + N(4, N_B - 1) \tag{C.5}
\end{aligned}$$

But from Equation (5.10)

$$N(3, N_B - n_A + 1) = \binom{n_A - 2}{1} \binom{N_B - 2n_A + 2}{2} + \binom{N_B - 2n_A + 3}{3}$$

and since

$$\sum_{i=0}^{n_A - 3} \binom{N_B - n_A - i}{2} = \binom{N_B - n_A + 1}{3} - \binom{N_B - 2n_A + 3}{3}$$

using

$$\sum_{i=0}^A \binom{i}{1} \binom{B-i}{1} = \binom{B+1}{3} - \binom{B-A+1}{3} - \binom{A}{1} \binom{B-A}{2}$$

we obtain for the second term in Equation (C.5)

$$\begin{aligned}
\sum_{i=1}^{n_A - 3} \binom{i}{1} \binom{N_B - n_A - 1 - i}{1} &= \binom{N_B - n_A}{3} - \binom{N_B - 2n_A + 3}{3} \\
&\quad - \binom{n_A - 3}{1} \binom{N_B - 2n_A + 2}{2}
\end{aligned}$$

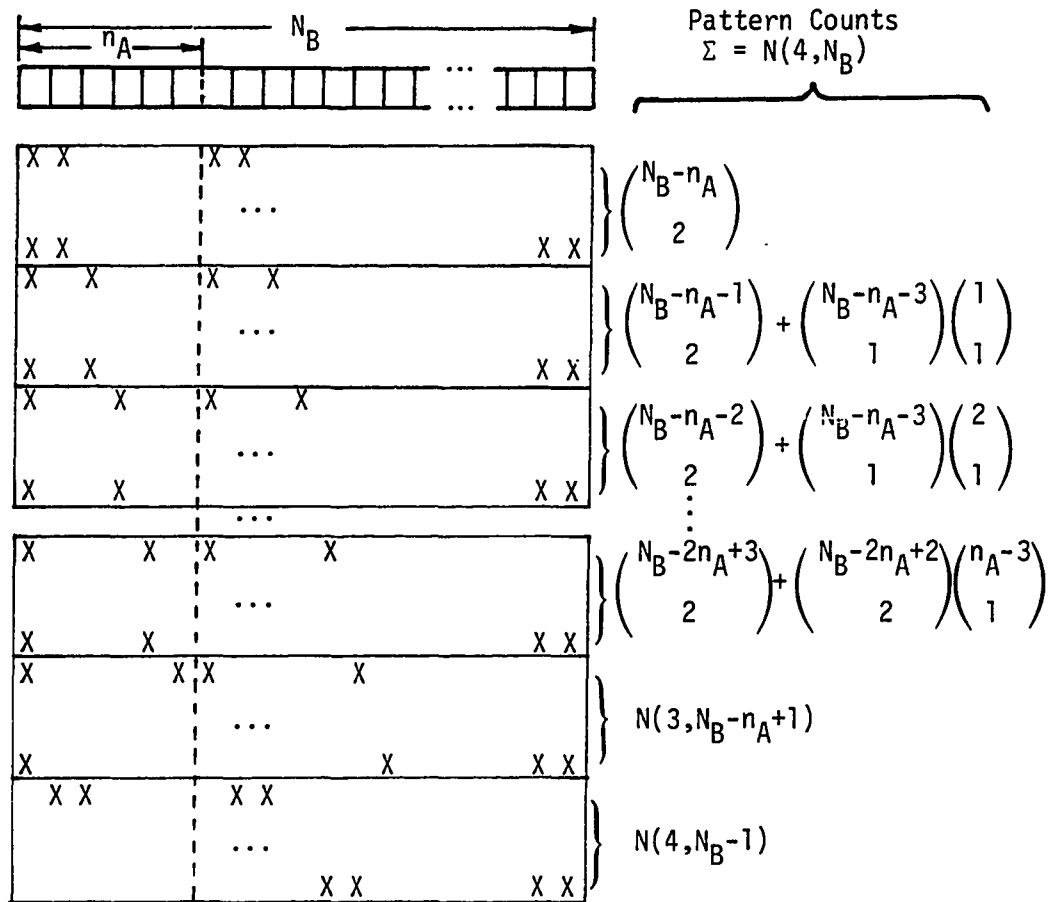


Figure C.2

No. of Correctable Error Patterns $N(4, N_B)$
of 4 Errors in N_B Bits $t=2$

Gathering terms, Equation (C.5) becomes

$$\begin{aligned}
 N(4, N_B) &= 2 \binom{N_B - n_A}{3} + \binom{N_B - n_A}{2} - 2 \binom{N_B - 2n_A + 3}{3} + \binom{N_B - 2n_A + 2}{2} \\
 &+ \binom{N_B - 2n_A + 3}{2} + N(4, N_B - 1) \\
 &= \binom{N_B - n_A + 1}{3} + \binom{N_B - n_A}{3} - \binom{N_B - 2n_A + 2}{3} + N(4, N_B - 1) \quad (C.6)
 \end{aligned}$$

Hence, from the boundary conditions as discussed for $j=3$ we obtain the solution

$$N(4, N_B) = \binom{N_B - n_A + 2}{4} + \binom{N_B - n_A + 1}{4} - \binom{N_B - 2n_A + 3}{4}$$

as given in Theorem 5.2, Equation (5.11).

Other useful variants for $N(4, N_B)$ are obtained using Equations (C.2) and (C.3). For example

$$\begin{aligned}
 N(4, N_B) &= \binom{N_B - 2n_A + 3}{4} + \sum_{i=1}^{\min(n_A - 1, 4)} \left\{ \binom{n_A - 1}{i} \right. \\
 &\left. + \binom{n_A - 2}{i} \right\} \binom{N_B - 2n_A + 3}{4 - i}
 \end{aligned}$$

