Type-II Hybrid-ARQ Protocols Using Punctured MDS Codes

Stephen B. Wicker, Senior Member, IEEE and Michael J. Bartz, Member, IEEE

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.

I. 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 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 II 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.

II. 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, $C_1$ and $C_2$, to create a pair of code words $c_1$ and $c_2$. These codes have corresponding decoding operations $D_1$ and $D_2$ which can recover the original code word from noise corrupted versions of $c_1$ and $c_2$ respectively. $c_1$ comprises the initial transmission while $c_2$ is set aside. Upon receiving $c_1$, 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 acknowledgment (ACK) to the transmitter; otherwise, a negative acknowledgment (NACK) is sent. The transmitter responds to the NACK by sending $c_2$. The receiver then attempts to decode $c_2$ by itself using decoding operation $D_2$. If successful, the receiver inverts the corrected version of $c_2$ to recover the desired information and sends an ACK to the receiver. If unsuccessful, the receiver combines $c_1$ and $c_2$ to create $c_3$, a code word in a lower rate code $C_3$. 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 $c_1$ or $c_2$. 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. 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 event occurs, while 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:

$$IT = \left\{ ab + ac (d + ef + egh + egi) \left( \frac{1}{1 - eigk} \right) \right\} CA$$

(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 $C_1$ or $C_2$ respectively). Assuming a selective repeat protocol, the throughput generating function is as follows:

$$G(T) = T^{n_1}p_C^{(1)} + T^{n_1 + n_2}p_R^{(1)} \left[ p_C^{(2)} + T^{n_1}p_R^{(2)} p_R^{(3)}p_C^{(1)} + \right. \left. p_C^{(3)} \left( p_R^{(2)} + T^{n_1}p_R^{(2)} p_R^{(3)}p_R^{(1)} \right) \right]$$

$$\times \left( \frac{1}{1 - T^{n_1 + n_2}p_R^{(1)}p_R^{(2)}p_R^{(3)}} \right)$$

(2)

Once the throughput generating function has been obtained,
TABLE 1: Graph labels for the derivation of throughput and reliability generating functions

<table>
<thead>
<tr>
<th>Branch label</th>
<th>Throughput label</th>
<th>Reliability label</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$T_{b_1}$</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>$1 - P_E^{(1)}$</td>
<td>$P_E^{(1)}$</td>
</tr>
<tr>
<td>c</td>
<td>$P_E^{(0)}, T_{b_2}$</td>
<td>$P_E^{(0)}$</td>
</tr>
<tr>
<td>d</td>
<td>$1 - P_E^{(2)}$</td>
<td>$P_E^{(2)}$</td>
</tr>
<tr>
<td>e</td>
<td>$P_E^{(2)}, T_{b_1}$</td>
<td>$P_E^{(2)}$</td>
</tr>
<tr>
<td>f</td>
<td>$1 - P_E^{(3)}$</td>
<td>$P_E^{(3)}$</td>
</tr>
<tr>
<td>g</td>
<td>$P_E^{(3)}, T_{b_1}$</td>
<td>$P_E^{(3)}$</td>
</tr>
<tr>
<td>h</td>
<td>$1 - P_E$</td>
<td>$P_E$</td>
</tr>
<tr>
<td>i</td>
<td>$P_E$</td>
<td>$P_E$</td>
</tr>
<tr>
<td>j</td>
<td>$1 - P_E^{(1)}$</td>
<td>$P_E^{(1)}$</td>
</tr>
<tr>
<td>k</td>
<td>$P_E^{(1)}, T_{b_2}$</td>
<td>$P_E^{(1)}$</td>
</tr>
</tbody>
</table>

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 invertible [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 follows directly from the fact that any $(k \times k)$
submatrix of an MDS parity check matrix is nonsingular. Properties 3 and 4 can be shown to be equivalent.

