```
16:55:34          OCA PAD INITIATION - PROJECT HEADER INFORMATION          10/23/91
                                                                            Active
Project #: E-21-Z97          Cost share #: E-21-338           Rev #: 0
Center # : 10/24-6-R7356-0A0  Center shr #: 10/22-1-F7356-0A0  OCA file #:
                                                               Work type : RES
Contract#: FOREIGN AGREEMENT 10/14/91      Mod #:              Document  : AGR
Prime    #:                                                    Contract entity: GTRC

Subprojects ? : N                                              CFDA: N/A
Main project #:                                                PE #: N/A


Project unit:                 ELEC ENGR     Unit code: 02.010.118
Project director(s):
   WICKER S B                 ELEC ENGR     (404)894-3129



Sponsor/division names: BINARY COMMUNICATIONS INC      / VICTORIA, CANADA
Sponsor/division codes: 701                            / 017


Award period:     911014   to    921013  (performance)    921013  (reports)

Sponsor amount             New this change              Total to date
     Contract value                 0.00                         0.00
     Funded                         0.00                         0.00
Cost sharing amount                                          4,950.00

Does subcontracting plan apply ?: N

Title: ERROR CONTROL CODING SIMULATOR


                         PROJECT ADMINISTRATION DATA

OCA contact: E. Faith Gleason         894-4820

 Sponsor technical contact               Sponsor issuing office

 DR. VIJAY K. BHARGAVA                   BINARY COMMUNICATIONS INC.
 (604)477-5664                           (604)477-5664

 BINARY COMMUNICATIONS INC               2769 ARBUTUS ROAD
 VICTORIA, B.C. V8N 5X8                  VICTORIA, BRITISH COLUMBIA V8N 5X8
 CANADA                                  CANADA


Security class (U,C,S,TS) : U            ONR resident rep. is ACO (Y/N): N
Defense priority rating   :   N/A        N/A supplemental sheet
Equipment title vests with:    Sponsor        GIT X

Administrative comments -
* INITIATION. FOREIGN SPONSOR.
```

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 06/22/93

Project No. E-21-Z97 Center No. 10/24-6-R7356-0A0

Project Director WICKER S B School/Lab ELEC ENGR

Sponsor BINARY COMMUNICATIONS INC/VICTORIA, CANADA

Contract/Grant No. FOREIGN AGREEMENT 10/14/91 Contract Entity GTRC

Prime Contract No.

Title ERROR CONTROL CODING SIMULATOR

Effective Completion Date 921013 (Performance) 921013 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory & Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

CommentsEFFECTIVE DATE 10-14-91. CONTRACT VALUE $0.

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other CARL BAXTER-FMD	Y
FRED CAIN-OOD	Y

NOTE: Final Patent Questionnaire sent to PDPI.

AN ERROR CONTROL CODING SIMULATOR

Binary Communications Inc.
Victoria, British Columbia
CANADA

September 15, 1991 - June 30, 1993

FINAL REPORT

Professor Stephen B. Wicker
Principal Investigator

Coding and Information Theory Laboratory
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332 USA
(404) 894-3129
wicker@ee.gatech.edu

1. Introduction

For more than a decade following its genesis in 1949, the field of error control coding focused on the problem of error control for unidirectional communication channels. This design problem may be stated as follows: how can one encode a block of data so that the receiver can correct a sufficient number of the error patterns induced by the channel while simultaneously maximizing the data rate of the system? Error control coding for unidirectional channels is frequently called Forward Error Correction and is referred to here by its acronym "FEC". The advent of the communication network led to the development of a second approach to error control. Networks frequently provide for bidirectional communication between users, allowing one user to react to a message from another user by sending a response of some sort. The return channel that carries the receiving user's response is frequently called the feedback channel. The error control system design problem can now be augmented by the following question: how can one use the feedback channel to improve the receiver's ability to correct errors while further improving the data rate of the communication system? Error control coding systems that take advantage of a feedback channel are referred to here as bidirectional error control (BEC) systems.

This report summarizes the results of a research program that examined various means of constructing BEC systems. It is intended to illustrate the publications that emerged from this effort. The publications are listed at the end of the report, while reprints of the most important publications are included as appendices.

2. Single Decoder Type-I Hybrid-ARQ Protocols

The simplest and oldest of the BEC systems introduce redundant symbols into the data stream solely for the purpose of error detection. The redundant symbols ensure that there are a large number of sequences of symbol values that are invalid. If the receiver sees an invalid pattern in a demodulated packet of symbols, it requests a retransmission of that packet under the natural assumption that one or more error have been caused by noise on the channel. These "detection only" BEC schemes are called "Automatic Repeat Request" (ARQ) protocols [LIN]. They provide significantly better reliability performance than FEC systems that introduce the same number of redundant symbols. The price for this improved reliability performance is exacted in the form of a reduction in throughput caused by retransmission requests. If the channel remains good and received symbol errors remain infrequent, the ARQ protocol performs quite well. But, if the channel degrades to the point that one or more symbol errors occur in every received packet, then the ARQ protocol fails catastrophically.

In applications where symbol errors occur far too frequently for the use of an ARQ protocol, the data can be encoded for both error detection and error correction. The error control system is designed to correct the most frequently occurring error patterns while reserving detection capacity for the less frequently occurring patterns. The throughput performance is improved through the correction of the frequently occurring patterns while reliability performance is maintained by not trying to correct the less frequently occurring (and therefore higher weight and more likely to cause a decoder error) error patterns. Such a combined ARQ/FEC protocol is called a "type-I hybrid-ARQ" protocol [LIN].

Type-I hybrid-ARQ protocols have traditionally been implemented using two encoders and two decoders (FEC error correction combined with CRC error detection). One of the first research projects pursued under this contract was the investigation of single encoder/single decoder type-I systems. This work was motivated by a paper by Drukarev and Costello [DRU] in which a sequential FEC decoder for convolutional codes is modified to implement two different type-I hybrid-ARQ protocols. In each of the two protocols, the decoder computes a reliability statistic using information generated during the execution of a sequential decoding operation. The reliability statistic is compared to a threshold to determine whether the decoded data obtained from a received packet is sufficiently reliable. If not, a retransmission of the packet is requested. The retransmission threshold is used to establish a balance between error detection and error correction within the decoder. If detection is emphasized, reliability is favored at the expense of throughput. If correction is emphasized, the opposite effect is seen. Single encoder/single decoder protocols have several advantages over double encoder/double decoder systems. The most obvious is that there is no longer any need for the additional CRC encoder and decoder circuitry. Though CRC circuitry is very simple, it can introduce a substantial amount of delay. The single encoder/single decoder approach also allows for the dynamic reallocation of error correction and error detection capacity through the manipulation of the retransmission threshold. The FEC/CRC approach does, however, allow for more flexibility in setting the error detection parameters without affecting the error correction system. The Principal Investigator and his students have applied the single encoder/single decoder approach to a series of decoding algorithms.

2.1 Majority Logic Decoders for Cyclic and Convolutional Codes

The first of the FEC decoder modifications investigated focused on majority logic decoders for convolutional [5] and cyclic block [7],[16] codes. Majority logic decoders poll a set of check sums computed from a received codeword to estimate the values of the error bits affecting individual information bits. The extent of the majority formed during these polling operations is a source of reliability information that can be used in the definition of a type-I protocol.

The hybrid-ARQ majority-logic decoding rule is as follows. Let η be the number of the J check sums orthogonal on error bit e that have values of one. Let τ be a nonnegative integer less than J. If $\eta \geq \lfloor J/2 \rfloor + \lfloor \tau/2 \rfloor + 1$, then e is assumed to have a value of one. If $\eta \leq \lfloor J/2 \rfloor - \lceil \tau/2 \rceil$, then e is assumed to have a value of zero. If $\lfloor J/2 \rfloor - \lceil \tau/2 \rceil < \eta < \lfloor J/2 \rfloor + \lfloor \tau/2 \rfloor + 1$ then a retransmission of the packet is requested

Figures 1 and 2 show the impact on reliability and throughput for a typical case. Note that as the width of the retransmission region increases, the reliability performance improves with respect to the FEC case. Note also the penalty paid in the form of reduced throughput. In all cases, however, there is an operating region that allows for good throughput performance while providing significantly improved reliability performance. Analysis and simulation results are discussed in detail in [5], [7], and [16].

2.2 Reed-Solomon and BCH Decoders

Consider a bounded distance FEC decoder for a linear block code. Under most channel conditions, if the decoder corrects a large number of errors in a received packet, it is more likely to be committing a decoder error than if it corrects few or no errors. A type-I protocol can be created by simply establishing an allowed error correction threshold somewhere below the maximum correction capability of the code. If the number of errors corrected in a received word exceeds the threshold, then a retransmission request is generated. This simple idea leads to the development of an extremely powerful type-I protocol based on Reed-Solomon codes. In the Berlekamp-Massey and Euclidean decoding algorithms for Reed-Solomon and BCH codes, the degree of the error location polynomial indicates the number of errors to be corrected by the decoder. In the RS-HARQ system [4] the degree of the error location polynomial is compared to a retransmission threshold and the appropriate action taken. The geometry of the Reed-Solomon codes make this an unexpectedly powerful system. Reed-Solomon codes are excellent error correcting codes, but the decoding spheres defined by a Reed-Solomon bounded distance decoder do not define a good sphere packing. This can be seen by noting that the probability of decoding failure in most Reed-Solomon error control systems is substantially higher than the probability of decoder error. The space between the decoding spheres can thus be used to great advantage for error detection in a type-I hybrid-ARQ protocol.

In [9], [19], and [21] the RS-HARQ system is extended to include erasure decoding and is analyzed for the case of the slowly fading Rayleigh channel. In the extended system, the number of erasures *s* and the number of errors *e* to be corrected in a received packet are used to compute a reliability statistic $l = (2e + s)$. l is compared to the *effective diameter* d_e of the RS-HARQ system to determine whether a retransmission request is to be generated. Figures 3 and 4 indicate the performance provided for a slowly fading Rayleigh channel.

2.3 The Viterbi Decoder and the Error Trapping Algorithm

Yamamoto and Itoh were the first to introduce a type-I hybrid-ARQ protocol based on the Viterbi decoder [YAM]. They noted that the Viterbi decoder is more likely to make a decoder error whenever the partial path metric of the maximum likelihood path is close to that of a nonsurviving path at one or more of the nodal decision points. The Principal Investigator and one of his students (B. A. Harvey, doctoral thesis [2]) developed an alternative type-I protocol by taking a somewhat different approach. It was noted that the Viterbi decoder is more likely to make a decoder error whenever the partial path metric of the maximum likelihood path increases rapidly over a short span of the trellis. This observation lead to the development of the error trapping algorithm [6], [17]. The rate of increase of the partial path metric for all paths is computed over a sliding window of fixed length. Any path whose rate of increase exceeds a preset threshold is declared unreliable. If all paths are declared unreliable before decoding is completed, then a retransmission request is generated.

The error trapping algorithm has a number of interesting properties when applied to hard decision Viterbi decoders. Using a combinatorial attack, it is possible to derive an exact expression for the probability that a packet is correctly accepted on any given transmission. No such expression has been found for the Yamamoto and Itoh algorithm. The error trapping algorithm also has a very natural implementation in Viterbi decoders that use the trace-back algorithm to make decisions on information bits.

2.4 Other Modified FEC Systems

In [3] the application of the Yamamoto and Itoh algorithm to trellis coded modulation systems is investigated. Upper and lower bounds on throughput and reliability are derived; the bounds indicate that the performance of the resulting type-I system is significantly better than that of the TCM FEC system.

The most recent efforts in the area of modified FEC systems for type-I protocols focused on Reed-Muller codes. Reed-Muller codes do not currently receive a great deal of attention because they are "weak codes that are easy to decode" [BERL]. As there are now several strong codes that are easy to decode, Reed-Muller codes have lost the initial appeal that sent them out into the far reaches of the solar system aboard the Mariner spacecraft of the late 1960's and early 1970's. Increased attention is expected, however, as the extraordinarily fast RM decoders find applications in optical systems with high data rates.

It was found that significant throughput of reliable data can be obtained from a hypothetical Mariner spacecraft with a severely reduced transmitter power level through retransmission requests and type-I hybrid-ARQ Reed-Muller decoding. The algorithm presented in [12] and [26] is a modification of Green's maximum likelihood FEC decoding algorithm for first-order Reed-Muller codes (also known as the Green machine). The best results were obtained, however, through the creation of a type-II hybrid-ARQ Reed-Muller protocol. Type-II protocols provide a simple form of packet combining, the subject of the next section.

3. Packet Combining

Type-I hybrid-ARQ protocols postpone the catastrophic failure exhibited by ARQ protocols as the communication channel degrades, but they are unable to avoid it altogether. Eventually a point is reached when uncorrectable error patterns are occurring with such frequency that throughput on the type-I system becomes negligible. It is possible to further extend the operating range of a BEC system in the low SNR region through the use of packet combining. Packet combining systems differ from type-I protocols in that they do not discard received packets that have caused the generation of a retransmission request, but instead combine all received packets in an attempt to create a single packet that can be reliably decoded. As the channel deteriorates, the packet combining system uses an increasing number of packets, maintaining at least a small level of throughput under extremely adverse channel conditions.

Packet combining systems can be divided into two categories: diversity combining systems and code combining systems. Diversity combining systems use symbol-by-symbol combining to create a single rate R encoded packet from several copies of a rate R encoded packet. The goal is to increase the effective signal to noise ratio of the symbols comprising the packet resulting from the combining operation. In code combining systems [CHASE], L received codewords encoded at rate R are combined to create a single noise corrupted codeword with rate R/L. The goal is to decrease the rate of the code until there is sufficient redundancy in the packet resulting from the combining operation to reliably correct the errors caused by the noise on the channel.

3.1 Convolutional Codes: Majority Logic Diversity Combining

Majority logic diversity combiners poll multiple copies of a received symbol, selecting the majority value for the corresponding coordinate position in the combined packet. In [5] a hard decision voting scheme was used to implement majority logic combining in the type-I hybrid-ARQ majority logic decoder for convolutional codes discussed in section 2.1. To ensure that no ties occur, retransmissions in this scheme alternate between packets containing two copies of the information bits and packets containing two copies of the parity bits (assuming a rate 1/2 code is in use; other strategies can be used for other rates). When combined with the packet from the initial transmission, the retransmitted packets ensure that there is always an odd number of copies of both the information and parity bits. Figure 5 shows the resulting improvement in throughput performance at low signal to noise ratios. Note that the throughput reduction with decreasing SNR is much more graceful than in the case without combining (Figure 2).

Majority logic combining can also be applied to hard decision Viterbi decoders that have been modified to implement the error trapping algorithm discussed earlier [22]. In this case the retransmitted packet formatting strategy is independent of the code rate. Retransmissions alternate between packets containing two copies of the even numbered bits and packets containing two copies of the odd numbered bits. Figure 6 shows the resulting improvement in throughput performance.

3.2 Convolutional Codes: Averaged Diversity Combining

Diversity combining can be extended to soft decision Viterbi decoders through the use of symbol copy averaging. In averaged diversity combining systems, the demodulated values for all received copies of a given bit are simply averaged by adding them up and dividing the result by the number of copies used in the summation. In [14] and [22] averaged diversity combining is applied to soft decision Viterbi decoders that have been modified to implement the Yamamoto and Itoh type-I hybrid-ARQ algorithm. The resulting improvement in performance is shown in Figure 7. [14] and [22] also show that the averaged diversity combining system provides exactly the same reliability and throughput performance as a code combining system that interleaves all received packets to create convolutional codewords of lower rate.

In [14] and [22] the averaged diversity combining results are extended to cover "moderately varying channels" (i.e. channels that are constant over a packet transmission time, but may vary from packet to packet). One of the more interesting results to emerge from this study was that averaged diversity combiners perform as well as weighted diversity combiners over many highly nonstationary channels.

3.3 Convolutional Codes: Weighted Diversity Combining

For moderately varying channels, it is sometimes necessary to weight the received packets before combining to prevent a single extremely noisy packet from forcing a large number of retransmission requests. In [CHASE] a series of weights are derived for code combining over Binary Symmetric and AWGN channels. These weighting factors are obtained using ideal channel side information. In [14] and [22] these weights are used to implement a weighted diversity combining scheme for hard and soft decision Viterbi decoders. [14] concludes by demonstrating a method for deriving the packet weights using side information generated by the Viterbi decoder, obviating the need for some type of channel noise measurement. It is shown that the resulting performance degradation caused by the suboptimal weights is extremely small.

3.4 Code Combining System Based on Punctured MDS Codes

Code combining systems use multiple received packets to create noise corrupted codewords from codes with increasingly lower rates. From an implementational standpoint this creates a problem, for the code combining decoder must be capable of decoding several different rates. In this project an effort was made to identify codes whose structure allows for a natural (and therefore easily implemented) approach to variable rate decoding. In [10] and [23] this approach is applied to create a code combining system based on punctured MDS codes (the MDS codes include the Reed-Solomon codes). It is shown that codewords from an extremely low rate, long MDS code can be partitioned to form a series of codewords from high rate punctured MDS codes. Codewords from this series can be combined to create increasingly lower rate MDS codewords down to the limiting rate of the "mother code". The decoder for the mother code can be used to decode all of the smaller codewords. This system uses the type-I protocol discussed earlier to generate retransmission requests.

In [10] and [23] this system is analyzed in detail for the case of Reed- Solomon codes in a type-II hybrid-ARQ protocol (a type-II protocol is a code combining system in which combining operations are limited to two packets at a time). An exact method for characterizing packets that have caused the generation of a retransmission requests is developed and used in a graph theoretic analysis to obtain exact throughput and reliability expressions.

3.5 The Reed-Muller Type-II Hybrid-ARQ Protocol

The type-I hybrid-ARQ Reed-Muller protocol discussed earlier can be extended to create a type-II protocol through puncturing and the use of a series of RM subcodes [12] and [26]. The modification allows for moderate levels of throughput at extremely low signal to noise ratios, providing some improvement over the performance of the type-I protocol. Unfortunately the relatively poor performance of long Reed-Muller codes prevented the definition of a practical RM system that combines more than two packets.

4. Rate Switching Systems

Packet combining systems can adapt rapidly to changes in channel conditions, but are limited in the number and selection of effective code rates that they can use. Typically all of the code rates available are of the form R/L, where R is a base rate and L is an integer. If the channel is sufficiently slowly varying with respect to the data rate (e.g. satellite links vary slowly with the weather), a different approach can be taken. It is possible to adopt a rate switching system whose code rate is selected through a channel state estimation procedure. The rate switching systems can provide theoretically infinite rate resolution. In practice some concessions must be made in the selection of code rates, but the possibilities are still more numerous than in the case of packet combining systems.

It has been noted by a number of authors that the retransmission statistics exhibited by a hybrid-ARQ protocol can be used as the basis for an adaptive error control system. This idea is pursued in [8], where a channel estimation strategy based on the number of retransmissions within a frame of packets is introduced and analyzed. The channel is modeled as an M-state Markov chain. Each state in the chain has a corresponding type-I protocol that provides acceptable performance as defined by a set of characteristic functions.

The rate switching approach is further developed in [1] and [13], where a sequential testing scheme based on retransmission statistics is used to drive a rate switching system. The sequential test is able to react much more quickly to changes in the channel noise level than the framed approach in [8]. The reliability and throughput performance of the sequential system is shown to be quite close to that of an ideal rate switching system with perfect channel state information.

In [1] and [13] several rate switching systems are developed and analyzed. The most promising systems use rate compatible punctured convolutional (RCPC) codes or Reed-Solomon codes in type-I hybrid-ARQ protocols. The RCPC system in particular provides for an extremely simple implementation in which the encoder and decoder need not be substantially modified beyond the type-I hybrid-ARQ modification.