Proof for $j=5, 6, 7$. The procedure to obtain $N(j, N_B)$ for values of $j=5, 6, 7$ is as set out for the previous cases $j=3, 4$ except for a rapid increase in complexity.

Appendix D

In this appendix, methods for evaluating the conditional probabilities q_1 and q_2 (defined in Section 6.2) are given. We have from equation (6.5) reproduced here for convenience,

$$q_1 = 1 - P_B + \frac{P_r \{D_1 D_2 F_2\}}{P_B}$$

where the events D_1 , D_2 and F_2 are defined in Section 6.2. To compute the joint probability $P_r\{D_1 D_2 F_2\}$, consider Table D.1 which illustrates all 16 possible combinations of the presence or absence of detected errors in \hat{u} , $\hat{P}_0(u)$, $\hat{P}_1(u)$ and $\hat{P}_0(P_1(u))$ respectively. In the table, e represents a pattern of at least 1 error detected and 0 represents no errors detected.

The error patterns to be included in the joint event $\{D_1 D_2 F_2\}$ are indicated in Table D.1. As indicated in the last column the joint probability can be computed using

$$P_r\{D_1 D_2 F_2\} = \phi_1 + \phi_2 + \phi_3 + \phi_4$$

where $\phi_1 = P_r\{\text{at least one error in } \hat{u}, \text{ at least one error in } \hat{P}_1(u) \text{ and the error pattern in } \hat{u} \text{ \& } \hat{P}_1(u) \text{ is correctable by the decoder using } C_1\}$

$\phi_2 = P_r\{\text{No errors in } \hat{u}, \text{ at least 1 error in } \hat{P}_0(u), \text{ at least 1 error in } \hat{P}_1(u) \text{ and the error pattern in } \hat{u}, \hat{P}_1(u) \text{ is correctable using } C_1\}$

$\phi_3 = P_r\{\text{at least 1 error in } \hat{u}, \text{ no error in } \hat{P}_1(u), \text{ at least 1 error in } \hat{P}_0(P_1(u)) \text{ and the error pattern in } \hat{u}, \hat{P}_1(u) \text{ is correctable using } C_1\}$

Table D.1
 Error Patterns to be Considered
 in Calculation of $P_r\{D_1 D_2 F_2\}$

Pattern No.	u	$P_0(u)$	$P_1(u)$	$P_0(P_1(u))$	Pattern Counted?	Probability term
1	0	0	0	0	No	
2	0	0	0	e	No	
3	0	0	e	0	No	
4	0	0	e	e	No	
5	0	e	0	0	No	
6	0	e	0	e	Yes	ϕ_4
7	0	e	e	0	Yes	ϕ_2
8	0	e	e	e	Yes	ϕ_2
9	e	0	0	0	No	
10	e	0	0	e	Yes	ϕ_3
11	e	0	e	0	Yes	ϕ_1
12	e	0	e	e	Yes	ϕ_1
13	e	e	0	0	No	
14	e	e	0	e	Yes	ϕ_3
15	e	e	e	0	Yes	ϕ_1
16	e	e	e	e	Yes	ϕ_1

$$\phi_4 = P_r \{ \text{no error in } \hat{u}, \text{ at least 1 error in } \hat{P}_0(u), \text{ no error in } \hat{P}_1(u) \text{ and at least 1 error in } \hat{P}_0(P_1(u)) \}$$

We assume that the error detection code C_0 is properly chosen so that the probability of an undetected error is negligible.

Consider now a binary symmetric channel with independent bit errors with bit-error-rate p . To evaluate ϕ_1 we note that the number of errors in the combined $2k$ -bit received vectors \hat{u} & $\hat{P}_1(u)$ must be at least two. Hence

$$\phi_1 = \sum_{j=2}^{\infty} N(j) p^j (1-p)^{2k-j} - P_r \{ (\text{Errors detected in } \hat{u} \text{ only}) \cup (\text{errors detected in } \hat{P}_1(u) \text{ only}) \cap (\text{Remaining error pattern correctable by } C_1) \}.$$

Consider now the sliding block decoder performance for the case where $\hat{u}, \hat{P}_1(u)$ are interleaved bit by bit before decoding. Consider also the case where there are no errors detected in \hat{u} . Also note that the constraint length n_A of the codes of interest (see Table 5.1) are such that n_A is an even integer.

Then the sliding block decoder is capable of correcting all error patterns of t or less errors in a 'sliding window' of n_A bits, half of which are error-free. The probability that the errors in the block will be corrected is

$$P_C^1 = \sum_{j=2}^{\infty} N^1(j) p^j (1-p)^{k-j}$$

where $N^1(j)$ is the number of correctable patterns of j errors in a block of k bits (any one of which may be in error with probability p) such that no more than t errors occur in any span of $n_A/2$ bits. $N^1(j)$ can therefore be computed using the methods of Section 5. It follows that

$$\phi_1 = \sum_{j=2}^{\infty} N(j) p^j (1-p)^{2k-j} - 2(1-p)^k \sum_{j=2}^{\infty} N^1(j) p^j (1-p)^{k-j} \quad (D.1)$$

By a similar argument it follows that

$$\phi_2 = \phi_3 = 2(1-p)^k [1 - (1-p)^{n-k}] \sum_{j=1}^{\infty} N^1(j) p^j (1-p)^{k-j} \quad (D.2)$$

and also

$$\phi_4 = (1-p)^{2k} [1 - (1-p)^{n-k}]^2 \quad (D.3)$$

Consider now the probability q_2 that a message u is successfully recovered from the second received information vector $\{\hat{u}^1, \hat{P}_0^1(u)\}$ (i.e. the third transmission attempt) given that u could not be recovered error-free after the first message block $\{\hat{u}, \hat{P}_0(u)\}$ and second parity block $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ were received. Following the method of Lin & Yu [34] we define the following events:

1. Let H_1 be the event that errors are detected in the first received information vector $\{\hat{u}, \hat{P}_0(u)\}$ and let H_1^1 be the event that \hat{u} contains at least one error;
2. Let H_2 be the event that errors are detected in the received parity retransmission vector $\{\hat{P}_1(u), \hat{P}_0(P_1(u))\}$ and let H_2^1 be the event that $\hat{P}_1(u)$ contains at least one error;
3. Let H_3 be the event that the errors in $\{\hat{u}, \hat{P}_1(u)\}$ form a detectable but not correctable error pattern;
4. Let H_4 be the event that errors are detected in the second received information vector $\{\hat{u}^1, \hat{P}_0^1(u)\}$ and let H_4^1 be the event that \hat{u}^1 contains at least one error; and
5. Let H_5 be the event that the errors in $\{\hat{u}^1, \hat{P}_1(u)\}$ form a correctable error pattern.