Technical Publications and Presentations: NCR-9009877

Doctoral Theses

[1] Rice, M. D., "Adaptive Error Control Over Slowly Varying Channels," doctoral thesis, Spring 1991

[2] Harvey, B. A. "Adaptive Rate Convolutional Coding Using the Viterbi Decoder," doctoral thesis, Spring 1991

Books

[3] Wicker, S. B., "Trellis Coded Hybrid-ARQ Protocols," in *Communication, Control, and Signal Processing*, pp. 339 - 346, (Erdal Erikan, Ed.), Amsterdam: Elsevier Publishing (1990).

Refereed Journal Articles

[4] Wicker, S. B., "High Reliability Data Transfer over the Land Mobile Radio Channel," *IEEE Transactions on Vehicular Technology*, Volume 39, No 1, pp. 48 - 55, February 1990.

[5] Wicker, S. B., "Adaptive Rate Error Control Through the Use of Diversity Combining and Majority-Logic Decoding in a Hybrid-ARQ Protocol," *IEEE Transactions on Communications*, Volume 39, Number 3, pp. 380 - 386, March 1991.

[6] Harvey, B and Wicker, S. B., "Error-Trapping Viterbi Decoding for Type-I Hybrid-ARQ Protocols," *Canadian Journal of Electrical and Computer Engineering*, Volume 16, Number 1, pp. 5 - 12, January 1991.

[7] Rice, M. D. and Wicker, S. B., "Modified Majority-Logic Decoding of Cyclic Codes in Hybrid-ARQ Systems, *IEEE Transactions on Communications*, Volume 40, Number 9, pp. 1413 - 1417, September, 1992.

[8] Rice, M. D. and Wicker, S. B., "Adaptive Error Control for Slowly Varying Channels," *IEEE Transactions on Communications*, accepted.

[9] 9. Wicker, S. B., "Reed-Solomon Error Control Coding for Data Transmission over Rayleigh Fading Channels with Feedback," *IEEE Transactions on Vehicular Technology*, Volume 41, Number 2, pp. 124 - 133, May 1992.

[10] Wicker, S. B. and Bartz, M., "Type-II Hybrid-ARQ Protocols Using Punctured MDS Codes," *IEEE Transactions on Communications,* accepted pending revision.

[11] Rasmussen, L. and Wicker, S. B., "Trellis Coded Hybrid-ARQ Protocols over AWGN and Slowly Fading Rician Channels," *IEEE Transactions on Information Theory,* accepted pending revision.

[12] Wicker, S. B. and Bartz, M., "The Design and Implementation of Type-I and Type-II Hybrid-ARQ Protocols Based on First-Order Reed-Muller Codes," *IEEE Transactions on Communications,* accepted pending revision.

[13] Rice, M. D. and Wicker S. B., "A Sequential Testing Scheme for Adaptive Error Control on Slowly Varying Channels" *IEEE Transactions on Communications,* accepted.

[14] Harvey, B. A. and Wicker, S. B., "Packet Combining Systems Based on the Viterbi Decoder" *IEEE Transactions on Communications,* accepted.

Refereed Conference Presentations

[15] Wicker, S. B. and Harvey B., "Error-Trapping Viterbi Decoding in Type-I Hybrid-ARQ Protocols" 1990 International Symposium on Theory, San Diego, California, Paper WA2-6, January 17, 1990.

[16] Rice, M. D. and Wicker, S. B., "Modified Majority-Logic Decoding of Cyclic Block Codes," *Proceedings of the 1990 IEEE International Conference on Communications,* Atlanta, Georgia, pp. 332.3.1 - 332.3.5, April 16 - 19, 1990.

[17] Harvey A. B. and Wicker, S. B., "Error-Trapping Viterbi Decoding in a Type-I Hybrid-ARQ Protocol," *Proceedings of the 1990 IEEE International Conference on Communications,* Atlanta, Georgia, pp. 332.5.1 - 332.5.5, April 16 - 19, 1990.

[18] Rice, M. D. and Wicker, S. B., "Adaptive Rate Coding for Slowly Varying Channels." *Proceedings of the 1990 SBT/IEEE International Telecommunications Symposium,* Rio de Janerio, Brazil, pp. 176 - 180, September 3 - 6, 1990.

[19] Wicker, S. B., "Reed-Solomon Error Control Coding for Data Transmission over Personal Communication Networks," *Proceedings of the IEEE Southeastcon 1991,* Williamsburg, Virginia, pp. 437 - 441, April 7 - 10, 1991.

[20] Wicker, S. B. and Rice, M., "A Sequential Testing Scheme for Adaptive Error Control on Slowly Varying Channels" *Proceedings of the 1991 International Symposium on Theory,* Budapest, Hungary, pg. 30, June 23 - 28, 1991.

[21] Wicker, S. B., "A Reed-Solomon Hybrid-ARQ Protocol with Erasure Decoding for Mobile Communication Networks," *Proceedings of the International Symposium on Personal, Indoor, and Mobile Radio Communications,* London, England, pp. 58-62, September 23 - 24, 1991.

[22] Wicker, S. B., "Adaptive Rate Convolutional Coding using the Viterbi Decoder," *Proceedings of the International Winter Meeting on Coding and Information Theory,* p. 25 (summary), Essen, Germany, December 15 - 17. 1991.

[23] Wicker, S. B., "Type-II Hybrid-ARQ Protocols Using Punctured Reed-Solomon Codes," *Proceedings of the 1991 IEEE Military Communications Conference,* McLean, Virginia, pp. 52.2.1 - 52.2.6, November 4 - 6, 1991.

[24] Wicker, S. B., "Reed-Solomon Error Control Systems for Bidirectional Fading Channels", *Proceedings of the International Conference on Selected Topics in Wireless Communication,* pp. 76 - 79, Vancouver, British Columbia, Canada, June 25 - 26, 1992.

[25] Harvey, B. A. and Wicker, S. B., "Packet Combining Systems Based on the Viterbi Decoder," *Proceedings of the 1992 IEEE Military Communications Conference,* pp. 757 - 761, San Diego, California, October 11 - 14, 1992.

[26] Bartz, M. and Wicker, S. B., "Type-I and Type-II Reed-Muller Hybrid-ARQ Protocols" accepted for inclusion in the *Proceedings of the 1993 IEEE International Communications Conference,* May 23 - 26, 1993, Geneve, Switzerland.

Referenced Work by Other Authors

[BERL] E. Berlekamp, *Algebraic Coding Theory,* revised edition, Laguna Hills: Aegean Park Press, 1984.

[CHASE] D. Chase, "Code Combining - A Maximum-Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets," *IEEE Transactions on Communications,* Volume COM-33, pp. 385 - 393, May 1985.

[DRU] A. Drukarev and D. J. Costello Jr., "Hybrid ARQ Error Control Using Sequential Decoding," IEEE Transactions on Information Theory, Volume IT-29, pp. 521 - 535, July 1983.

[LIN] S. Lin, D. J. Costello Jr., and M. J. Miller, "Automatic-Repeat-Request Error Control Schemes," *IEEE Communications Magazine*, Volume 22, pp. 5 - 17, December, 1984.

[YAM] H. Yamamoto and K. Itoh, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," *IEEE Transactions on Information Theory*, Volume IT-26, pp. 540 - 547, September 1980

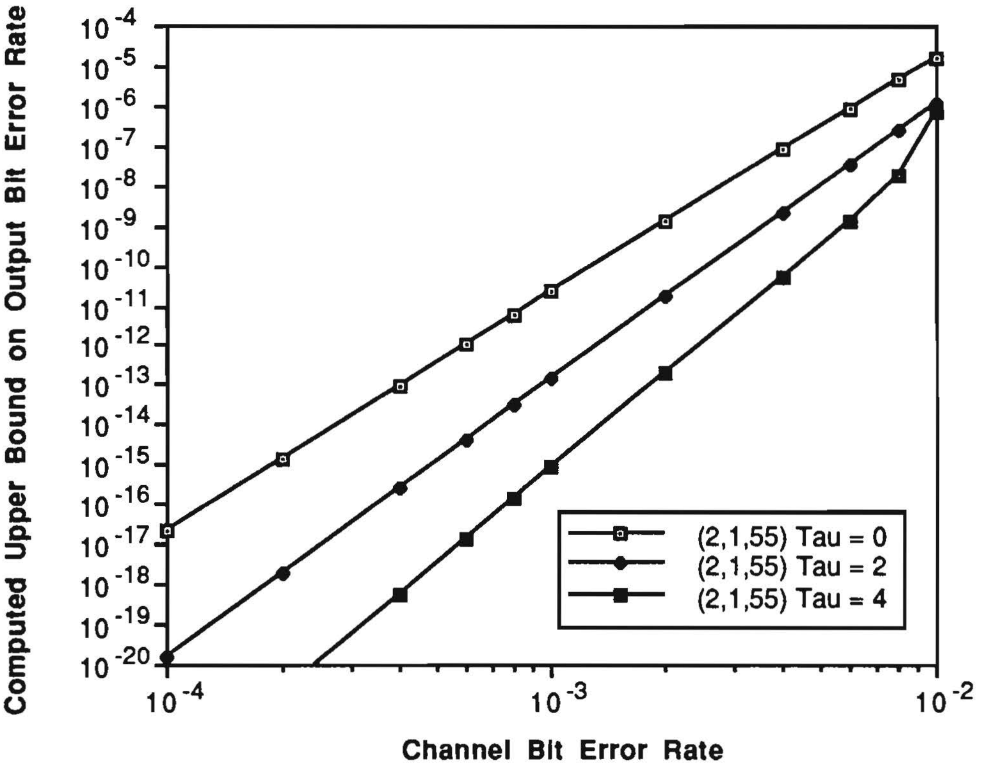

Figure 1: Reliability Performance for (2,1,55) Convoluational Code in a Majority-Logic Type-I Hybrid-ARQ Decoder (Packet Length = 1000).

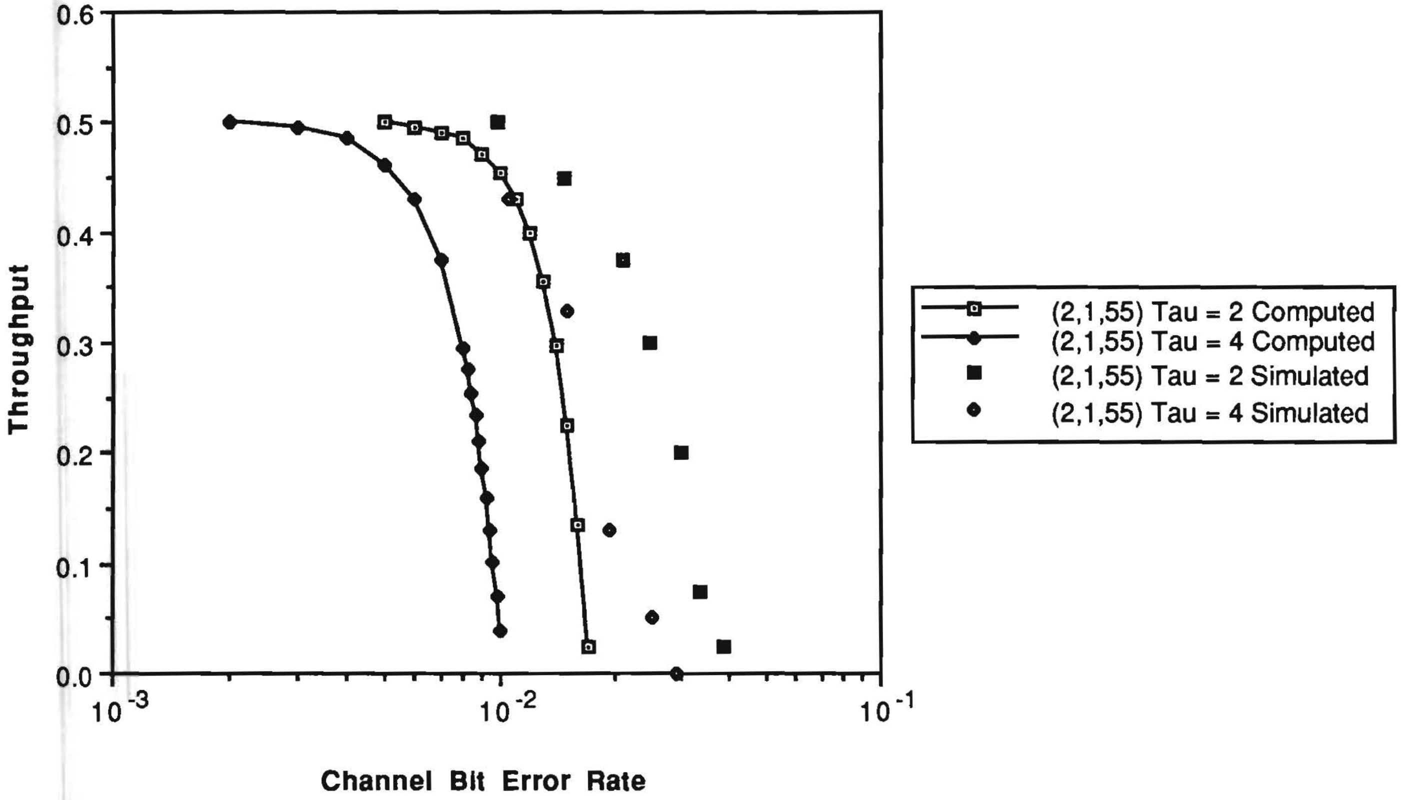

Figure 2: Throughput Performance for (2,1,55) Convoluational Code in a Majority-Logic Type-I Hybrid-ARQ Decoder (Packet Length = 1000).

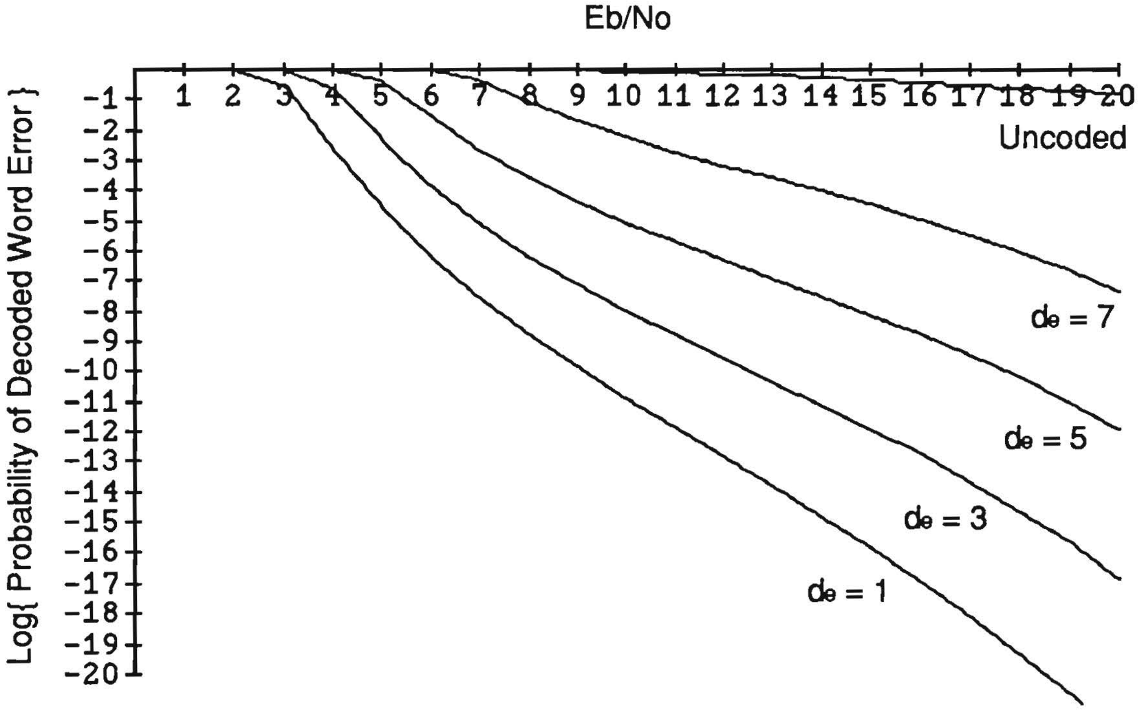

Figure 3 : Word Error Rate for a (64, 56) Reed-Solomon Code in the RS/HARQ System with erasures.

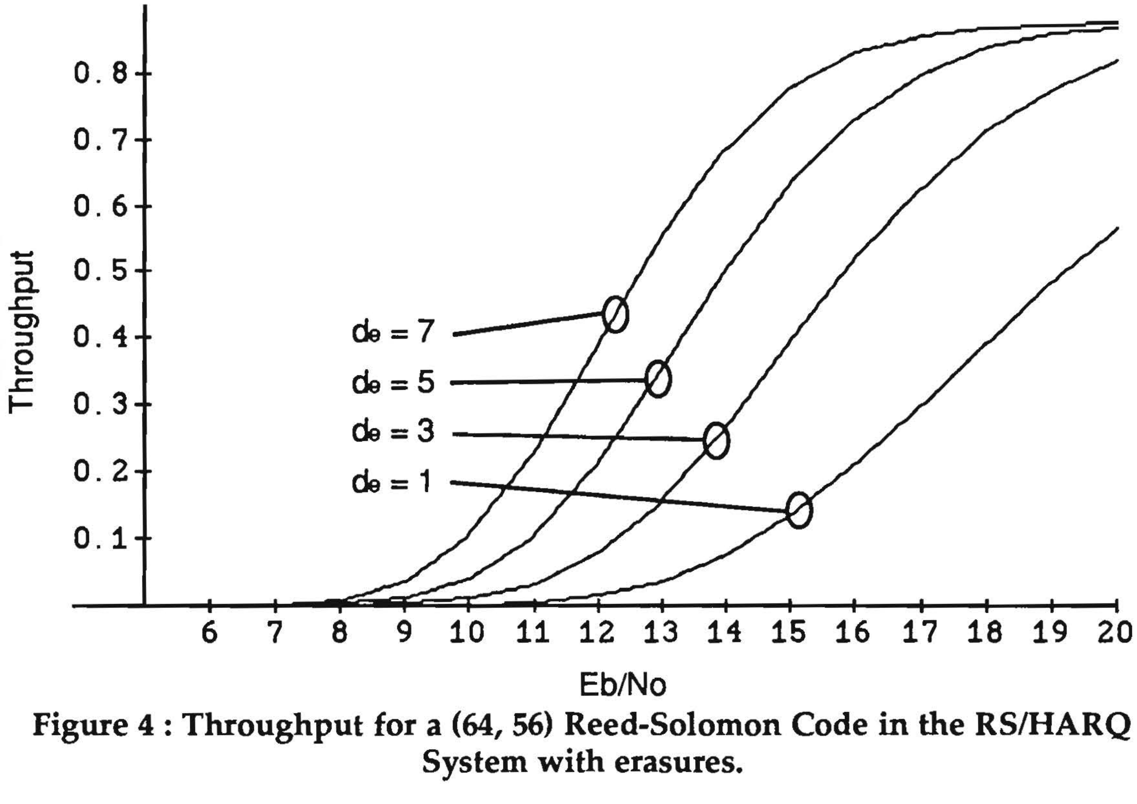

Figure 4 : Throughput for a (64, 56) Reed-Solomon Code in the RS/HARQ System with erasures.