Then it follows that

$$\begin{aligned}
 q_2 &= P_r \{ \bar{H}_4 \cup H_4 H_5 / H_1 H_2 H_3 \} \\
 &= P_r \{ \bar{H}_4 / H_1 H_2 H_3 \} + P_r \{ H_4 H_5 / H_1 H_2 H_3 \} \\
 &= P_r \{ \bar{H}_4 \} + \frac{P_r \{ H_1 H_2 H_3 H_4 H_5 \}}{P_r \{ H_1 H_2 H_3 \}} \quad (D.4)
 \end{aligned}$$

where

$$P_r \{ \bar{H}_4 \} = 1 - p_B = (1-p)^n \quad (D.5)$$

and

$$P_r \{ H_1 H_2 H_3 \} = p_B (1 - q_1)$$

Since H_1^1 , H_2^1 and H_4^1 are subsets of H_1 , H_2 and H_4

$$P_r \{ H_1 H_2 H_3 H_4 H_5 \} \geq P_r \{ H_1^1 H_2^1 H_3^1 H_4^1 H_5 \} \quad (D.6)$$

which we use to derive a lower bound on q_2 (which will be quite tight for $n \gg k$). Let M_i be the event that $P_1(u)$ contains i errors in one of the ℓ possible error patterns where

$$\ell = \binom{k}{i}$$

then

$$P_r \{ H_1^1 H_2^1 H_3^1 H_4^1 H_5 \} = \sum_{i \geq 1} P_r \{ M_i \} P_r \{ H_1^1 H_3^1 H_4^1 H_5 / M_i \} \quad (D.7)$$

Now the conditional probability $P_r \{ H_1^1 H_3^1 H_4^1 H_5 / M_i \}$ can be expressed in the following terms

$$P_r \{ H_1^1 H_3^1 H_4^1 H_5 / M_i \} = P_r \{ \hat{u} \text{ contains an error pattern such that}$$

$\{ \hat{u}, \hat{P}_1(u) \}$ is uncorrectable}

- $P_r \{ \hat{u}^1 \text{ contains an error pattern such that}$

$\{ \hat{u}^1, \hat{P}_1(u) \}$ is uncorrectable}

which can be lower bounded as

$$P_r \{H_1^1 H_3^1 H_4^1 H_5^1 / M_i\} \geq \{P_B(k, \frac{n_A}{2})\} \{1 - P_B(k - \frac{in_A}{2}, \frac{n_A}{2})\} \quad (D.8)$$

where the terms of the form $P_B(x, y)$ represent the probability of incorrect decoding for a block of length x bits with a $1/2$ rate code decoder capable of correcting up to t errors in any 'sliding window' span of y bits. Following the procedure of Section 7.3, Equation (D.8) can be explained as follows. $P_B(k, \frac{n_A}{2})$ can be thought of as the probability that \hat{u} contains an error pattern and $\hat{P}_1(u)$ contains no errors and the interleaved block $\{\hat{u}, \hat{P}_1(u)\}$ is uncorrectable. Clearly, for $\hat{P}_1(u)$ containing i errors ($i \geq 1$), the block error probability after decoding would be greater.

Consider now the second term in (D.8) which provides a lower bound on the probability that the interleaved vectors \hat{u}^1 , and $\hat{P}_1(u)$ are correctable given that $\hat{P}_1(u)$ contains i errors. This lower bound is arrived at as follows.

For every error location in $\hat{P}_1(u)$, assume the decoder cannot correct errors in any n_A bits either side of the error. So far as the alternatively interleaved \hat{u}^1 bits are concerned, this is equivalent to precluding the possibility of error correction $\frac{n_A}{2}$ bits either side of the errors. There are i such errors and therefore $\frac{in_A}{2}$ bits affected in \hat{u}^1 . Outside these spans, there are no errors in $\hat{P}_1(u)$. The probability of being unable to correct all errors in \hat{u}^1 outside the precluded spans is given by $P_B(k - \frac{in_A}{2}, \frac{n_A}{2})$ as explained in Section 7.3. Therefore the probability of being able to correct all errors in $\{\hat{u}^1, \hat{P}_1(u)\}$ will be greater than the second term in equation (D.8).

Bibliography

1. CCITT: Recommendation X.25 "Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode on public data networks," VIIth Planary Assembly, Doc. No. 7, Geneva, 1980.
2. S. Lin and D.J. Costello, *Coding Fundamentals and Error Control Techniques*, Prentice Hall (to be published 1982).
3. M.J. Miller and S. Lin, "The analysis of some selective-repeat ARQ schemes with finite receiver buffer," *IEEE Trans. Comm.*, Vol. COM-29, No. 9, pp. 1307-1315, September 1981.
4. J.J. Metzner and K.C. Morgan, "Coded Binary Decision-Feedback Communication Systems," *IRE Trans. Comm. Syst.*, Vol. CS-8, pp. 101-113, June 1960.
5. B. Reiffen, W.G. Schmidt, H.L. Yudkin, "The Design of an Error-Free Data Transmission System for Telephone Circuit," *AIEE Trans.* Vol. 80, pp. 224-231, July 1961.
6. 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.*, pt. 1, No. 58, pp. 577-580, January 1962.
7. M. Horstein, "Sequential transmission using noiseless feedback," *IEEE Trans. Inform. Th.*, IT-9, pp. 136-143, July 1963.
8. R.J. Benice and A.H. Frey, "An analysis of retransmission systems," *IEEE Trans. Comm. Tech.*, COM-12, pp. 135-145, December 1964.
9. R.J. Benice and A.H. Frey, "Comparison of error control techniques," *IEEE Trans. Comm. Tech.*, COM-12, pp. 146-154, December 1964.
10. A.J. Viterbi, "The effect of sequential decision feedback on communication over the Gaussian channel," *Information and Control*, Vol. 8, pp. 80-92, February 1965.
11. R.L. Kashyap, "Feedback coding schemes for an additive noise channel with a noisy feedback link," *IEEE Trans. Inform. Th.*, IT-14, pp. 471-480, May 1968.
12. E.Y. Roher and R.L. Pickholtz, "An analysis of effectiveness of hybrid transmission schemes," *IBM J. Res. Develop.*, pp. 426-433, July 1970.

13. H.O. Burton and D.D. Sullivan, "Errors and error control," *Proc. IEEE*, Vol. 60, pp. 1293-1301, November 1972.
14. A.G. Gatfield, "ARQ error control on the satellite channel," ICC '74 Conf. Rec., June 1974
15. D.M. Mandelbaum, "An Adaptive-Feedback Coding Scheme using Incremental Redundance," *IEEE Trans. Info. Theory*, Vol. IT-20. No. 3, pp. 388-389, May 1974.
16. J.R. Kersey, "Synchronous Data Link Control," *Data Communications*, May/June 1974.
17. A.R.K. Sastry, "Improving ARQ performance on satellite channels under high error rate conditions," *IEEE Trans. Comm.*, COM-23, pp. 436-439, April 1975.
18. A.R.K. Sastry, "Performance of hybrid error control schemes on satellite channels," *IEEE Trans. Comm.*, COM-23, pp. 689-694, July 1975.
19. A.R.K. Sastry and L.N. Kanal, "Hybrid error control using retransmission and generalized burst-trapping codes," *IEEE Trans. Comm.*, COM-24, pp. 385-393, April 1976.
20. J.J. Metzner, "A study of an efficient retransmission strategy for Data Links," NTC '77 Conf. Rec., pp. 3B: 1-1 to 3B: 1-5.
21. J.M. Morris, "On another Go-Back-N ARQ technique for high error rate conditions," *IEEE Trans. Comm.*, COM-26, pp. 187-189, January 1978.
22. A. Kaul, "Performance of High-Level Data Link control in Satellite Communications", *COMSAT Technical Rev.*, Vol. 8, pp. 41-87, Spring 1978.
23. J.M. Morris, "Optimal Blocklengths of ARQ Error Control Schemes," *IEEE Trans. Comm.*, Vol. COM-27, pp. 488-493, February 1979.
24. J.J. Metzner, "Improvements in Block-Retransmission Schemes," *IEEE Trans. Comm.*, Vol. COM-27, No. 2, pp. 525-532, February 1979.
25. M.C. Easton, "Efficient Transfer of Large Data Files over Satellite Links" Proc. 13th Hawaii International Conference on System Sciences, Honolulu, Hawaii, January 1979.
26. C. Fujiwara, M. Kasahara, K. Yamashita, and T. Namekawa, "Evaluation of error control techniques in both independent-error and dependent error channels," *IEEE Trans. Comm.*, COM-26, pp. 785-793, June 1978.