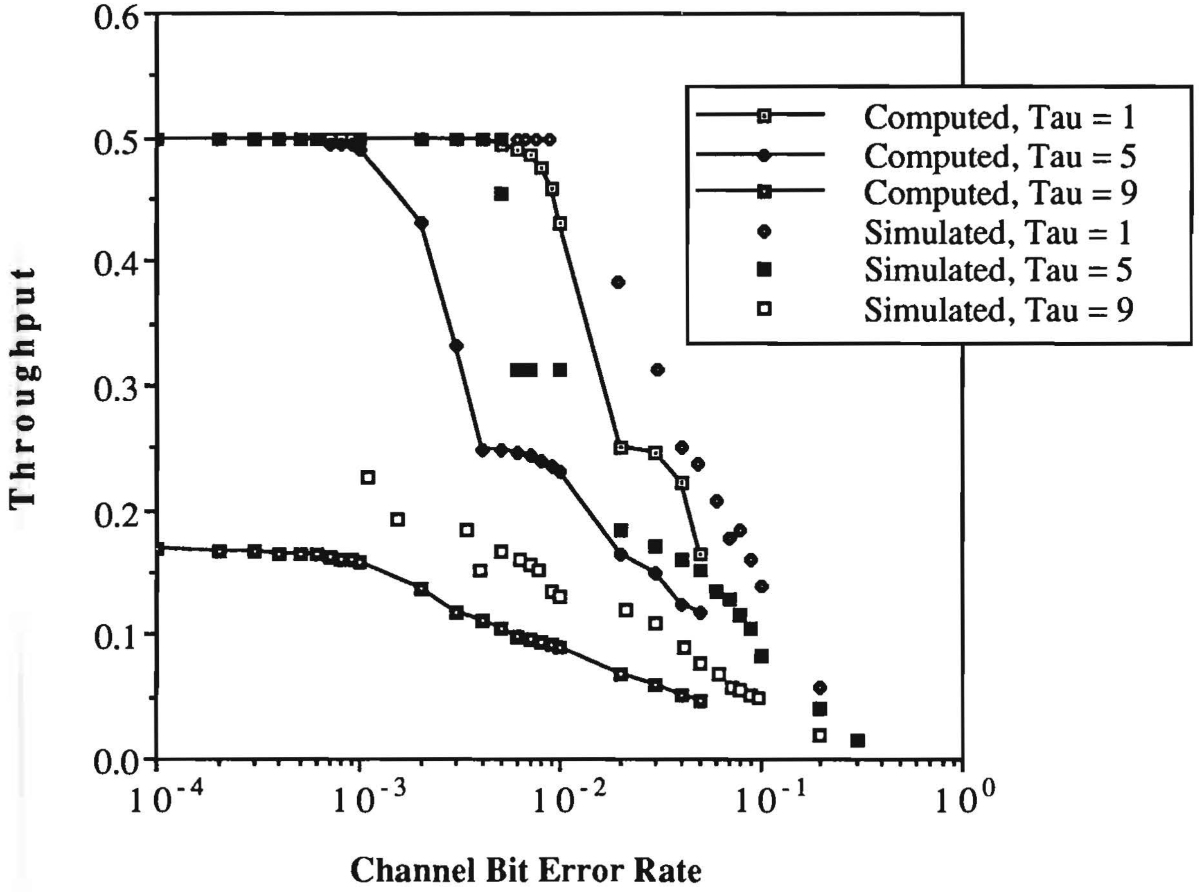

Figure 5: The lower bound on throughput and simulation results for a (2,1,55) convolutional code in a diversity combining majority-logic hybrid-ARQ protocol. Packet Length = 2000

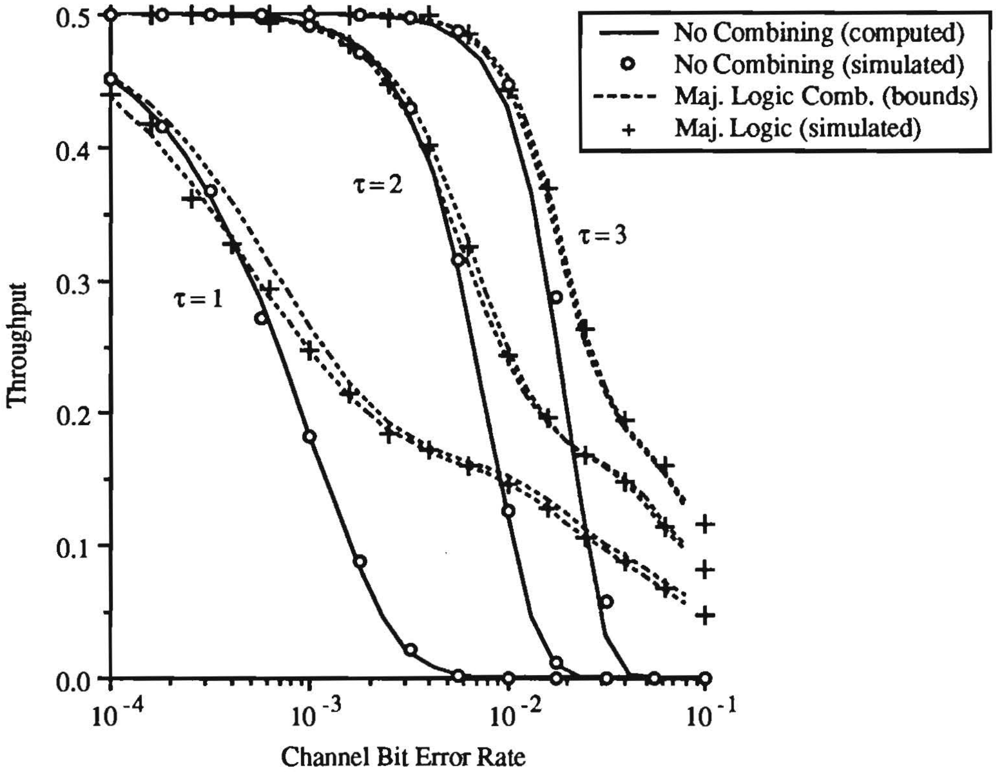

Figure 6: Throughput Performance for Hard Decision Viterbi Decoding with and without Majority-Logic Diversity Combining

(Decoder uses Error Trapping Algorithm with a (2, 1, 3) Convolutional Code)

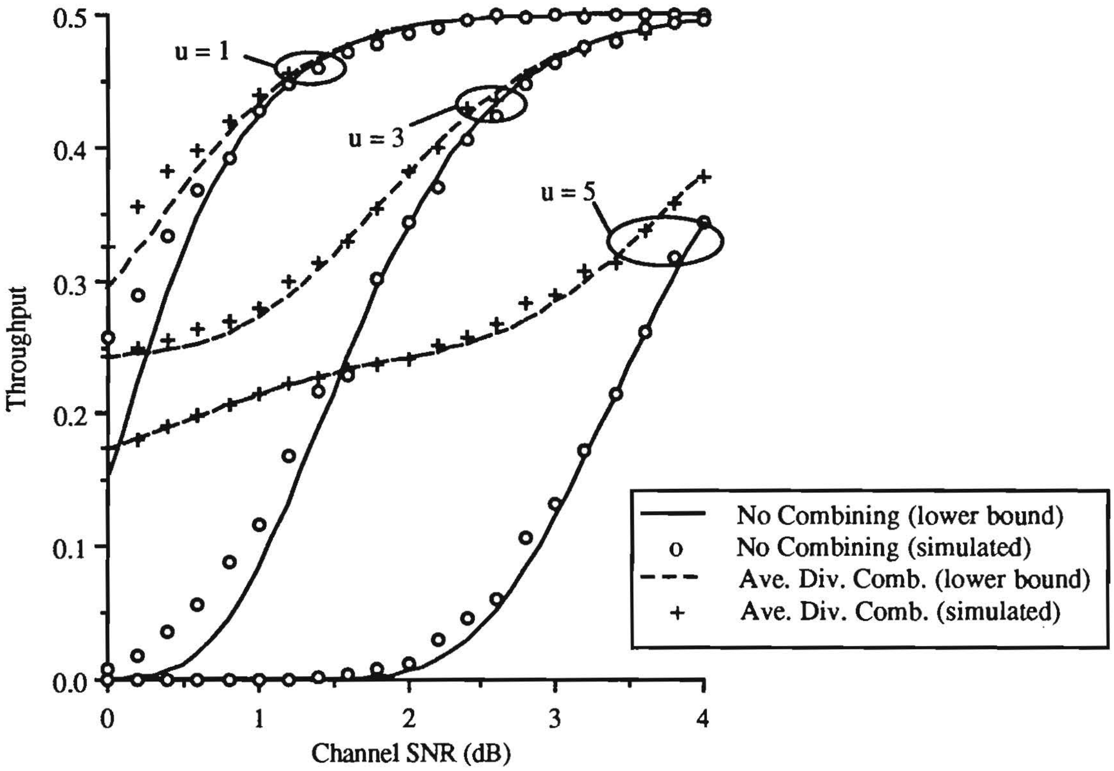

Figure 7: Throughput Performance for Soft Decision Viterbi Decoding with and without Averaged Diversity Combining

(Decoder uses Yamamoto-Itoh Algorithm with a (2, 1, 3) Convolutional Code)

APPENDICES

Reprints of several papers are included as appendices.

Reed–Solomon Error Control Coding for Rayleigh Fading Channels with Feedback

Stephen B. Wicker, *Member, IEEE*

***Abstract*— The use of nonbinary block error control codes over Rayleigh fading channels with feedback is examined. It is assumed that the fading is slow with respect to the rate of symbol transmission. Expressions are derived for the probabilities of channel symbol error and erasure, which are in turn used to develop expressions for code symbol error and erasure. Two erasure generation mechanisms are considered, one based on the existence of channel amplitude side information, the other not. This analytical framework is used to evaluate the performance of the Reed–Solomon/hybrid-ARQ protocol (RS/HARQ) over fading channels with feedback. The RS/HARQ system uses erasure decoding in a hybrid-ARQ protocol to provide excellent reliability performance at the expense of a reduction in throughput. The RS/HARQ protocol allows for the variation of the erasure threshold and the effective diameter of the decoding operation, providing a powerful, flexible error control system for a variety of fading channel applications.**

I. Introduction

In a large number of communication networks, the links established between users are bidirectional. The existence of two channels (forward and feedback when referenced to a single user) offers a significant opportunity for the development of powerful automatic-repeat-request (ARQ) error control systems. It has been shown in the literature that ARQ systems and their hybrids can provide better reliability performance than their forward-error-correcting (FEC) counterparts at the expense of a reduction in throughput [1]–[3]. A block diagram for a generalized ARQ system is shown in Fig. 1. The underlying functional principle of ARQ systems is the use of the feedback channel by the receiver to request retransmissions of data packets that are believed to be unreliable. In this paper the case of the Rayleigh fading forward channel with additive white Gaussian noise is considered. The feedback channel is assumed to be ideal (constant amplitude and noise-free).[1]

In the next section the performance of nonbinary block codes is considered in conjunction with both channel symbol and code symbol interleaving over Rayleigh fading channels. Two erasure generation mechanisms are considered in this development, one assuming the existence of ideal channel amplitude side information and the other not. Using both approaches, code symbol error and erasure processes are characterized in terms of the channel symbol error and erasure processes. It is shown that code symbol interleaving provides better symbol error and erasure performance than channel symbol interleaving.

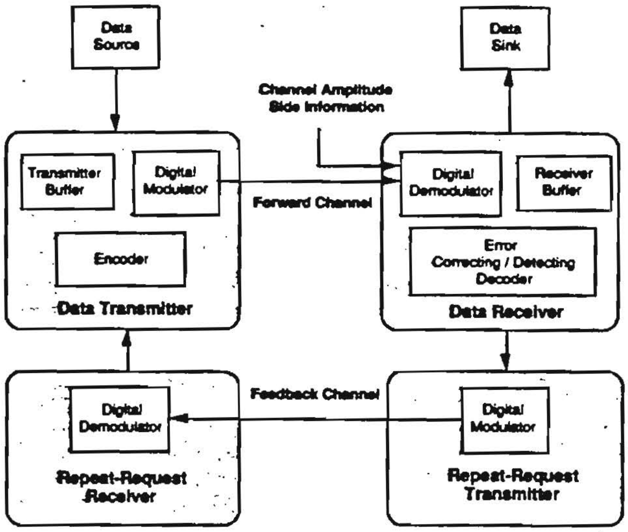

Fig. 1. Block diagram for a generalized ARQ error control system.

In the following section, a Reed–Solomon hybrid-ARQ (RS/HARQ) error control system is described in detail. This RS/HARQ system is an extension of the system described in an earlier paper [1]. The new system allows for the use of erasure decoding, providing a significant performance improvement when used in applications involving fading channels.

In the final section, examples are used to illustrate various aspects of the performance of the RS/HARQ system over fading channels with feedback.

II. The Performance of Nonbinary Codes Over Fading Channels

The amplitude of the forward channel is defined as a Rayleigh random variable a with the probability density function

$$p_a(a) = 2ae^{-a^2}. \tag{1}$$

The forward channel is also corrupted by additive white Gaussian noise (AWGN) with one-sided power spectral density N_0. It is assumed throughout the rest of this paper that

Manuscript received November 26, 1990; revised May 31, 1991 and October 8, 1991. This work was supported by National Science Foundation Grant NCR-9009877.

The author is with the School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

IEEE Log Number 9106590.

[1] This assumption can be readily removed by inserting the results of the analysis in this paper into a graph theoretical framework that accounts for noise on the feedback channel. See, for example, Lu and Chang, "The effect of return channel errors on the performance of ARQ protocols," in *Proc. 1991 IEEE Int. Symp. Inform. Theory*, p. 221.

0018-9545/92$03.00 © 1992 IEEE

channel phase variations due to multipath fading are detected and removed during the demodulation process. Methods for realizing such performance through pilot tone techniques are discussed in [4] and [5]. It is also assumed that channel fading is frequency-nonselective.

To determine the performance of nonbinary codes over this channel, one must first select a modulation format and derive expressions for the probabilities of channel symbol error and erasure. There are two basic methods for generating channel symbol erasures that are distinguished by the existence or nonexistence of forward channel amplitude side information at the receiver; both are considered in the following analysis.

Let the channel symbol alphabet have cardinality 2^b and the code symbol alphabet have cardinality[2] 2^m. Nonbinary codes provide a level of burst error correction for fading channels that is a function of the amount by which m exceeds b (it is assumed that the symbol transmission rate is greater than the fade rate). The rationale is that a deep fade may affect several consecutive channel symbols while only affecting a few code symbols, thus allowing for the use of higher rate codes. It follows that this burst error correcting capability is lost if the channel symbols are individually interleaved (as opposed to their being interleaved in clusters, each cluster corresponding to a code symbol).

The channel symbol error and erasure expressions developed in this section are translated into code symbol error and erasure expressions for both the code symbol and channel symbol interleaved channels. An example is provided to demonstrate that nonbinary codes provide better performance when used in conjunction with code symbol interleaving as opposed to channel symbol interleaving.

A. Erasure Generation Without Side Information

If no channel amplitude information is available, the modem signal space is partitioned into several nonerasure (reliable) decision regions and an erasure (unreliable) decision region. The declaration of erasures by the receiver is then solely a function of the position of the received signal within the signal space. Consider the case of the 8-PSK modulation format [6]. Fig. 2 shows how the signal space is partitioned into eight nonerasure decision regions $\{\Lambda_0, \Lambda_1, \cdots, \Lambda_7\}$ and an erasure region Λ_s. In the general case an n-dimensional signal space Λ with $(2^b + 1)$ decision regions $\{\Lambda_s, \Lambda_0, \cdots, \Lambda_{2^b-1}\}$ is considered. Suppose that the channel symbol corresponding to the decision region Λ_i is transmitted. A conditional probability density function $p(z|a, \Lambda_i)$ is derived for a fixed channel amplitude a and a received signal $z = (z_0, z_1, \cdots, z_{n-1})$. This conditional pdf can be obtained by inverting the product of the characteristic functions for the Gaussian and Rayleigh processes defining the channel (e.g. the MPSK case is treated by Proakis in [7, appendix 7A]). The probability of channel symbol error as a function of channel amplitude is then

$$p_{ce}(a) = E_i\left\{\sum_{j\neq i}\left(\int_{\Lambda_j} p(z|a, \Lambda_i)\, dz\right)\right\} \tag{2}$$

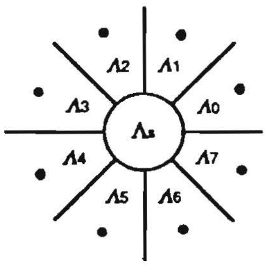

Fig. 2. Decision regions for the 8-PSK modulation format without side information [6].

where the expected value operation is taken over all possible transmitted signals i corresponding to reliable decision regions Λ_i. The probability of channel symbol erasure as a function of a is

$$p_{cs}(a) = E_i\left\{\int_{\Lambda_s} p(z|a, \Lambda_i)\, dz\right\}. \tag{3}$$

It is important to note that when side information is unavailable, erasures can be caused by the AWGN process as well as the channel fading process.

B. Erasure Generation with Side Information

If it is assumed that side information containing the exact value of the channel amplitude a is available, a simpler approach to erasure generation can be considered. Hagenauer and Lutz [8] have examined the case in which erasure generation is based solely on the value of a, eliminating the impact of the AWGN process on the probability of channel and code symbol erasures. Let λ_s be the erasure threshold. A received channel symbol shall be declared an erasure any time the channel amplitude is less than λ_s. The probability of this occurring is

$$p_{cs}^{SI} = \int_0^{\lambda_s} p_a(a)\, da. \tag{4}$$

The derivation of an expression for the probability of channel symbol error is also quite simple. Once the channel amplitude is known for a received channel symbol and it has been determined that the amplitude is not below the erasure threshold, the decision regions for the demodulator are scaled to match the channel amplitude and the symbol decision is made. The 8-PSK decision regions for the side information case are shown in Fig. 3. The signal scaling caused by the fading channel amplitude is indicated. Due to the radial symmetry of the signal constellation in this example, the decision regions do not change with the value of a. This is not the case, however, with nonconstant envelope modulation formats (e.g., ASK and QAM). The probability of channel symbol error for channel amplitude a and a given modulation format is obtained by taking the standard AWGN symbol error rate expression and weighting the signal energy by a^2. For example, the probability of bit error for coherent BPSK, which is used extensively in

[2] The following presentation can be readily modified for use with channel and/or code symbol alphabets whose cardinalities are not a power of two.

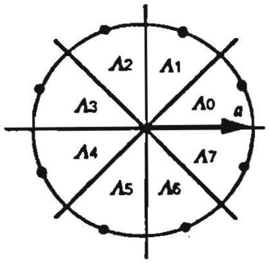

Fig. 3. Decision regions for the 8-PSK modulation format with side information.

the examples in a later section, is

$$p_{ce}^{SI}(a) = \frac{1}{2}\,\mathrm{erfc}\left(a\sqrt{\frac{RE_b}{N_0}}\right) \tag{5}$$

where R is the rate of the code in use.

C. Code Symbol Errors and Erasures on Code Symbol Interleaved Slowly Fading Channels

In the literature frequent reference is made to "fast" and "slowly" fading channels. These terms can be understood only when taken in relation to something; for example, the channel symbol transmission rate. It is assumed here that a slowly fading channel exhibits fades whose duration exceeds the time required to transmit several channel symbols. A fade affecting one channel symbol is thus highly likely to affect temporally adjacent channel symbols. The techniques used to combat this correlative effect are one of the concerns of this paper. Two interleaving techniques are considered: interleaving at the *channel* symbol level and interleaving at the *code* symbol level.

The analysis of the code symbol interleaved slowly fading channel is predicated on the following assumptions.

- Channel amplitude is constant over the m/b consecutive channel symbols constituting a code symbol.
- The channel symbols are interleaved to infinite depth in clusters corresponding to individual code symbols. The noise processes affecting adjacent code symbols within a code word are thus uncorrelated.

In this case interleaving is being used to eliminate the correlation of the noise/fading process affecting adjacent symbols in a received code word, but not that between adjacent channel symbols comprising a single received code symbol.

A code symbol erasure occurs whenever one or more of the m/b constituent channel symbols are declared to be erasures. The derivation of the code symbol erasure probabilities for both channel symbol erasure generation mechanisms is straightforward. For the case where side information is not available and the channel amplitude is a constant a, the probability of there being at least one erased channel symbol among m/b channel symbols is

$$p_s(a) = 1 - (1 - p_{cs}(a))^{m/b}. \tag{6}$$

The code symbol erasure probability p_s is then obtained through an expected value operation using the probability density function for a. For the case with side information, the probability of code symbol erasure is determined solely by the value of a, which is constant during the transmission of the code symbol. The probability of symbol erasure in this case is simply the probability that the value of a at any given moment is below the erasure threshold λ_s. The following expressions result (see (7) below):

A code symbol error occurs whenever the code symbol has not been declared an erasure and one or more of the constituent channel symbols is in error. For the case when side information is not available, the code symbol error probability for constant channel amplitude a is

$$p_e(a) = (1 - p_{cs}(a))^{m/b} - (1 - p_{ce}(a) - p_{cs}(a))^{m/b}. \tag{8}$$

The code symbol error probability is then obtained through an expected value operation. The corresponding result for the case with side information is similar, though the limits of integration rule out the possibility of a channel symbol erasure, simplifying the integrand. The probability of code symbol error in both cases is (see (9) below):

D. Code Symbol Errors and Erasures on Channel Symbol Interleaved Slowly Fading Channels

The analytical ground rules are changed considerably for the case of channel symbol interleaving. The following assumptions are made.

- The channel amplitude is constant over the time required to transmit one or more channel symbols.

$$p_s = \begin{cases} \int_0^\infty \left[1 - (1 - p_{cs}(a))^{m/b}\right] p_a(a)\, da, & \text{no side information} \\ p_{cs}^{SI}, & \text{side information.} \end{cases} \tag{7}$$

$$p_e = \begin{cases} \int_0^\infty \left[(1 - p_{cs}(a))^{m/b} - (1 - p_{ce}(a) - p_{cs}(a))^{m/b}\right] p_a(a)\, da, & \text{no side information} \\ \int_{\lambda_s}^\infty \left[1 - (1 - p_{ce}^{SI}(a))^{m/b}\right] p_a(a)\, da, & \text{side information.} \end{cases} \tag{9}$$

- The channel symbols are interleaved to infinite depth. The noise processes affecting adjacent channel symbols are thus uncorrelated.

From the decoder's perspective, the channel symbol interleaved channel appears to fade more rapidly than the code symbol interleaved channel, which in turn fades more rapidly than a stationary channel. It should be noted, however, that for both interleaved channels, *the physical propagation medium may be the same:* a Rayleigh fading channel whose fade duration exceeds the time required for the transmission of several channel symbols. It is the method used to format the coded information prior to transmission that differs between the code symbol and channel symbol interleaved cases. Consider, for example, a mobile radio channel that is used for the transmission of data. Variations in vehicle velocity cause the physical channel fade frequency and duration to vary. In almost all applications, however, the fading is slow with respect to the channel symbol transmission rate. Assuming sufficient depth, channel (code) symbol interleaving ensures that a given fade does not affect adjacent channel (code) symbols. The *effective* fade rate seen by the receiver after deinterleaving thus appears to be faster than the channel (code) symbol transmission rate.

The expected value operations of the previous section are performed at the channel symbol level instead of the code symbol level for the case of the channel symbol interleaved channel. Consider first the probability of channel symbol erasure for the case without side information. The expected value of the channel erasure probability is computed using the probability density function for a as follows:

$$p_{cs} = \int_0^\infty p_{cs}(a) p_a(a)\, da. \tag{10}$$

For the case with side information, the probability of channel symbol erasure is again the probability that the channel amplitude during the bit transmission time is less than λ_s.

$$p_{cs}^{SI} = \int_0^{\lambda_s} p_a(a)\, da. \tag{11}$$

In both cases the probability of code symbol erasure is the probability that, among m/b consecutive channel symbols, at least one symbol is erased. The probability of code symbol erasure for both cases is then

$$p_s = \begin{cases} 1 - (1 - p_{cs})^{m/b}, & \text{no side information} \\ 1 - \left(1 - p_{cs}^{SI}\right)^{m/b}, & \text{side information.} \end{cases} \tag{12}$$

The probability of channel symbol error for the case without side information is the expected value of the channel error probability for fixed channel amplitude a.

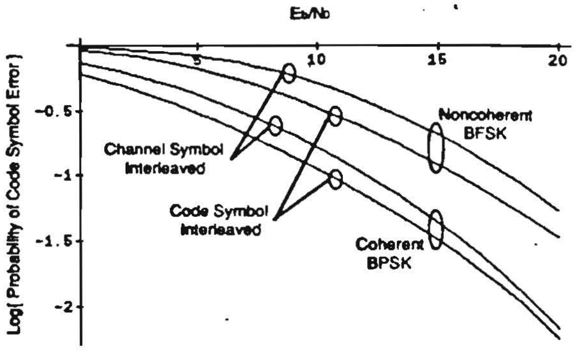

Fig. 4. Probability of symbol error curves for eight-bit code symbols.

$$p_{ce} = \int_0^\infty p_{ce}(a) p_a(a)\, da. \tag{13}$$

The probability of code symbol error is then the probability that none of the channel symbols is erased and at least one channel symbol is in error. For the case with side information, the limits of integration are changed to reflect the erasure threshold.

$$p_{ce}^{SI} = \int_{\lambda_s}^\infty p_{ce}^{SI}(a) p_a(a)\, da. \tag{14}$$

The probability of code symbol error for both cases is thus

$$p_e = \begin{cases} (1 - p_{cs})^{m/b} - (1 - p_{cs} - p_{ce})^{m/b}, & \text{no side information} \\ \left(1 - p_{cs}^{SI}\right)^{m/b} - \left(1 - p_{cs}^{SI} - p_{ce}^{SI}\right)^{m/b}, & \text{side information.} \end{cases} \tag{15}$$

E. Comparing Channel Symbol and Code Symbol Interleaving

From the standpoint of a random nonbinary error control code, the probabilities of error and erasure for the symbols entering the decoder completely determine reliability performance. The following figures thus focus on code symbol error and erasure probabilities. These can in turn be translated directly into word error rates using the development in Section III. It should be noted here, however, that the word error rate is a monotonically increasing function of the channel symbol error rate. A reduction in the latter will thus ensure a reduction in the former.

Fig. 4 compares the code symbol error probabilities for the channel and code symbol interleaved channels. These curves assume eight-bit code symbols transmitted using a coherent BPSK modem and a noncoherent BFSK modem. Both modems are assumed to have channel amplitude side information. These curves indicate an increase in the probability of symbol error that corresponds to an effective reduction in E_b/N_0 of 1 to 3 dB when eight-bit code symbols are interleaved bit-by-bit instead of symbol-by-symbol. In the examples in Section IV it will be shown that this reduction has a substantial impact on the decoder error probability for Reed–Solomon codes.

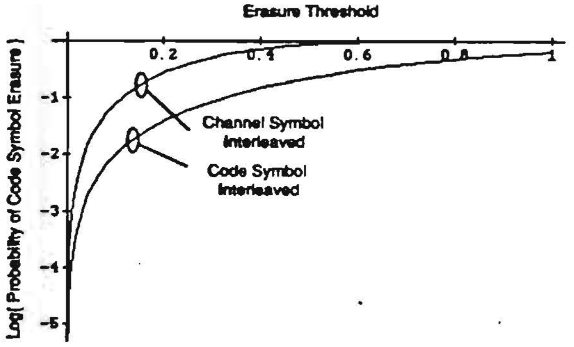

Fig. 5. Probability of symbol erasure curves for eight-bit code symbols.

Fig. 5 compares the probabilities of code symbol erasure for the channel and code symbol interleaved channels as a function of the erasure threshold. As in Fig. 4, the transmission of eight-bit code symbols over a binary modem with side information is assumed. The probability of code symbol erasure is substantially higher for the channel symbol interleaved channel. Equations (7), (9), (12), and (15) can be used in deriving the performance of an arbitrary nonbinary code so long as appropriate reliability and throughput expressions as a function of code symbol erasure and error probabilities are known. In the next section such expressions will be derived for the RS/HARQ error control system.

III. Reed–Solomon Hybrid-ARQ Protocols

A type-I hybrid-ARQ protocol encodes data packets for both error detection and error correction. The decoder uses the error correction capability to correct the most frequently occurring error patterns, while the residual detection capacity is used to detect less frequently occurring patterns. In the event of the latter, or a decoder failure, a retransmission request is sent back to the transmitter via the feedback channel. This form of hybrid-ARQ protocol can be realized using two codes, one for error correction, the other for error detection. It is also possible to perform both the correction and detection functions using a single encoder/decoder pair. The resulting system is, in general, easier to implement for in most cases it is equivalent in complexity and power usage to the error correcting portion of the two codec system. In this paper a single codec Reed–Solomon hybrid-ARQ system is discussed that requires virtually no increase in complexity over the corresponding FEC Reed–Solomon codec.

In an earlier paper [1] a method was demonstrated for modifying FEC Reed–Solomon error control systems for use in type-I hybrid-ARQ protocols. In this section the earlier method is extended to allow for erasure decoding. The resulting Reed–Solomon Hybrid-ARQ protocol (RS/HARQ) is then shown to provide good reliability performance at the expense of a negligible to moderate reduction in throughput within well defined E_b/N_0 operating regions.

One of the most widely used decoding techniques for Reed–Solomon codes is the Berlekamp–Massey algorithm (BMA). The BMA provides bounded distance decoding and is readily extended to provide erasure decoding.

Given a Reed–Solomon code with minimum distance $d_{\min}$, this algorithm can correct all received words containing e symbol errors and s symbol erasures within the constraint $(2e+s) < d_{\min}$. The BMA uses an iterative approach to generate an error location polynomial whose roots indicate the positions of the errors in the received word. An erasure location polynomial is then generated using the erasure locations indicated by the demodulator. Both polynomials are used in conjunction with an error/erasure magnitude polynomial to compute valid symbol values for the erroneous and erased coordinates in the received word. The details of this process are described in a variety of places in the literature (e.g., [3], [9]–[11]). For the purpose of this paper it is only important to note the following. If the received word is within e errors and s erasures of a valid code word and $(2e+s) < d_{\min}$, then the decoder will output that code word. If the selected code word is not the code word that was transmitted, then a decoder error has occurred. If there is no code word within e errors and s erasures, where $(2e+s) < d_{\min}$, then a decoder failure is declared. In the event that a code word is selected by the decoder, the values of e and s can be obtained by examining the degrees of the error and erasure location polynomials respectively.

The underlying principle of the RS/HARQ protocol is that for a given completed decoding operation, the probability that the decoder has made a mistake is proportional to the values of e and s. Reliability performance can thus be increased by reducing the number of errors and erasures to be corrected below some threshold and requesting a retransmission of the code word whenever the threshold is exceeded or a decoder failure is declared.

This simple idea is readily reduced to a realizable form. Let d_e be defined as the *effective diameter* of the RS/HARQ decoder. The effective diameter is the maximum value of the sum $(2e+s)$ for which decoding will be completed. d_e must thus be an integer in the range $[0, d_{\min}-1]$. Whenever $(2e+s) > d_e$, or any time a decoder failure occurs, a retransmission will be requested. d_e thus defines the balance between error correction and error detection in the RS/HARQ system.

Fig. 6 shows the channel model that will be used in the following performance analysis. The expressions for computing p_e and p_s were derived in the previous section for both the code symbol and channel symbol interleaved Rayleigh fading channels. (In the nonerasure decoding case, p_s is set to zero in the model.) It is assumed that incorrect code symbols are equally probable.[3]

The reliability and throughput performance of the RS/HARQ system can be determined through a series of combinatorial exercises. Consider the case of an $(n, k, d_{\min})$ Reed–Solomon

[3] Since a complete weight enumerator (i.e., one that states how many times each nonzero symbol occurs in each code word) has not yet been found for the RS codes, some type of simplifying assumption is necessary. The worst case approach sets the probability of occurrence for each incorrect symbol equal to p_e, the probability of symbol error. This is overly pessimistic, for 1) a fade affecting one channel symbol is likely to corrupt the other channel symbols comprising the code symbol (code symbol interleaved case), and 2) given the large number of low weight nonzero code words, the nonzero code symbol distribution in the most likely error events is approximately uniform.

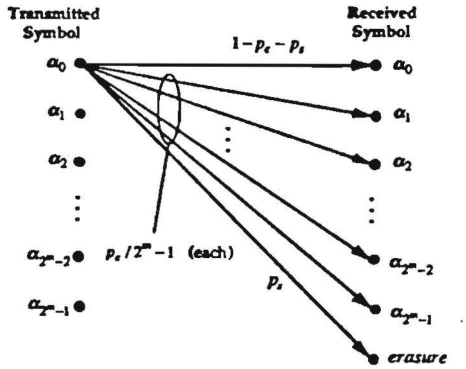

Fig. 6. Channel model for the RS/HARQ system with erasure decoding.

code used in the RS/HARQ protocol. Since Reed–Solomon codes are linear, one may assume without loss of generality that the all-zero code word has been transmitted. Let $P_{d_e}^j$ be the probability that a received word is within the decoding sphere of effective diameter d_e surrounding a code word of weight j. If simple error correction is to be performed without erasure decoding, $P_{d_e}^j$ takes on the value

$$P_{d_e}^j = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{d_e-2v}{2} \rfloor} \binom{n-j}{v}\binom{j}{w}(2^m-1)^{w-j} \cdot \left(1-\frac{p_e}{2^m-1}\right)^w (1-p_e)^{n-j-v} p_e^{j+v-w}. \quad (16)$$

If erasure decoding is used, $P_{d_e}^j$ takes on the value

$$P_{d_e}^j = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \sum_{x=0}^{\lfloor \frac{d_e-2v-w}{2} \rfloor} \sum_{y=0}^{d_e-2v-w-2x} \sum_{z=0}^{\lfloor d_e-2v-w-2x-y \rfloor} \cdot \binom{n-j}{v}\binom{n-j-v}{w}\binom{j}{x}\binom{j-x}{y}\binom{j-x-y}{z} \cdot (2^m-2)^x(2^m-1)^{y+z-j} p_e^{j+v-y-z} p_s^{w+y} \cdot (1-p_e-p_s)^{n+z-j-v-w}. \quad (17)$$

Both (16) and (17) are derived in the Appendix. The weight distribution of Reed–Solomon codes is known to be [9]

$$A_j = \binom{n}{j}(2^m-1)\sum_{i=1}^{j-d_{\min}} (-1)^i \binom{j-1}{i} 2^{m(j-i-d_{\min})}. \quad (18)$$

The probability that the decoder with effective diameter d_e will make a decoder error is thus

$$P_E = \sum_{j=1}^{n} A_j P_{d_e}^j. \quad (19)$$

A retransmission request will be generated whenever the received word is not within the decoding sphere surrounding the correct or any one of the incorrect code words. For the nonerasure and erasure decoding cases the following expressions result (see (20) below):

The probability of word error P_{WE} for packets sent to the data sink is a function of both the decoder error probability P_E and the probability of retransmission P_R. This is due to the fact that multiple transmissions of the same packet allow the decoder multiple opportunities to make a mistake. It is shown in [2] that the probability of word error among accepted packets is

$$P_{\mathrm{WE}} = \frac{P_E}{(1-P_R).} \quad (21)$$

The throughput for the RS/HARQ system is a function of the retransmission protocol selected. If a selective-repeat protocol is used, as is assumed in the following examples, then the throughput can be shown to be [2]

$$\eta = R(1-P_R) \quad (22)$$

where R is the rate of the code in use.

IV. Performance Examples

In the following examples the performance of the RS/HARQ system is examined for a coherent BPSK modem used over a code symbol interleaved slowly fading Rayleigh channel. The performance of the RS/HARQ system over a channel symbol interleaved channel is degraded because of the 1 to 2 dB decrease in the effective E_b/N_0 (see Fig. 4). It is assumed that side information is available for the declaration of erased channel symbols.

A. The Impact of Erasure Decoding

The first set of examples considers the impact of a variation in the erasure threshold λ_s. In Fig. 7–9 the performance of a (16, 12) Reed–Solomon code in the RS/HARQ system is examined. Fig. 7 shows the variation in the probability of word error P_{WE} as the erasure threshold is increased. It is clear that a significant amount of improvement in reliability performance can be obtained through erasure decoding in the RS/HARQ system at medium to high E_b/N_0, but very little improvement is obtained at smaller values of E_b/N_0.

It should be noted that these word error rate curves and those that follow assume that the hybrid-ARQ protocol allows for an unlimited number of retransmission attempts. As a result, the word error rate in all cases approaches unity as the signal to noise ratio is reduced. The performance of a system that employs retry limits can only be understood through comparison of the word error rate curves with the throughput

$$P_R = \begin{cases} 1-P_E-\sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \binom{n}{v} p_e^v (1-p_e)^{n-v}, & \text{nonerasure decoding} \\ 1-P_E-\sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \binom{n}{v}\binom{n-v}{w}(1-p_e-p_s)^{n-v-w} p_e^v p_s^w, & \text{erasure decoding.} \end{cases} \quad (20)$$

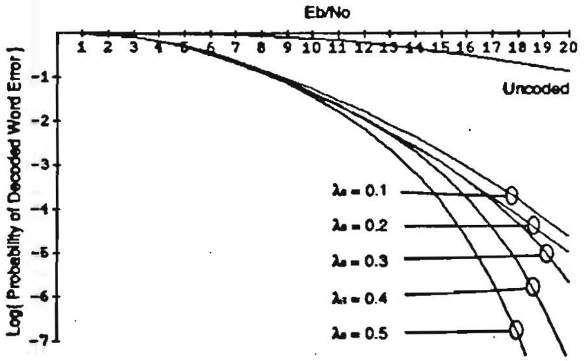

Fig. 7. Probability of word error for a (16, 12) Reed–Solomon code in the RS/HARQ system with $d_e = 4$.

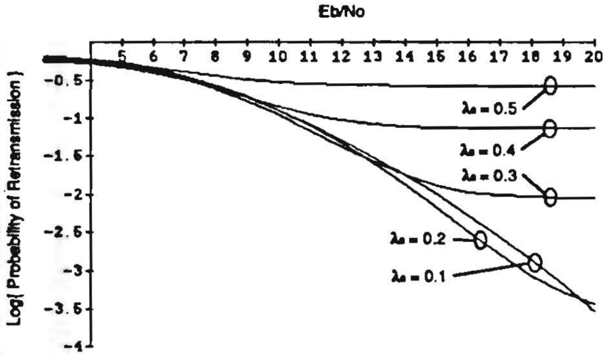

Fig. 8. Probability of retransmission for a (16, 12) Reed–Solomon code in the RS/HARQ system with $d_e = 4$.

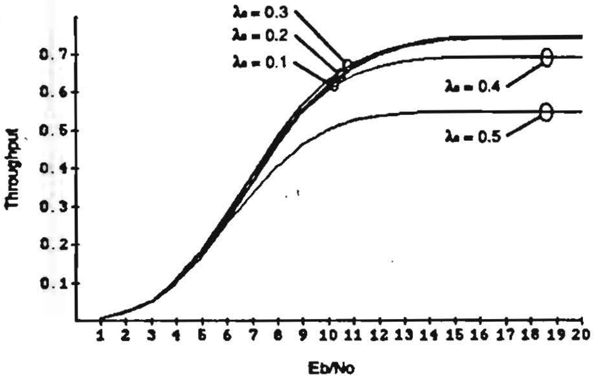

Fig. 9. Throughput for a (16, 12) Reed–Solomon code in the RS/HARQ system with $d_e = 4$.

curves that follow. It is then seen that for extremely low signal-to-noise ratios the throughput is essentially zero, so that given a reasonable retry limit, it is highly improbable that erroneous words are accepted by the decoder.