27. D. Towsley and J.K. Wolf, "On the statistical analysis of queue length and waiting times for statistical multiplexers with ARQ retransmission schemes," *IEEE Trans. Comm.*, COM-27, pp. 693-702, April 1979.
28. D. Towsley, "The Stutter Go-Back-N ARQ Protocol," *IEEE Trans. Comm.*, Vol. COM-27, pp. 869-875, June 1979.
29. T.C. Ancheta, "Convolutional Parity Check Automatic Repeat Request," presented at the 1979 *IEEE International Symposium on Information Theory*, Grignano, Italy, June 25-29, 1979.
30. M.C. Easton, "Batch Throughput Efficiency of ADCCP/HDLC/SKLC Selective-Repeat Protocols," *IEEE Trans. Comm.*, Vol. COM-28, No. 2, pp. 187-195, February 1980.
31. S. Lin and P.S. Yu, "An efficient error control scheme for satellite Communications," *IEEE Trans. Comm.*, Vol. COM-28, pp. 396-401, March 1980.
32. P.S. Yu and S. Lin, "An efficient selective-repeat ARQ scheme for satellite channels," *ICC '80 Conf. Rec.*, June 1980.
33. M.J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," *NTC '80 Conf. Proc.*, Houston, Texas, November 30-December 3, 1980
34. S. Lin and P.S. Yu, "A Hybrid ARQ scheme with parity retransmission for error control of Satellite Channels." *IEEE Trans. Comm.*, Vol. COM-30, June 1982.
35. A Drukarev and D.J. Costello, Jr., "ARQ error control using sequential decoding," *ICC '80 Conf. Rec.*, June 1980.
36. M.C. Easton, "Design Choices for Selective-Repeat Retransmission Protocols," *IEEE Trans. Comm.*, Vol. COM-29, No. 7, pp. 944-953, July 1981.
37. D. Towsley, "A Statistical Analysis of ARQ Protocols Operating in a Non-independent Error Environment," *IEEE Trans. Comm.*, Vol. COM-29, No. 7, pp. 971-982, July 1981.
38. A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, 1979.
39. F.J.W. Symons, "The verification of Communication Protocols using Numerical Petrinets," *Australian Telecom Research Journal*, Vol. 14, No. 1, pp. 34-38, 1980.

40. P. Bylanski and D.G.W. Ingram, *Digital Transmission Systems*, Peregrinus Ltd., U.K., 1976.
41. J. Riordan, *Combinatorial Identities*, Wiley, 1968.
42. S.S.L. Chang, "Theory of information feedback systems," *IRE Trans. Inform. Th.*, Vol. IT-2, pp. 29-40, September 1956.
43. B. Harris and K.C. Morgan, "Binary symmetric decision-feedback systems," *Trans. AIEE (Commun. and Electronics)*, Vol. 38, pp. 436-443, September 1958.
44. R.D. Stuart, "An insert system for use with feedback communication links," *IEEE Trans. Commun. Sys.*, Vol. CS-11, pp. 142-143, March 1963.
45. J. Labetoulle and G. Pujolle, "HDLC Throughput and response time for directional data flow with non uniform frame sizes," *IEEE Trans. on Computers*, Vol. C-30, No. 6, pp. 405-413, June 1981.
46. R.E. Kahn, et al., "Advances in packet radio technology" *Proc. IEEE*, Vol. 66, pp. 1468-1496, November 1978.
47. H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Trans. Inform. Th.*, Vol. IT-26, No. 5, pp. 540-547, September 1980.
48. E.J. Weldon, Jr., "An Improved Selective-Repeat ARQ Strategy," *IEEE Trans. Comm.*, Vol. COM-30, No. 3, March 1982.
49. J.L. Massey, *Threshold Decoding*, Cambridge, Mass.: MIT Press, 1963.
50. E. Parzen, *Stochastic Processes*, Holden-Day, 1962.