Fig. 8 shows the probability of retransmission P_R for the (16, 12) code as a function of E_b/N_0 and the erasure threshold λ_s. The floor effect is due to the fact that the probability of erasure is independent of the signal to noise ratio on the channel. After a certain point, any further reduction in the probability of symbol error is negated by the probability that the number of erasures alone will be sufficient to exceed the effective diameter of the decoder.

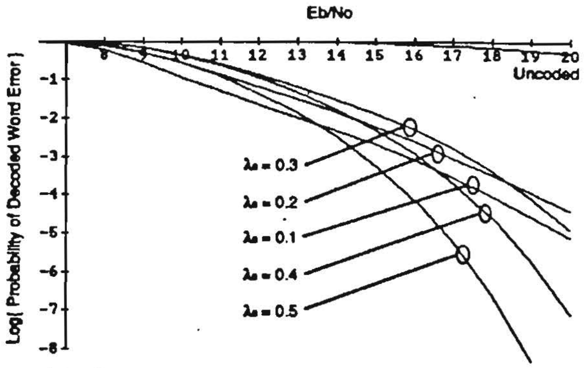

Fig. 10. Probability of word error for a (64, 56) Reed–Solomon code in the RS/HARQ system with $d_e = 8$.

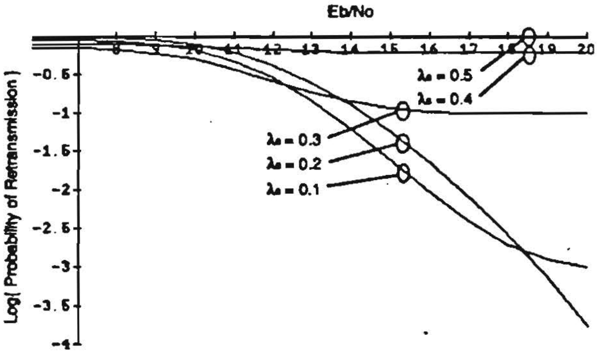

Fig. 11. Probability of retransmission for a (64, 56) Reed–Solomon code in the RS/HARQ system with $d_e = 8$.

Fig. 9 shows that, all else being equal, the RS/HARQ system throughput for the (16, 12) Reed–Solomon code is only affected by large values of λ_s. This is apparent in Fig. 8, for though there are floors on the probability of retransmission curves, the floors themselves are at relatively low values for P_R. P_R must take on values above 0.1 before any appreciable reduction in throughput can be seen.

Figs. 10–12 show the performance of the (64, 56) Reed–Solomon code in the RS/HARQ system with an effective diameter of eight. Fig. 10 shows the same asymptotic trends seen in Fig. 7, but also shows that higher values of the erasure threshold can actually cause performance to degrade at low signal to noise ratios. The reason for this is seen in Fig. 11: for the cases $\lambda_s = 0.4$ and 0.5 the probability of word error is being significantly increased by the large number of attempts required before the packet is accepted by the receiver (note the denominator in (21)). As E_b/N_0 increases, however, a point is reached beyond which the higher erasure thresholds provides better reliability performance than the lower erasure thresholds.

Fig. 12 shows the drastic reduction in throughput caused by the high P_R floors in Fig. 11. It is interesting to note that in the middle range of E_b/N_0, the throughput curves for the higher erasure thresholds is temporarily better than that for the lower values. This is because the probability of symbol error is reduced in this erasure generation system by increasing the range of channel amplitudes which will cause erasures. This

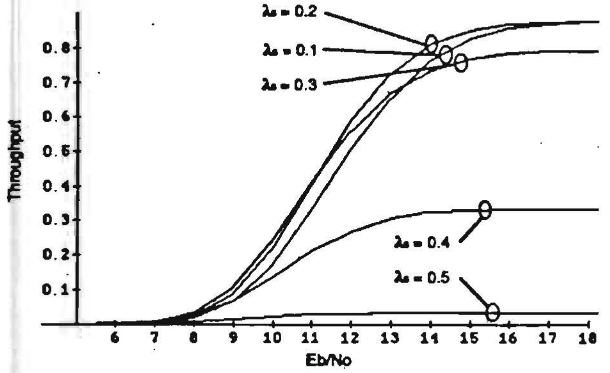

Fig. 12. Throughput for a (64,56) Reed–Solomon code in the RS/HARQ system with $d_e = 8$.

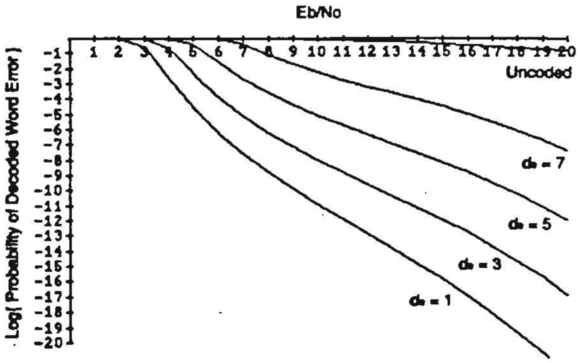

Fig. 13. Word error rate for a (64,56) Reed–Solomon code in the RS/HARQ system with $\lambda_s = 0.1$.

effect is not seen elsewhere in the curves because it is masked by other factors.

B. The Impact of the Type-I Hybrid-ARQ Protocol

In this section examples are provided to show the impact of the setting of the effective diameter d_e in the RS/HARQ protocol. Fig. 13 shows several curves depicting the word error rate for the (64,56) RS/HARQ system with an erasure threshold $\lambda_s = 0.1$ and various values of d_e. The impact of the conversion of the FEC BMA to a hybrid-ARQ protocol is readily apparent. Throughout the operating range of the decoder, each incremental reduction in d_e results in a reduction of several orders of magnitude in the word error rate. Fig. 14 shows the word error rate for the same values of d_e when the erasure threshold has been increased to $\lambda_s = 0.4$. In this case the reliability performance is even more sensitive to a decrease in the effective diameter of the decoder. Figs. 15 and 16 show the price paid for the improvement in reliability performance; the throughput drops significantly in the middle of the operating range with each successive decrease in d_e. It should be noted that for the case $\lambda_s = 0.1$, the setting of d_e does not affect the asymptotic throughput ceiling established by the erasure threshold (recall Fig. 12). For the case $\lambda_s = 0.4$, however, the successive reductions in d_e cause large reductions in the throughput ceiling. In this case the average number of erasures seen per received packet is approaching the effective diameter of the decoder, causing repeated retransmission requests.

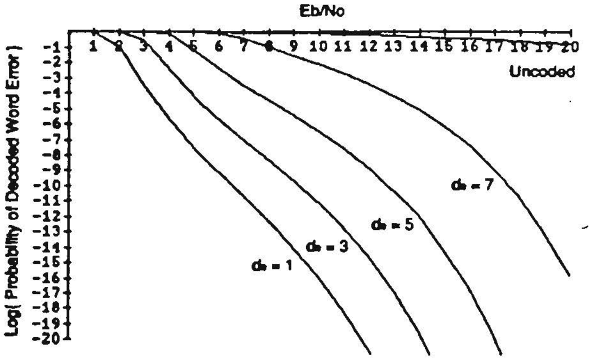

Fig. 14. Word error rate for a (64,56) Reed–Solomon code in the RS/HARQ system with $\lambda_s = 0.4$.

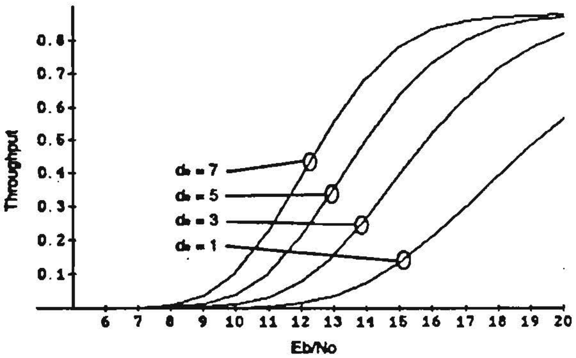

Fig. 15. Throughput for a (64,56) Reed–Solomon code in the RS/HARQ system with $\lambda_s = 0.1$.

V. Conclusion

Expressions for the probabilities of channel symbol error and erasure were derived for slowly fading Rayleigh channels. Two methods for generating channel symbol erasures were considered, the two being differentiated by the existence or nonexistence of forward channel amplitude side information. The resulting probability expressions were then translated into code symbol error and erasure probability expressions for use in evaluating the performance of nonbinary error control codes over code symbol and channel symbol interleaved channels. It was shown that code symbol interleaving offers significantly better symbol error and erasure performance than channel symbol interleaving.

The RS/HARQ error control system for fading channels with feedback was then presented. This system is based on the use of Reed–Solomon codes with erasure decoding in a type-I hybrid-ARQ protocol. The RS/HARQ system requests retransmissions whenever a received word falls outside of the decoding sphere defined by the effective diameter. It was shown that this system provides a substantial improvement in reliability performance at the expense of a reduction in throughput. In conjunction with a variable threshold erasure generation system, the RS/HARQ system provides an extremely powerful and flexible means for controlling errors on a Rayleigh fading channel.

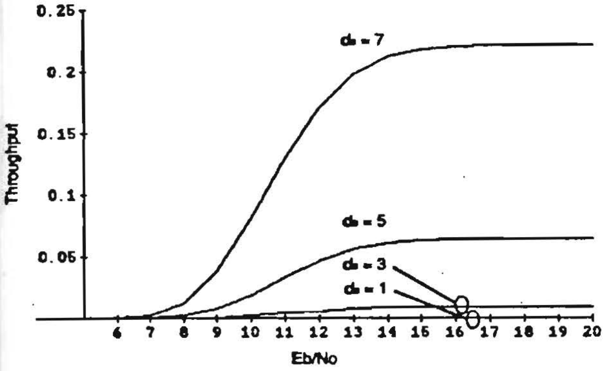

Fig. 16. Throughput for a (64,56) Reed–Solomon code in the RS/HARQ system with $\lambda_s = 0.4$.

APPENDIX
DERIVATION OF (16) AND (17)

Equation (17) shall de derived first, followed by (16). We wish to determine $P^j_{d_e}$, the probability that a received word R falls within a decoding sphere of effective diameter d_e surrounding a code word C_j of weight j. It shall be assumed that the all-zero code word has been transmitted. It is also assumed that sufficient code symbol interleaving has been employed to render the communication channel memoryless. Let p_0 be the probability of correct symbol reception (i.e., a zero symbol is received), p_e the probability of incorrect symbol reception (a zero symbol is received), and p_s the probability of symbol erasure.

Let θ be the set containing the $(n-j)$ coordinates of C_j that contain zeros. Let ϕ be the set containing the j coordinates of C_j that contain nonzero symbols. The desired probability expression can be obtained by allocating e errors and s erasures among the two sets of coordinates in all possible combinations within the constraint $(2e + s) \leq d_e$.

There are five distinct events that must be accounted for in this derivation.

- A θ or ϕ coordinate in R contains a zero symbol. This event occurs with probability $p_0 = (1 - p_s - p_e)$.
- A θ or ϕ coordinate in R contains an erasure. This event occurs with probability p_s.
- A θ coordinate in R contains a nonzero symbol. This event occurs with probability p_e.
- A ϕ coordinate in R contains the same nonzero symbol as at the same coordinate in C_j. Since it is assumed that erroneous code symbols occur with equal probability, this event occurs with probability $p_e/(2^m - 1)$.
- A ϕ coordinate in R contains a nonzero symbol that is different from the symbol at the same coordinate in C_j. This event occurs with probability $p_e(2^m - 2)/(2^m - 1)$.

The allocation of errors and erasures in the expression is controlled by five counting variables as follows.

v number of θ coordinates for which R has a nonzero symbol
w number of θ coordinates for which R has an erasure
x number of ϕ coordinates in which R has a nonzero symbol other than the nonzero symbol in C_j
y number of ϕ coordinates in which R has an erasure
z number of ϕ coordinates in which R has a zero.

$P^j_{d_e}$ is computed by summing over all possible error/erasure patterns for the all-zero code word such that $(v + w + x + y + z) \leq d_e$. Using the substitution $p_0 = (1 - p_e - p_s)$, the following results.

$$
\begin{aligned}
P^j_{d_e} = &\left\{ \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \binom{n-j}{v} p_e^v \sum_{w=0}^{d_e - 2v} \binom{n-j-v}{w} p_s^w \right. \\
&\left. \cdot (1 - p_e - p_s)^{n-j-v-w} \right\} \left\{ \sum_{x=0}^{\lfloor \frac{d_e - 2v - w}{2} \rfloor} \binom{j}{x} \right. \\
&\cdot \left[\left(\frac{2^m - 2}{2^m - 1} \right) p_e \right]^x \sum_{y=0}^{d_e - 2v - w - 2x} \binom{j-x}{y} p_s^y \sum_{z=0}^{\lfloor \frac{d_e - 2v - w - 2x - y}{2} \rfloor} \\
&\left. \cdot \binom{j-x-y}{z} (1 - p_e - p_s)^z \left(\frac{p_e}{2^m - 1} \right)^{j-x-y-z} \right\} \\
= &\sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e - 2v} \sum_{x=0}^{\lfloor \frac{d_e - 2v - w}{2} \rfloor} \sum_{y=0}^{d_e - 2v - w - 2x} \sum_{z=0}^{\lfloor \frac{d_e - 2v - w - 2x - y}{2} \rfloor} \\
&\cdot \binom{n-j}{v} \binom{n-j-v}{w} \binom{j}{x} \binom{j-x}{y} \binom{j-x-y}{z} \\
&\cdot (2^m - 2)^x (2^m - 1)^{y+z-j} p_e^{j+v-y-z} p_s^{w+y} \\
&\cdot (1 - p_e - p_s)^{n+z-j-v-w}.
\end{aligned}
$$

The case without erasures is obtained in a similar manner. Once again the coordinates in C_j are partitioned into two sets: θ containing the $(n-j)$ zero coordinates and ϕ containing the j nonzero coordinates. The probabilities of the following events are used in the derivation.

- A θ coordinate in R contains a zero symbol. This event occurs with probability $p_0 = (1 - p_e)$
- A θ coordinate in R contains a nonzero symbol. This event occurs with probability p_e.
- A ϕ coordinate in R contains the same nonzero symbol as at the same coordinate in C_j. This event occurs with probability $p_e/(2^m - 1)$.
- A ϕ coordinate in R contains a zero symbol or a nonzero symbol that is different from the symbol at the same coordinate in C_j. This event occurs with probability $1 - p_e/(2^m - 1)$.

Let the counting variable v denote the number of coordinates that differ between the received word and the code word. The counting variable w shall control the allocation of the v differences between the sets ϕ and θ. Using the probabilities of the events listed above, the following expression results:

$$
\begin{aligned}
P^j_{d_e} = &\sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{d_e - 2v}{2} \rfloor} \left\{ \binom{n-j}{v} p_e^v (1 - p_e)^{n-j-v} \right\} \\
&\cdot \left\{ \binom{j}{w} \left(1 - \frac{p_e}{2^m - 1} \right)^w \left(\frac{p_e}{2^m - 1} \right)^{j-w} \right\}
\end{aligned}
$$

$$= \sum_{v=0}^{\lfloor \frac{d_c}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{d_c - 2v}{2} \rfloor} \binom{n-j}{v} \binom{j}{w} (2^m - 1)^{w-j} \cdot \left(1 - \frac{p_e}{2^m - 1}\right)^w (1 - p_e)^{n-j-v} p_e^{j+v-w}.$$

References

[1] S. B. Wicker, "High reliability data transfer over the land mobile radio channel using interleaved hybrid-ARQ error control," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 48–55, Feb. 1990.

[2] S. Lin, D. J. Costello Jr., and M. J. Miller, "Automatic-repeat-request error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5–17, Dec. 1984.

[3] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[4] F. Davarian, "Mobile digital communications via tone calibration," *IEEE Trans. Veh. Technol.*, vol. VT-36, pp. 55–62, May 1987.

[5] J. P. McGeehan and A. J. Bateman, "Phase-locked transparent tone-in-band (TTIB): A new spectrum configuration particularly suited to the transmission of data over SSB mobile radio networks," *IEEE Trans. Commun.*, vol. COM-32, pp. 81–87, Jan. 1984.

[6] C. Schlegel and D. J. Costello, Jr., "Bandwidth efficient coding for fading channels: Code construction and performance analysis," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1356–1368, Dec. 1989.

[7] J. G. Proakis, *Digital Communications*, 2nd ed. New York: McGraw-Hill, 1989.

[8] J. Hagenauer and E. Lutz, "Forward error correction coding for fading compensation in mobile satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 215–225, Feb. 1987.

[9] E. Berlekamp, *Algebraic Coding Theory*, revised ed. Laguna Hills, CA: Aegean Park Press, 1984.

[10] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122–127, Jan. 1969.

[11] G. Clark and J. Cain, *Error-Correction Coding for Digital Communications*. New York: Plenum, 1981, p. 22.

Stephen B. Wicker (S'83–M'83) was born in Hazelhurst, MS, on September 25, 1960. He received the B.S.E.E. degree with high honors from the University of Virginia, Charlottesville, in 1982, the M.S.E.E. degree from Purdue University, West Lafayette, IN, in 1983, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1987.

From 1983 through 1987 he was a subsystem and system engineer with the Space and Communications Group of the Hughes Aircraft Company, El Segundo, CA. In September 1987 he joined the faculty of the School of Electrical Engineering at the Georgia Institute of Technology, where he currently holds the title of Associate Professor. From June 1991 until March 1992 he served as the Academic Coordinator for Georgia Tech–Lorraine in Metz, France. His current research interests center on the development of algorithms for error control, data compression, and data security for digital communication systems. He is the inventor or co-inventor of several adaptive error control algorithms for nonstationary communication channels, many of which are now seeing commercial application. He has served as a consultant in these areas for several telecommunications companies in the United States, Canada, and France. He is the author of over 45 technical publications.

Dr. Wicker is a member of the IEEE Communications, Information Theory, and Vehicular Technology Societies. He is also a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and Omicron Delta Kappa. He was named a Visiting Fellow of the British Columbia Advanced Systems Institute in 1992.

Type-II Hybrid-ARQ Protocols Using Punctured MDS Codes

Stephen B. Wicker[†]
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Michael Bartz
Department of Electrical Engineering
Memphis State University
Memphis, Tennessee 38152

Abstract

MDS codes possess several properties that make them an ideal choice for type-II hybrid-ARQ protocols. These properties include "strong separability", "strong invertibility", and excellent reliability performance when used for simultaneous error detection and correction. In this paper these properties are shown to lead to a natural definition of an MDS type-II hybrid-ARQ protocol. An (n, k) MDS code is decomposed into a pair of $(n/2, k)$ punctured MDS codes. The original code and the two derivative codes are used individually in type-I hybrid-ARQ protocols. These three type-I protocols combine to form a single type-II protocol. The performance of this system is analyzed in detail, with particular attention paid to the definition of an effective channel model for code words that are known to have caused the generation of retransmission requests.

1 Introduction

Maximum distance separable (MDS) codes provide excellent reliability performance when used for error detection or combined error detection and error correction [1]. They are thus natural candidates for use in hybrid-ARQ protocols [2], in which error detection and

[†] This work was supported by the National Science Foundation under Grant Number NCR-9009877.

correction are combined in a receiver that detects unreliably decoded code words and requests their retransmission. It has been noted, however, that the combined error detection and correction capabilities of MDS codes can become a liability when the communication channel is nonstationary [3]. Consider a type-I hybrid-ARQ protocol that has been designed for a fixed channel noise level. In a type-I protocol each transmitted code word is encoded for both error detection and error correction. The error correction capacity is used to correct frequently occurring error patterns, while the detection capacity is used to detect the less frequently occurring patterns, which cause the generation of retransmission requests. In a type-I protocol the transmitter responds to retransmission requests by sending another copy of the transmitted code word. This error control scheme performs quite well on channels that are essentially stationary except for infrequent bursts of additional noise. However, if the channel noise level deviates from the design level for a significant period of time, the performance of the protocol can be seriously degraded. As the channel noise level increases, the probability that each received word contains an uncorrectable error pattern also increases. The new error patterns are detected, and a flood of retransmission requests ensues that persists until the channel noise level returns to its original level. As the channel noise level decreases, the error correction capacity is sufficient to correct all error patterns, rapidly driving the frequency of retransmission requests to zero. The redundancy reserved for error detection thus assumes the status of useless overhead. In either case the throughput performance of the type-I protocol is suboptimal. This problem is particularly acute when the type-I protocol is based on codes whose error correction and detection performance curves have strongly negative slopes as a function of channel noise (e.g. MDS codes).

Code combining offers a solution to this problem. The code combining receiver concatenates received code words until their combined code rate is sufficient to reliably recover the transmitted information [4]. As the channel noise level varies, the receiver varies the effective code rate of the error control system, reducing the throughput degradation observed with a fixed-rate system. The simplest code combining system is the type-II hybrid-ARQ protocol, a truncated form that limits combining operations to a maximum of two received code words [5],[6],[7]. In a type-II protocol the transmitter responds to an initial retransmission request by transmitting a code word containing parity bits for the first code word. The original message is obtained through decoding operations on the first or second code words alone, or through a combined decoding operation on the composite code word created through the concatenation of the two received code words.

MDS codes possess a number of properties that make them well suited for use in type-II protocols. Mandelbaum [8], [9] has noted that Reed-Solomon codes (members of the MDS family) can be punctured to provide a primary code word and one or more secondary blocks that provide incremental redundancy as needed. This scheme is optimal in the sense that the incremental redundancy increases the minimum distance of the composite received word by the greatest possible amount per additional symbol. This paper modifies and extends Mandelbaum's work by defining a type-II hybrid-ARQ protocol based on the general class of punctured MDS codes.

Pursley and Sandberg have proposed the use of Reed-Solomon codes in an incremental redundancy system for meteor-burst channels [10], [11]. In this paper we consider their version of the RS type-II system as well as a modified version with fewer decoding operations. The analytical framework presented here can be used to accurately predict the performance of both systems.

In Section 2 an analysis of the reliability and throughput performance of a generic type-II hybrid-ARQ protocol is provided. A general review of the relevant properties of MDS codes follows. It is then shown that the various properties of MDS codes can be used to construct a type-II protocol from a series of type-I protocols based on punctured MDS codes. The performance parameters of the individual type-I protocols provide the necessary data for the complete characterization of the type-II system using the general expressions derived in the earlier section. Several examples are provided to indicate the excellent throughput and reliability performance offered by the MDS type-II hybrid-ARQ protocol.

2 Performance Model for the General Type-II Hybrid-ARQ Protocol

In a type-II hybrid-ARQ protocol, a code word is encoded using two codes, $\mathbf{C}_1$ and $\mathbf{C}_2$, to create a pair of code words $\mathbf{c}_1$ and $\mathbf{c}_2$. These codes have corresponding decoding operations D_1 and D_2 which can recover the original code word from noise corrupted versions of $\mathbf{c}_1$ and $\mathbf{c}_2$ respectively. $\mathbf{c}_1$ comprises the initial transmission while $\mathbf{c}_2$ is set aside. Upon

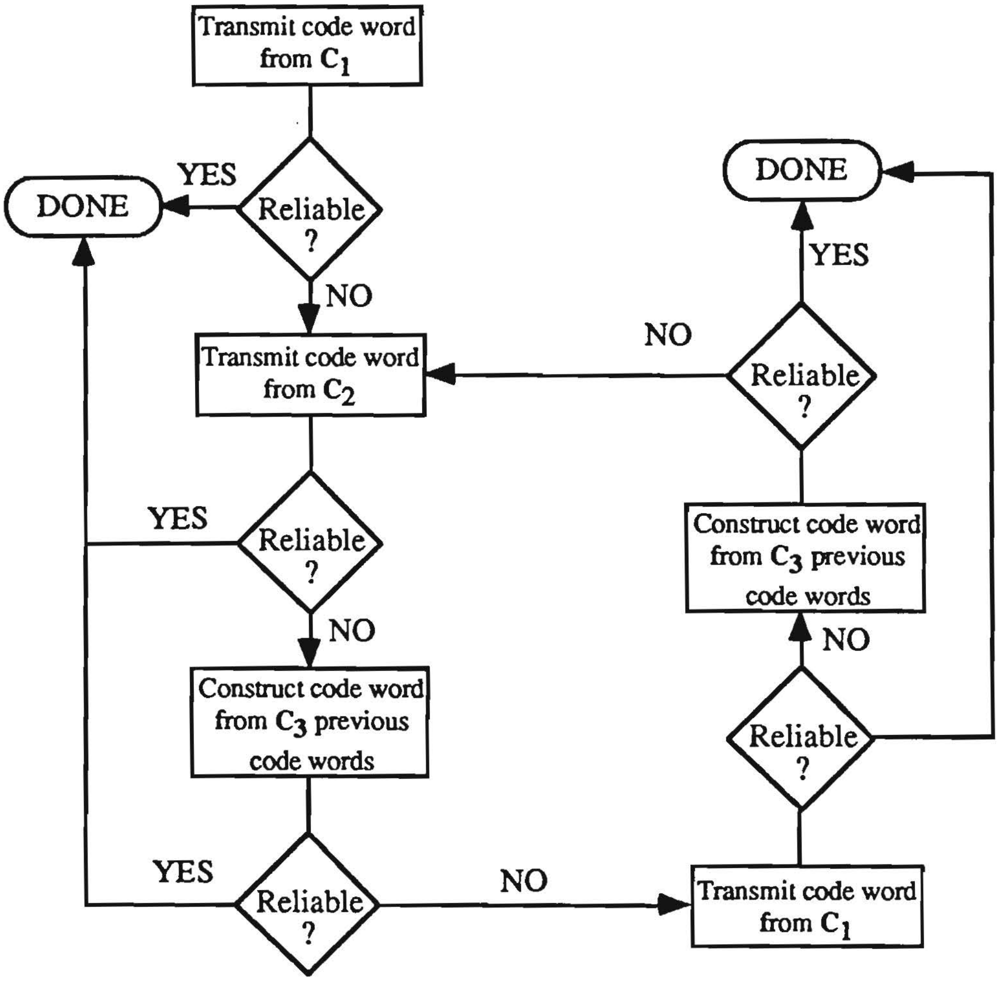
Transmit code word from C_1
DONE
YES
Reliable ?
NO
Transmit code word from C_2
YES
Reliable ?
NO
Construct code word from C_3 previous code words
YES
Reliable ?
NO
Transmit code word from C_1
Reliable ?
NO
Construct code word from C_3 previous code words
Reliable ?
YES
DONE
NO

Figure 1: Flowchart for a type-II hybrid-ARQ protocol

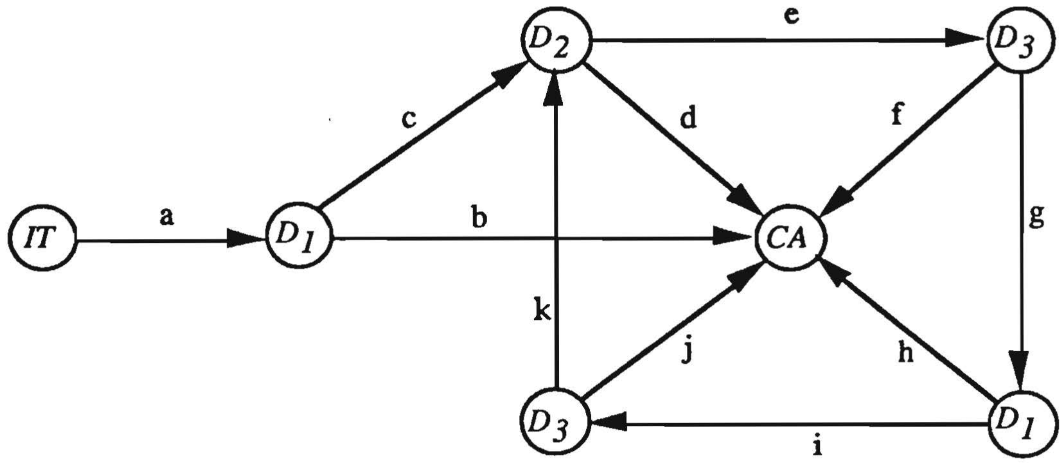

Figure 2: Generic graph for a type-II hybrid-ARQ protocol

receiving $\mathbf{c}_1$, the receiver attempts decoding operation D_1 to recover the transmitted data. If the attempt is successful (i.e. no retransmission request is generated), the receiver sends an acknowledgement (ACK) to the transmitter; otherwise, a negative acknowledgement (NACK) is sent. The transmitter responds to the NACK by sending $\mathbf{c}_2$. The receiver then attempts to decode $\mathbf{c}_2$ by itself using decoding operation D_2. If successful, the receiver inverts the corrected version of $\mathbf{c}_2$ to recover the desired information and sends an ACK to the receiver. If unsuccessful, the receiver combines $\mathbf{c}_1$ and $\mathbf{c}_2$ to create $\mathbf{c}_3$, a code word in a lower rate code $\mathbf{C}_3$. If the third decoding operation D_3 is successful, the data is recovered and an ACK is sent to the receiver; otherwise, a NACK is sent and the entire process is repeated. After the first pair of transmissions, the combined decoding operation D_3 is always available after the receipt of subsequent copies of either $\mathbf{c}_1$ or $\mathbf{c}_2$. This decoding protocol is shown as a flowchart in Figure 1. Note that this protocol is a slight generalization of the type-II protocol originally presented by Lin and Yu [5].

Reliability and throughput generating functions for this protocol are obtained using signal flow graph techniques [12]. A generic graph that describes this protocol is depicted in Figure 2. The nodes of the graph consist of the initial transmission IT, code word acceptance CA, and the decoding operations D_1, D_2, and D_3. The branches indicate the directions by which the code word transmission and decoding processes proceed. For reliability analysis the branches are labeled with the probability that the associated

Branch label	Throughput label	Reliability label
a	T^{n_1}	1
b	$1 - p_R^{(1)}$	$p_E^{(1)}$
c	$p_R^{(1)} \cdot T^{n_2}$	$p_R^{(1)}$
d	$1 - p_R^{(2)}$	$p_E^{(2)}$
e	$p_R^{(2)}$	$p_R^{(1)}$
f	$1 - p_R^{(3)}$	$p_E^{(3)}$
g	$p_R^{(3)} \cdot T^{n_1}$	$p_R^{(3)}$
h	$1 - p_R^{(1)}$	$p_E^{(1)}$
i	$p_R^{(2)}$	$p_R^{(1)}$
j	$1 - p_R^{(3)}$	$p_E^{(3)}$
k	$p_R^{(3)} \cdot T^{n_2}$	$p_R^{(3)}$

Table 1: Graph labels for the derivation of throughput and reliability generating functions

event occurs, whereas for throughput calculations, the branch labels also help determine the number of transmitted symbols. The branch labels for determining reliability and throughput are found in Table 1. The generic graph yields the following transfer function:

$$IT = \left\{ ab + ac\,(d + ef + egh + egij) \left(\frac{1}{1 - egik} \right) \right\} CA \tag{1}$$

Substituting the appropriate branch values, one obtains the throughput and reliability generating functions for the type-II protocol.

For throughput calculation the branches are labeled with the probabilities of the generation of a retransmission request p_R, decoder error p_E, and code word acceptance, $1 - p_R$, as appropriate. The superscripts for the various probabilities reference the probabilities to a specific decoding operation. The variable T is used to indicate the transmission of a code word, while its superscript denotes the number of code word symbols contained in the code word (either n_1 or n_2 for code $\mathbf{C}_1$ or $\mathbf{C}_2$ respectively). For the calculation of the throughput, a selective repeat protocol is assumed. The throughput generating function is as follows:

$$
\begin{aligned}
G(T) \;&=\; T^{n_1} p_C^{(1)} + T^{n_1+n_2} p_R^{(1)} \Big[p_C^{(2)} + T^{n_1} p_R^{(2)} p_R^{(3)} p_C^{(1)} + \\
&\qquad p_C^{(3)} \left(p_R^{(2)} + T^{n_1} p_R^{(2)} p_R^{(3)} p_R^{(1)} \right) \Big] \left(\frac{1}{1 - T^{n_1+n_2} p_R^{(1)} p_R^{(2)} {p_R^{(3)}}^2} \right) \qquad (2)
\end{aligned}
$$

Once the throughput generating function has been obtained, the throughput η of the protocol can be computed. The throughput η is defined here as the ratio of the number of information symbols transmitted (k, the dimension of code $\mathbf{C}_1$) to the average number of symbols transmitted before the code word is correctly accepted. By taking the partial derivative of Equation (2) with respect to T and setting T equal to unity, the probability of each of the distinct paths through the graph in Figure 2 is weighted by the total number of symbols transmitted along that path. The following expression results.

$$
\begin{aligned}
\eta \;&=\; k \left(\frac{\partial}{\partial T} G(T) \Big|_{T=1} \right)^{-1} \\
&=\; k \cdot \left\{ \frac{1 - p_R^{(1)} p_R^{(2)} {p_R^{(3)}}^2}{n_1 + n_2 p_R^{(1)} + n_1 p_R^{(1)} p_R^{(2)} p_R^{(3)} - n_1 p_R^{(1)} p_R^{(2)} {p_R^{(3)}}^2} \right\} \qquad (3)
\end{aligned}
$$

The reliability generating function provides the following expression for the probability that an accepted, decoded code word contains one or more symbol errors.

$$
P(E) = \left\{ p_E^{(1)} + p_R^{(1)} \left(p_E^{(2)} + p_R^{(2)} p_E^{(3)} + p_R^{(2)} p_R^{(3)} p_E^{(1)} + p_R^{(1)} p_R^{(2)} p_R^{(3)} p_E^{(3)} \right) \left(\frac{1}{1 - p_R^{(1)} p_R^{(2)} {p_R^{(3)}}^2} \right) \right\} \qquad (4)
$$

3 The Properties of MDS Codes

The use of MDS codes in type-II protocols is motivated by a series of properties that are unique to MDS codes. The most pertinent are listed here. The first property is frequently used as the definition for MDS codes, though it can be shown to be equivalent to a number of other definitions.

Property 1 The Singleton Bound states that given a linear code **C** with length n and dimension k, the minimum distance $d_{\min}$ must satisfy $d_{\min} \leq (n - k + 1)$. A code **C** is MDS if and only if it satisfies the Singleton Bound with equality [13].

Property 1 shows that MDS codes are optimal in the sense that they provide "maximum distance" between code words. MDS codes were once called "optimal codes", but this proved to be confusing and was abandoned in later literature [14], [15]. The Singleton Bound can be used in conjunction with the BCH bound to show that Reed-Solomon codes are MDS [15].

When the "natural" length of a particular code is unsuitable for an application, the length can be changed by puncturing, extending, shortening, or lengthening the original code [14], [15], [16]. In this paper the technique of interest is *puncturing*. A code is punctured through the consistent deletion of parity coordinates from each code word in the code. Puncturing j coordinates reduces an (n, k, d) code to an $(n - j, k, d')$ code. In most cases the goal is to minimize the reduction in minimum distance through the judicious selection of the deleted coordinates. In the case of MDS codes, however, the minimum distance of the resulting code is solely a function of the number of coordinates punctured. Any combination of j puncturing operations changes an $(n, k, n-k+1)$ MDS code into an $(n - j, k, n - k - j + 1)$ MDS code.

Property 2 Punctured MDS codes are MDS.

This is easily proven by noting that the elimination of a coordinate in a code can reduce the code's minimum distance by at most one, while the Singleton Bound implies that the minimum distance of an MDS code must be reduced by at least one when the length is reduced by one. The result, of course, is a consistent reduction of the minimum distance by one with each successive puncturing operation.

Property 1 can be shown to imply a "separability" property which proves quite useful in the development of code combining schemes [13]. Unfortunately the term "separable" has enjoyed a variety of definitions in the literature that are not equivalent. In works related to MDS codes "separable" is taken to mean that a code can be partitioned (separated) into message symbols and parity symbols (i.e. a systematic representation of the code exists) [13], [15]. In this sense of the word, any linear code is separable, for a generator matrix **G** for an (n, k) code must have at least one combination of k linearly independent columns.

In works involving type-II hybrid-ARQ protocols, however, "separable" has been used to describe any code $\{F(x)\}$ for which there is a punctured version $\{f(x)\}$ that is "capable of detecting by itself a number of errors eventually correctable by $\{F(x)\}$" [17]. In this paper the former definition is adopted, for it is this sense of separability that leads to the construction of the desired MDS code combining protocol. An (n, k) code shall be called *strongly separable* if *any* k code word coordinates can be used as the information symbols in a systematic representation.

Property 3 MDS codes are strongly separable [13].

A code is said to be invertible if the parity-check symbols of the code word can be used by themselves to uniquely determine the information symbols through an inversion process [18]. An (n, k) code shall be called *strongly invertible* if any k symbols from the code word can be used to recover the information symbols.

Property 4 MDS codes are strongly invertible.

This property is proved in Appendix B.

The final property of interest is the MDS weight enumerator, which allows for an exact determination of the probabilities of undetected error and retransmission request.

Property 5 The number of code words of weight j in an $(n, k, d_{\min})$ 2^m-ary MDS code is [14]

$$A_j = \binom{n}{j}(2^m - 1)\sum_{i=0}^{j-d_{\min}} (-1)^i \binom{j-1}{i} 2^{m(j-i-d_{\min})}. \tag{5}$$

4 Punctured MDS Codes in a Type-II Hybrid-ARQ Protocol

In the MDS type-II protocol, the codes $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{C}_3$ are formed in a very natural manner. The first step is to select an (n, k) MDS code with rate less than one-half for the combined code $\mathbf{C}_3$. Using decoding operation D_3, this code should provide sufficient error correction capability for the reliable transmission of information under the worst channel conditions expected. Figure 3 shows how code words from $\mathbf{C}_3$ are punctured to form code words in $\mathbf{C}_1$ and $\mathbf{C}_2$. The first $n/2$ coordinates in a given $\mathbf{C}_3$ code word $\mathbf{c}_{i,3}$ form the code word $\mathbf{c}_{i,1}$ in $\mathbf{C}_1$, while the remaining $n/2$ coordinates form the code word $\mathbf{c}_{i,2}$ in $\mathbf{C}_2$. Since codes $\mathbf{C}_1$ and $\mathbf{C}_2$ are punctured versions of the MDS code $\mathbf{C}_3$, they are themselves MDS by Property 2. Property 4 guarantees that corrected versions of code words from any of the three codes can be used to recover the information symbols. Decoding operations D_1 and D_2 are designed so as to maximize throughput while maintaining a minimum allowable level of reliability under optimum channel conditions. The design of the individual decoding operations is developed in the following section.

5 A Retransmission Request Mechanism for MDS Codes

All three of the decoding operations used in a type-II protocol need a retransmission request mechanism to detect code words whose completed decoding will result in unreliable information symbols. In the MDS type-II scheme, the same retransmission request mechanism is used with all three decoding operations. All three are treated as *type-I* hybrid-ARQ protocols that combine to form a *type-II* protocol. In this section the design and analysis of the MDS type-I hybrid-ARQ protocol is discussed.

5.1 The MDS Type-I Hybrid-ARQ Protocol

In earlier papers a method was demonstrated for modifying FEC Reed-Solomon error control systems for use in type-I hybrid-ARQ protocols [2], [19, fading channels with erasure decoding]. These discussions are easily generalized for application to bounded distance decoders for MDS codes.

Given an MDS code with minimum distance $d_{\min}$, a bounded distance decoding algorithm can correct all received words containing e symbol errors and s symbol erasures within the constraint $(2e+s) < d_{\min}$. If the received word is within e errors and s erasures of a valid code word and $(2e+s) < d_{\min}$, then the decoder will select that code word. If the selected code word is not the code word that was transmitted, then a decoder error has occurred. If there is no code word within e errors and s erasures, where $(2e+s) < d_{\min}$, then a decoder failure is declared. If decoding is completed, the values of e and s can be obtained by comparing the received and corrected words (or, in the case of the Berlekamp-

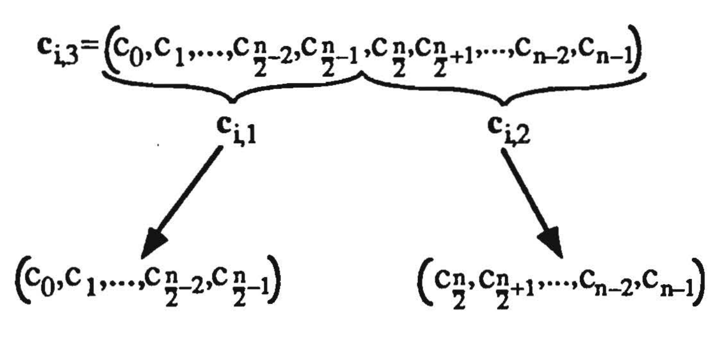
$c_{i,3}=(c_0,c_1,\ldots,c_{\frac{n}{2}-2},c_{\frac{n}{2}-1},c_{\frac{n}{2}},c_{\frac{n}{2}+1},\ldots,c_{n-2},c_{n-1})$
$c_{i,1}$
$c_{i,2}$
$(c_0,c_1,\ldots,c_{\frac{n}{2}-2},c_{\frac{n}{2}-1})$
$(c_{\frac{n}{2}},c_{\frac{n}{2}+1},\ldots,c_{n-2},c_{n-1})$

Figure 3: MDS code decomposition for type-II HARQ protocols: code word $\mathbf{c}_{i,3}$ is in the $(n, k, n-k+1)$ $\mathbf{C}_3$, code word $\mathbf{c}_{i,1}$ is in the punctured $(\frac{n}{2}, k, \frac{n}{2}-k+1)$ $\mathbf{C}_1$, and code word $\mathbf{c}_{i,2}$ is in the punctured $(\frac{n}{2}, k, \frac{n}{2}-k+1)$ $\mathbf{C}_2$.

Massey algorithm, by examining the degrees of the error and erasure locator polynomials respectively).

The bounded distance MDS type-I hybrid-ARQ protocol is defined as follows. Let d_e be defined as the *effective diameter* of the decoding operation. The effective diameter is the maximum value of the sum $(2e+s)$ for which decoding is allowed to be completed. The effective diameter d_e must thus be an integer in the range $[0, d_{\min}-1]$. Whenever $(2e+s) > d_e$, or any time a decoder failure occurs, a retransmission is requested. The effective diameter d_e thus defines the balance between error correction and error detection in this type-I hybrid-ARQ system.

5.2 The Performance of the MDS Type-I Protocols Within the Framework of a Type-II Protocol

When deriving the performance of a type-II protocol, two different categories of decoding operations must be considered: those operating on newly arrived code words and those operating on code words that have caused the generation of retransmission requests. Decoding operations D_1 and D_2 fall into the former category, D_3 falls into the latter. The rationale for this distinction lies in the fact that the average number of errors and erasures in the code word(s) to be decoded differs between the two cases. If a code word is known to have caused the generation of a retransmission request, then the expected number of errors and erasures within the code word is higher than that for a newly received code

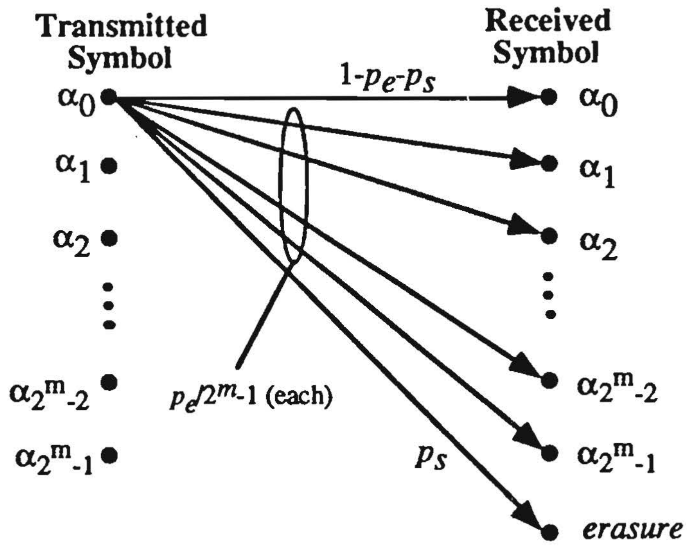

Figure 4: Channel model for the RS/HARQ system with erasure decoding

word for which decoding has not yet been attempted. An effective channel model must be developed for each of the two cases if the overall performance of the type-II protocol is to be accurately determined.

For decoding operations D_1 and D_2, the probabilities of symbol error and erasure are determined using information about the modulation format and the communication channel. Figure 4 shows the channel model used in the following analysis. This model assumes that transmitted code symbols are independent and that incorrect symbols are equally probable. The precise values for the probabilities of symbol error p_e and symbol erasure p_s are highly application dependent. For example, the case of the binary modem used over a slowly fading code symbol interleaved channel is treated in [19]. Once p_e and p_s are known, however, the following analysis can be used in most applications.

Using the values for p_e and p_s the probabilities of retransmission and decoder error are determined as follows. Consider the case of an (n, k) 2^m-ary MDS code in a bounded distance type-I hybrid-ARQ protocol. If only linear codes are being considered, one may assume without loss of generality that the all-zero code word has been transmitted. Let $P^j_{d_e}$ be the probability that a received word is within the decoding sphere of effective diameter d_e surrounding a code word of weight j. If simple error correction is to be performed without erasure decoding, $P_{d_e}^j$ takes on the value

$$P_{d_e}^j = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{d_e-2v}{2} \rfloor} \binom{n-j}{v} \binom{j}{w} (2^m-1)^{w-j} \left(1 - \frac{p_e}{2^m-1}\right)^w (1-p_e)^{n-j-v} p_e^{j+v-w} \tag{6}$$

This expression uses a series of counting variables to enumerate all possible received code words of length n that fall within the decoding sphere and weights them by their probability of occurrence using the channel model in Figure 4.

A similar expression can be obtained for those cases in which erasure decoding is used:

$$\begin{aligned} P_{d_e}^j \quad = \quad & \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \sum_{x=0}^{\lfloor \frac{d_e-2v-w}{2} \rfloor} \sum_{y=0}^{d_e-2v-w-2x} \sum_{z=0}^{\lfloor \frac{d_e-2v-w-2x-y}{2} \rfloor} \binom{n-j}{v} \binom{n-j-v}{w} \binom{j}{x} \binom{j-x}{y} \\ & \binom{j-x-y}{z} (2^m-2)^x (2^m-1)^{y+z-j} p_e^{j+v-y-z} p_s^{w+y} (1-p_e-p_s)^{n+z-j-v-w} \end{aligned} \tag{7}$$

Both Equations (6) and (7) are derived in [19].

Property 5 in Section 3 provides the weight distribution for MDS codes. If A_j is the number of code words of weight j, then the probability of undetected decoder error on a single code word transmission is

$$P_E = \sum_{j=d_{\min}}^{n} A_j P_{d_e}^j \tag{8}$$

A retransmission request will be generated whenever the received word is not within the decoding sphere surrounding the correct or any one of the incorrect code words. For the nonerasure and erasure decoding cases the following expressions result [19]:

$$P_R = \begin{cases} 1 - P_E - \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \binom{n}{v} p_e^v (1-p_e)^{n-v} & \text{nonerasure decoding} \\ 1 - P_E - \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \binom{n}{v} \binom{n-v}{w} (1-p_e-p_s)^{n-v-w} p_e^v p_s^w & \text{erasure decoding} \end{cases} \tag{9}$$

The value of P_E computed in Equation (8) is used for both $P_E^{(1)}$ and $P_E^{(2)}$ in Equations (3) and (4). The value of P_R computed in Equation (9) is used for both $P_R^{(1)}$ and $P_R^{(2)}$.

If a code word is known to have failed in decoding attempts by itself or in combination with other code words, then the mean number of code symbol errors and erasures in the code word is higher than that indicated by the channel model in Figure 4. The increase in the channel symbol erasure and error rate must be quantified if the performance of decoding operation D_3 is to be accurately computed. Consider the case of an (n, k) MDS code that has caused the generation of a retransmission request during a decoding attempt by a decoder with effective diameter d_e.

If the probability that a code symbol transmitted over a memoryless channel has been received in error is p_e, then the expected number of errors in a received code word $\mathbf{c}$ of length n is np_e before decoding. This new p_e can be written in terms of this expected value as follows

$$p_e = \frac{np_e}{n} = \frac{E\{\text{number of errors in } \mathbf{c}\}}{\text{length of } \mathbf{c}} \tag{10}$$

The expected value can also be computed by weighting the number of errors e by the sum of the probabilities of the error patterns of weight e and then summing over all possible values of e.

$$E\{\text{number of errors in } \mathbf{c}\} = \sum_{e=0}^{n} e \left[\binom{n}{e} p_e^e (1 - p_e)^{n-e} \right] \tag{11}$$

If it is assumed that the received code word $\mathbf{c}$ has caused the generation of a retransmission request, then $\mathbf{c}$ cannot have fallen within the decoding spheres of effective diameter d_e surrounding the correct and incorrect code words. Let Ω_0 be the summation of all terms in the above expression corresponding to error patterns that are contained within the decoding sphere surrounding the all-zero code word, i.e.,

$$\Omega_0 = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e - 2v} v \binom{n}{v} \binom{n-v}{w} (1 - p_e - p_s)^{n-v-w} p_e^v p_s^w. \tag{12}$$

Let Ω_1 be the summation of all terms in the above expression corresponding to error patterns that are contained within the decoding sphere surrounding nonzero code words. For code words of weight j define $\Omega_1(j)$ as

$$\Omega_1(j) = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \sum_{x=0}^{\lfloor \frac{d_e-2v-w}{2} \rfloor} \sum_{y=0}^{d_e-2v-w-2x} \sum_{z=0}^{\lfloor \frac{d_e-2v-w-2x-y}{2} \rfloor} (j+v-y-z)\binom{n-j}{v}$$
$$\binom{n-j-v}{w}\binom{j}{x}\binom{j-x}{y}\binom{j-x-y}{z}(2^m-2)^x(2^m-1)^{y+z-j}$$
$$p_e^{j+v-y-z}p_s^{w+y}(1-p_e-p_s)^{n+z-j-v-w}. \qquad (13)$$

The number of code words of weight j is known (Property 5), so Ω_1 can be computed as follows:

$$\Omega_1 = \sum_{j=d_{\min}}^{n} A_j\Omega_1(j) \qquad (14)$$

By removing the terms in Ω_0 and Ω_1 from the right hand side of Equation (11) and dividing the result by the probability of retransmission, the probability of symbol error within **c** given that **c** has caused the generation of a retransmission request can be obtained as

$$P(\text{code symbol error} \mid \text{request}) = p_e' = \frac{1}{nP_R}(np_e - \Omega_0 - \Omega_1) \qquad (15)$$

The value of P_R in the above expression is the probability of retransmission for the code word for its initial decoding attempt.

A similar result is obtained for the probability of symbol erasure given that a retransmission request has been generated. The above expressions are slightly modified to yield the following:

$$\Psi_0 = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} w\binom{n}{v}\binom{n-v}{w}(1-p_e-p_s)^{n-v-w}p_e^v p_s^w \qquad (16)$$

$$\Psi_1(j) = \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e-2v} \sum_{x=0}^{\lfloor \frac{d_e-2v-w}{2} \rfloor} \sum_{y=0}^{d_e-2v-w-2x} \sum_{z=0}^{\lfloor \frac{d_e-2v-w-2x-y}{2} \rfloor} (w+y)\binom{n-j}{v}\binom{n-j-v}{w}$$
$$\binom{j}{x}\binom{j-x}{y}\binom{j-x-y}{z}(2^m-2)^x(2^m-1)^{y+z-j}$$

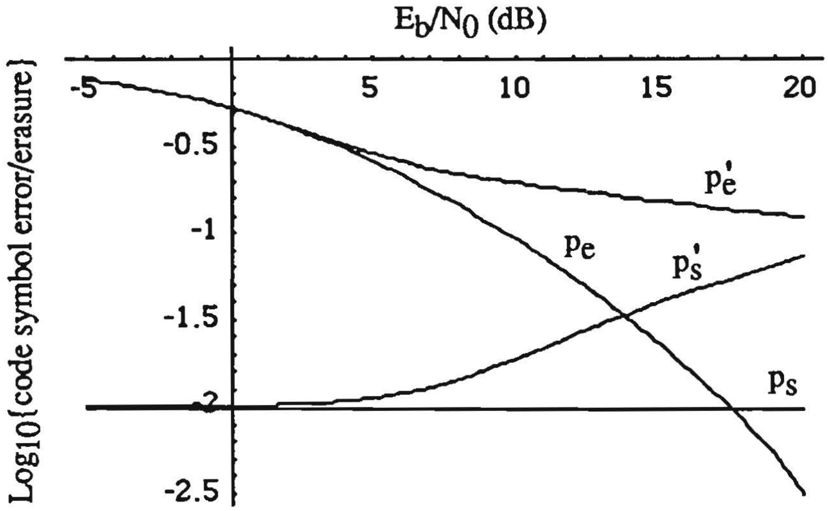

Figure 5: Probability of symbol error and erasure in code words that have caused the generation of retransmission requests

$$p_e^{j+v-y-z} p_s^{w+y} (1 - p_e - p_s)^{n+z-j-v-w} \tag{17}$$

$$\Psi_1 = \sum_{j=d_{\min}}^{n} A_j \Psi_1(j) \tag{18}$$

$$P(\text{code symbol erasure} \mid \text{request}) = p_s' = \frac{1}{nP_R}(np_s - \Psi_0 - \Psi_1) \tag{19}$$

Figure 5 indicates the necessity of the preceding analysis. It is assumed that a (32, 12) Reed-Solomon code has been decomposed into a pair of (16, 12) punctured Reed-Solomon codes. The initial decoding operation (D_1 or D_2) has an effective diameter of $d_e = 4$ and the combined operation uses $d_e = 20$. The 32-ary symbols are transmitted in bit-serial form using a coherent BPSK modem over a Rayleigh fading channel with background AWGN. Erasures are generated using channel side information with an erasure threshold of $\lambda_s = 0.1$ (assuming unity energy signaling) [19], [20]. Figure 5 clearly shows that the probability of symbol error in code words that are known to have caused the generation of a retransmission request (p_e') is substantially higher than that for newly arrived code words (p_e).

This preceding analysis can be carried through one additional step to account for code words that have caused the generation of retransmission requests in decoding operation D_3 as well. The additional increase in the probability of error is small compared to the initial increase indicated by the retransmission request generated during the first decoding attempt. Therefore, the reliability and throughput calculations are tight upper and lower bounds, respectively. As will be shown in Section 6, the additional computational complexity is thus not warranted in most cases.

All of the necessary probabilities are now available for the characterization of the performance of the MDS type-II hybrid-ARQ protocol. The probabilities of symbol error and erasure from Figure 4 (first attempt to decode) and Equations (15) and (19) (second and subsequent attempts to decode) are used in Equations (8) and (9) to determine the performance of the individual type-I hybrid-ARQ protocols. These performance parameters are then used in Equations (3) and (4) to determine the overall performance of the composite type-II protocol.

6 Examples

In this section several examples of the proposed protocol are examined. Additionally, the qualitative effects of the decoding sphere sizes are considered. This section concludes with consideration of a modification of the proposed protocol that reduces the complexity of the decoder. In the following examples code symbols are transmitted in bit-serial form using a coherent BPSK modem over a code symbol interleaved Rayleigh fading channel with background AWGN. Erasures are generated using channel amplitude side information [19],[20].

In the first set of performance curves (Figures 6 and 7), a (16, 4) MDS code (code $\mathbf{C}_3$) is decomposed into a pair of (8, 4) punctured MDS codes (codes $\mathbf{C}_1$ and $\mathbf{C}_2$). The original (16, 4) code and the punctured codes form a type-II HARQ protocol using the methods discussed in previous sections of this paper. Decoding operations D_1 and D_2 both have an effective diameter of $d_e = 2$, while D_3 has an effective diameter of $d_e = 12$. The performance of a type-I protocol ($d_e = 2$) based solely on one of the punctured (8, 4) codes has been included for reference. Figure 6 clearly shows that the type-II protocol offers substantially better throughput performance at lower signal to noise ratios. On a nonsta-

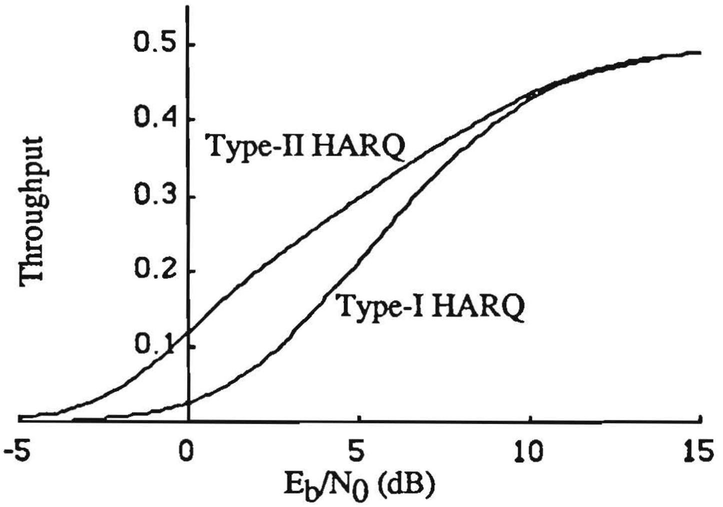

Figure 6: Throughput performance for (8,4)/(8,4) MDS type-II HARQ protocol compared to (8,4) MDS type-I HARQ protocol

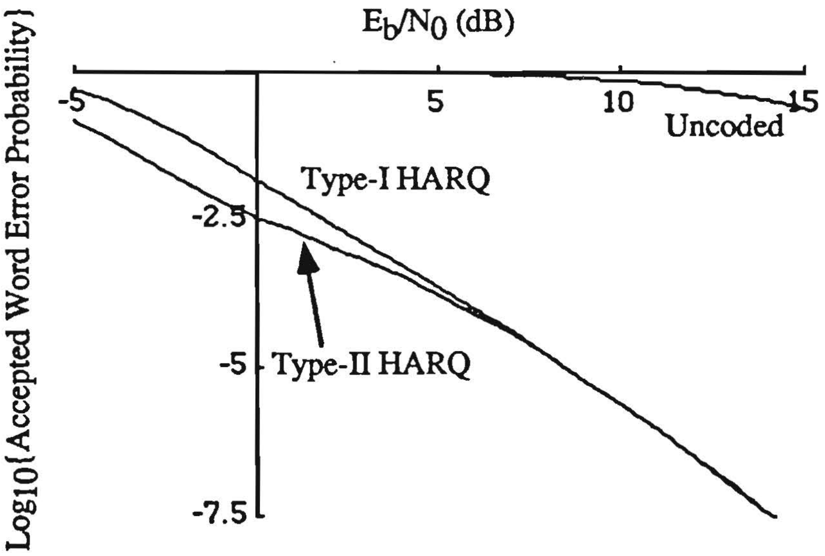

Figure 7: Reliability performance for (8,4)/(8,4) MDS type-II HARQ protocol compared to (8,4) MDS type-I HARQ protocol

tionary channel, the type-II protocol thus offers more graceful throughput degradation as the channel deteriorates. Figure 7 shows a improvement in reliability performance at low signal to noise ratios. This is a direct result of the reduction in the number of transmission attempts per code word in the type-II protocol (see the denominator in Equation (4)).

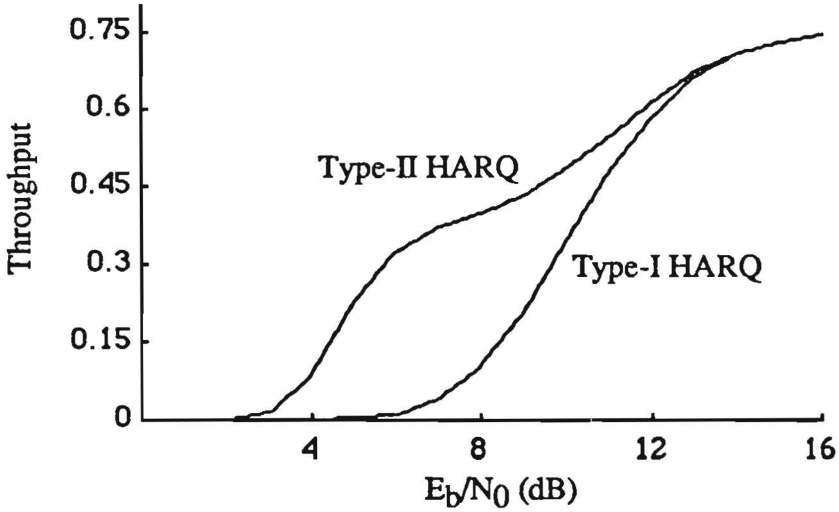

Figure 8: Throughput performance for (32,24)/(32,24) MDS type-II HARQ protocol compared to (32,24) type-I HARQ trotocol

In the next example a (64,24) MDS code (code C_3) is decomposed into a pair of (32, 24) punctured codes (codes C_1 and C_2). Decoding operation D_3 has an effective diameter of $d_e = 38$ while operations D_1 and D_2 have effective diameter of $d_e = 6$. The throughput data in Figure 8 indicates that the type-II protocol is still substantially better than the actual performance provided by the comparable type-I protocol. Also, as shown in Figure 9, the reliability of the type-II system is substantially better at low SNR's.

The effective diameter of the combined and single code affect both the reliability and throughput of the type-II system. Lower d_e reduces the size of the decoding spheres for the punctured codes. For a type-I system this reduction increases the reliability and decreases throughput. The combined diameter, d_{e3}, experiences the same affect. The optimal setting for d_e and d_{e3} is obviously a trade-off between reliability and throughput. For maximum throughput, the decoding diameters should be as large as possible. Consider

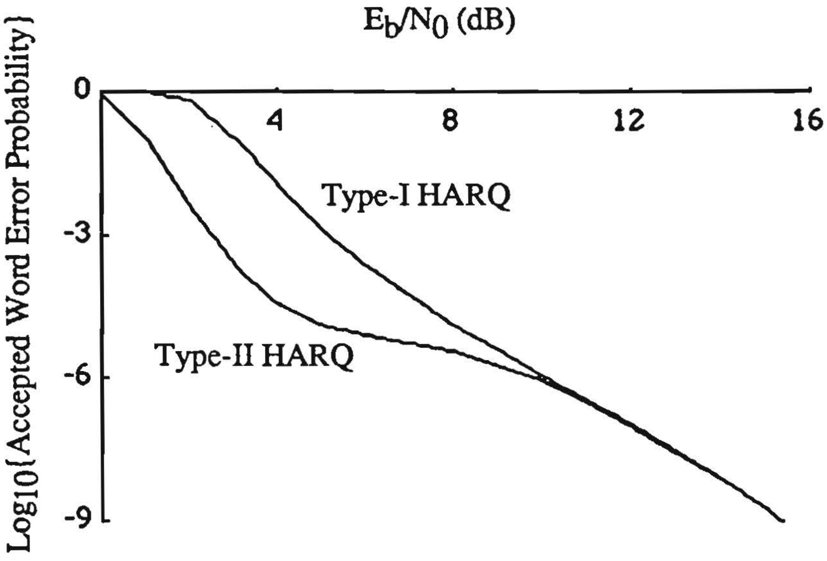

Figure 9: Reliability performance for (32,24)/(32,24) MDS type-II HARQ protocol compared to (32,24) type-I HARQ protocol

the throughput and reliability of the (8,4) system above. The reliability and throughput at SNR of 0 dB is summarized in Table 2. The first number in each entry is the throughput, while the second is the log of the accepted word error rate. The trade-off between reliability and throughput is very clear.

Finally, consider the implications of a slight modification to the proposed protocol. After a retransmission, let the newly received code word be combined directly with the previously determined unreliable code word. This step reduces the overall processing time of the decoder. The reader is referred to Appendix C for details of the throughput and reliability calculations. For the (8,4) MDS code considered above, the direct combination approach reduces the decoder processing complexity and results in a slight throughput performance improvement. These results are shown in Figure 10. The direct combination approach is possible because of the excellent incremental redundancy available in MDS codes. The direct combination approach justifies the assumption in Section 5.2 that the modified BER for the combined decoding operation does not differ significantly from the modified BER after a single decoding operation.

	$d_e = 0$	$d_e = 2$	$d_e = 4$
$d_{e3} = 4$	0.0053/-3.35	0.028/-1.83	0.126/-0.62
$d_{e3} = 5$	0.0057/-3.39	0.028/-1.83	0.126/-0.62
$d_{e3} = 6$	0.0126/-3.73	0.033/-1.89	0.127/-0.63
$d_{e3} = 7$	0.0142/-3.78	0.034/-1.91	0.128/-0.63
$d_{e3} = 8$	0.032/-4.14	0.047/-2.05	0.132/-0.65
$d_{e3} = 9$	0.036/-4.18	0.05/-2.08	0.134/-0.65
$d_{e3} = 10$	0.0697/-4.38	0.077/-2.27	0.145/-0.69
$d_{e3} = 11$	0.076/-4.25	0.082/-2.3	0.148/-0.69
$d_{e3} = 12$	0.119/-3.69	0.12/-2.44	0.168/-0.75

Table 2: Performance comparison of the (8,4) MDS code for various decoding sphere sizes.

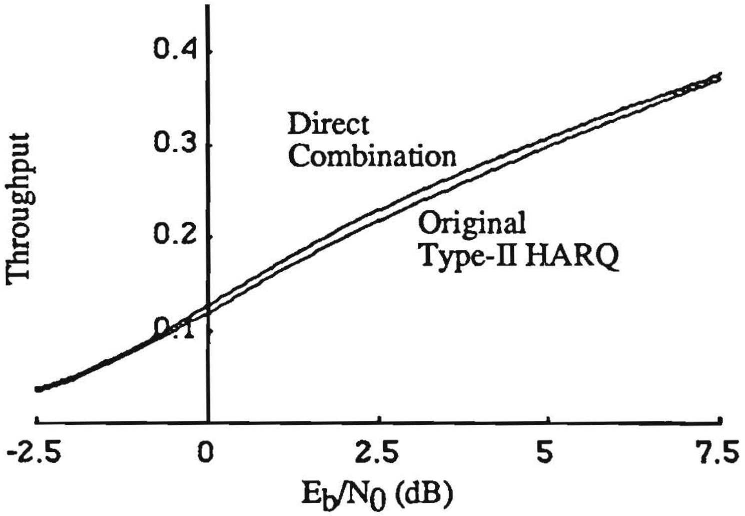

Figure 10: Comparison of the type-II protocol and the direct combination extension.

7 Conclusion and Comments

MDS codes have been shown to exhibit a series of properties that make them well suited for use in type-II hybrid-ARQ protocols. Strong separability and strong invertibility allow for the use of a decomposition process through which an (n, k) MDS code is used to create a pair of punctured $(n/2, k)$ MDS codes. The original code and the two derived codes are used individually in type-I hybrid-ARQ protocols. Together the three type-I protocols create a type-II protocol whose throughput and reliability performance is superior to that of any of the individual protocols.

The MDS decomposition process can be extended to develop more powerful code combining schemes. For example, a (64, 4) MDS code can be decomposed into eight (8, 4) codes to create a code combining system with eight different code rates similar to the variable-rate systems developed by Pursley and Sandberg [10]. Such a system will offer better performance than a type-II HARQ protocol in applications in which the channel varies slowly over a wide range of ambient noise levels.

A Performance Bounds on the Type-II System

Unfortunately, the complexity of Equation (7) increases with the fifth power of the effective diameter of the decoding operation. The computation of Equation (7) (as used in Equation (8)) thus begins to become a problem for the combined decoding operation D_3 for code lengths of 32 or more. If sufficient computing resources are not available, the following analysis can be used to obtain bounds on the performance of the type-II system.

Consider an (n, k) MDS code and a corresponding decoder with effective diameter d_e. A decoding error will occur if the received word is within the decoding sphere of diameter d_e surrounding an incorrect code word. The closest such code word is Hamming distance $d_{\min}$ away (the code is assumed to be linear). There must thus be a minimum of $(d_{\min} - \lfloor d_e/2 \rfloor)$ symbol errors in the received word for a decoder error to occur. If erasure decoding is available, a decoder error can occur only if, in addition to the above, the number of erasures is not greater than the effective decoding diameter d_e (otherwise a retransmission request will be generated). An upper bound is obtained by treating this pair of required events as if they were independent.

$$P_E \leq \begin{cases} 1 - \sum_{v=0}^{d_{\min} - \lfloor \frac{d_e}{2} \rfloor} \binom{n}{v} p_e^v (1-p_e)^{n-v} & \text{nonerasure decoding} \\ \left\{ 1 - \sum_{v=0}^{d_{\min} - \lfloor \frac{d_e}{2} \rfloor} \binom{n}{v} p_e^v (1-p_e)^{n-v} \right\} \cdot \left\{ \sum_{w=0}^{d_e} \binom{n}{w} p_s^w (1-p_s)^{n-w} \right\} & \text{erasure decoding} \end{cases} \tag{20}$$

The probability of retransmission is upper bounded by the probability that $(2e+s) > d_e$.

$$P_R \leq \begin{cases} 1 - \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \binom{n}{v} p_e^v (1-p_e)^{n-v} & \text{nonerasure decoding} \\ 1 - \sum_{v=0}^{\lfloor \frac{d_e}{2} \rfloor} \sum_{w=0}^{d_e - 2v} \binom{n}{v} \binom{n-v}{w} (1-p_e-p_s)^{n-v-w} p_e^v p_s^w & \text{erasure decoding} \end{cases} \tag{21}$$

The probabilities of symbol error and erasure in the above expressions (p_e and p_s respectively) are obtained from either Figure 4 (first attempt to decode) or Equations (15) and (19) (second and subsequent attempts to decode). If the code length is such that Equations (15) and (19) require unreasonable computation times, then the following approximations can be used.

$$P(\text{code symbol error} \mid \text{request}) = p_e' \approx \frac{1}{nP_R}(np_e - \Omega_0) \tag{22}$$

$$P(\text{code symbol erasure} \mid \text{request}) = p_s') \approx \frac{1}{nP_R}(np_s - \Psi_0) \tag{23}$$

where Ω_0 and Ψ_0 are as in Equations (12) and (16).

B Proof of Property 4

Property 4 can be proved constructively as follows. Let a k-bit message m be encoded using an arbitrary generator matrix **G** for an (n, k) MDS code **C**. Assume (without loss of generality) that it is desired to recover m from $[\mathbf{c}]_k$, the first k coordinates of code word $\mathbf{c} \in \mathbf{C}$.

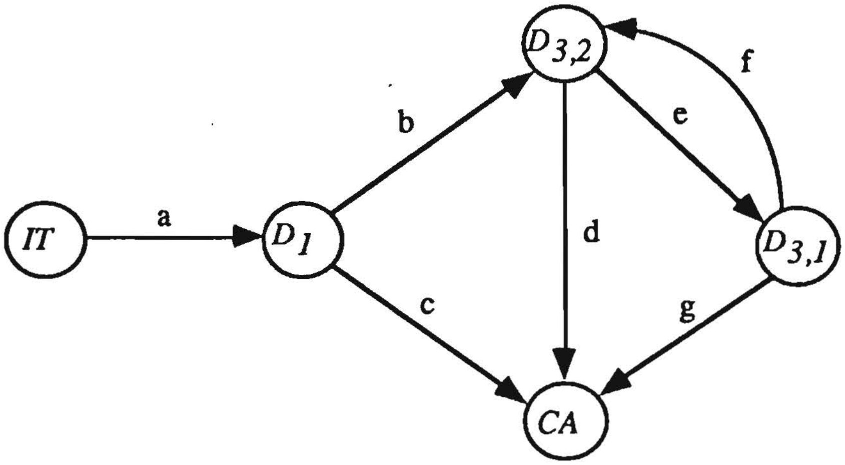

Figure 11: Generic graph for the direct combination type-II hybrid-ARQ protocol

Proof

Property 3 $\Rightarrow$ Any k columns of G are linearly independent [15]

$\Rightarrow$ Any $(k \times k)$ submatrix of $\mathbf{G}$ can be reduced to an identity matrix through Gaussian elimination.

$\Rightarrow$ $\exists$ nonsingular $(k \times k)$ matrix R such that $\mathbf{RG} = \mathbf{G} = [\mathbf{I}_k|\mathbf{P}]$
Let $[\mathbf{G}]_k$ be defined to be the $(k \times k)$ matrix containing the first k columns of $\mathbf{G}$.

$\Rightarrow$ $\mathbf{R}[\mathbf{G}]_k = [\mathbf{I}]_k$

$\Rightarrow$ $[\mathbf{G}]_k = \mathbf{R}^{-1}$

$\Rightarrow$ $[\mathbf{c}]_k = \mathbf{m}[\mathbf{G}]_k = \mathbf{mR}^{-1}$

$\Rightarrow$ $[\mathbf{c}]_k\mathbf{R} = [\mathbf{mR}^{-1}]\mathbf{R} = \mathbf{m}$ **QED.**

C Direct combination

The direct combination extension to the proposed type-II hybrid protocol is described by the generic graph and branch label table shown in Figure 11 and Table 3, respectively.

Branch label	Throughput label	Reliability label
a	T^{n_1}	1
b	$p_R^{(1)} \cdot T^{n_2}$	$p_R^{(1)}$
c	$1 - p_R^{(1)}$	$p_E^{(1)}$
d	$1 - p_R^{(3)}$	$p_E^{(3)}$
e	$p_R^{(3)} \cdot T^{n_1}$	$p_R^{(3)}$
f	$p_R^{(3)} \cdot T^{n_2}$	$p_R^{(3)}$
g	$1 - p_R^{(3)}$	$p_E^{(3)}$

Table 3: Graph labels for the derivation of throughput and reliability generating functions for the direct combination extension.

The generic transfer function is as follows:

$$IT = \left\{ ac + ab\left(d + \frac{eg}{1 - ef} \right) \right\} CA. \tag{24}$$

The reliability function is easily determined as

$$P(E) = P_E^{(1)} + \frac{P_E^{(3)} P_R^{(1)}}{1 - P_R^{(3)}},$$

and the throughput is

$$\eta = \frac{k\left(1 - P_R^{(3)^2}\right)}{n_1 + n_2 P_R^{(1)} + n_1 P_R^{(1)} P_R^{(3)} - n_1 P_R^{(3)^2}}.$$

The *a posteriori* or modified BER for the combined decoding operation must be averaged to account for the additional errors in the unreliable code word and the raw channel BER from the new code word.

References

[1] T. Kasami and S. Lin. On the probability of undetected error for the maximum distance separable codes. *IEEE Transactions on Communications*, COM-32(9):998–1006, September 1984.

[2] S. B. Wicker. High reliability data transfer over the land mobile radio channel using interleaved hybrid-ARQ error control. *IEEE Transactions on Vehicular Technology*, 39(1):48–55, February 1990.

[3] E. Berlekamp, R. Peile, and S. Pope. The application of error control to communications. *IEEE Communications Magazine*, 25:44–57, April 1987.

[4] D. Chase. Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Transactions on Communications*, COM-33(5):385 – 393, May 1985.

[5] S. Lin and P. S. Yu. A hybrid-ARQ scheme with parity retransmission for error control of satellite channels. *IEEE Transactions on Communications*, COM-30(7):1701 – 1719, July 1982.

[6] Y. Wang and S. Lin. A modified selective-repeat type-II hybrid-ARQ system and its performance analysis. *IEEE Transactions on Communications*, COM-31(5):593–607, May 1983.

[7] S. Lin, D. J. Costello Jr., and M. J. Miller. Automatic-repeat-request error control schemes. *IEEE Communications Magazine*, 22:5 – 17, December 1984.

[8] D. M. Mandelbaum. An adaptive feedback coding scheme using incremental redundancy. *IEEE Transactions on Information Theory*, 20:388 – 389, May 1974.

[9] D. M. Mandelbaum. On forward error correction with adaptive decoding. *IEEE Transactions on Information Theory*, 21:230 – 233, March 1975.

[10] Michael B. Pursley and Stuart D. Sandberg. Incremental-redundancy transmission for meteor-burst communications. *IEEE Transactions on Communications*, COM-39(3):689–702, May 1991.

[11] Michael B. Pursley and Stuart D. Sandberg. Variable-rate coding for meteor-burst communications. *IEEE Transactions on Communications*, 37(11):1105–112, November 1989.

[12] D. L. Lu and J. F. Chang. Analysis of ARQ protocols via signal flow graphs. *IEEE Transactions on Communications*, COM-37(3):245–251, March 1989.

[13] R. C. Singleton. Maximum distance q^nary codes. *IEEE Transactions on Information Theory*, IT-10:116 – 118, 1964.

[14] E. Berlekamp. *Algebraic Coding Theory.* Aegean Park Press, Laguna Hills, revised edition, 1984.

[15] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North Holland, New York, 1977.

[16] George C. Clark, Jr. and J. Bibb Cain. *Error-Correction Coding for Digital Communications.* Plenum Press, New York, 1981.

[17] J. Du, M. Kasahara, and T. Namekawa. Separable codes on type-II hybrid-ARQ systems. *IEEE Transactions on Communications*, COM-36(10):1089 – 1097, October 1988.

[18] Shu Lin and Daniel J. Costello, Jr. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[19] S. B. Wicker. Reed-solomon error control coding for slowly fading channels with feedback. *IEEE Transactions on Vehicular Technology*, 41(2):124–133, May 1992.

[20] Joachim Hagenauer and Erich Lutz. Forward error correction coding for fading compensation in mobile satellite channels. *IEEE Journal on Selected Areas in Communications*, SAC-5(2):215–225, February 1987.