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 \([14]\)

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

---

IV. PUNCTURED MDS CODES IN A TYPE-II HYBRID-ARQ PROTOCOL

In the MDS type-II protocol, the codes \( C_1, C_2, \) and \( C_3 \) 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 \( C_3 \). 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 \( C_3 \) are punctured to form code words in \( C_1, C_2 \). The first \( n/2 \) coordinates in a given \( C_3 \) code word \( c_{i,3} \) form the code word \( c_{i,1} \) in \( C_1 \), while the remaining \( n/2 \) coordinates form the code word \( c_{i,2} \) in \( C_2 \). Since codes \( C_1 \) and \( C_2 \) are punctured versions of the MDS code \( C_3 \), 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.

---

V. 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-II hybrid-ARQ protocols that combine to form a type-II protocol. In this section the design and analysis of the MDS type-II hybrid-ARQ protocol is discussed.

A. The MDS Type-II 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 \([2, 19]\) using 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-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 protocol.

B. 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 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 linear codes are only being considered, one may assume without loss of generality that the all-zero code word has been transmitted. Let $P_{d_e}$ 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, $P_{d_e}$ takes on the value:

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

(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:

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

(7)

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

Property 5 in Section III 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

$$
P_E = \sum_{j=0}^{n} A_j P_{d_e}^j$$

(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]:

$$
P_{R} = \begin{cases} 
1 - P_E - \sum_{v=0}^{\lfloor \frac{n}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{v} p_e v (1 - p_e)^{n-v} & \text{nonerasure} \\
1 - P_E - \sum_{v=0}^{\lfloor \frac{n}{2} \rfloor} \sum_{w=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{v} \binom{n-v}{w} (1 - p_e)^{n-v-w} p_e v & \text{erasure}
\end{cases}$$

(9)

The value of $P_E$ computed in Equation (8) is used for both $P_{R}^{(1)}$ and $P_{R}^{(2)}$ 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 $c$ of length $n$ is $np_e$ (before decoding).

$$
E\left\{\text{number of errors in } c\right\} = \frac{np_e}{n} = \frac{E\left\{\text{number of errors in } c\right\}}{\text{length of } c}
$$

(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 $e$.

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

(11)

If it is assumed that the received code word $c$ has caused the generation of a retransmission request, then $c$ cannot have fallen within the decoding spheres of effective diameter $d_e$ surrounding the correct and incorrect code words. Let $\Omega_0$ 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.,
\[ \Omega_0 = \sum_{i=0}^{n \choose 2} \sum_{w=0}^{n-u} \binom{n}{w} (n-w) (1-p_e-p_s)^{n-u-w} p_e^w p_s^w. \]  

Let \( \Omega_1 \) 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 weight \( j \) code words, define \( \Omega_1(j) \) to be

\[ \Omega_1(j) = \sum_{i=0}^{n \choose 2} \sum_{w=0}^{n-u} \sum_{x=0}^{n-j-w-2u} \sum_{y=0}^{n-j-v-2u} \binom{n-j-w-x-2u-2v}{j-x} \binom{n-j-v-y-2u-2v}{j-y} \binom{n-j-w-x-y-2u-2v}{j-x-y} \times (j-x-y) (n-w-x-y-2u-2v) (n-j-w-x-y-2u-2v) \times p_e^{j+y+j-y-2u} p_s^{j+y}(1-p_e-p_s)^{n-u-j-w}. \]

The number of code words of weight \( j \) is known (Property 5), so \( \Omega_1 \) can be computed as follows:

\[ \Omega_1 = \sum_{j=d_{\text{min}}}^{n} A_j \Omega_1(j) \]  

By removing the terms in \( \Omega_0 \) and \( \Omega_1 \) 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

\[ P(\text{code symbol error} \mid \text{request}) = p_e = \frac{1}{nP_k} (np_s - \Omega_0 - \Omega_1) \]  

The value of \( P_k \) 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:

\[ \psi_0 = \sum_{i=0}^{n \choose 2} \sum_{u=0}^{n-v} \binom{n}{u} (n-u) (1-p_e-p_s)^{n-v-u} p_e^u p_s^w \]  

\[ \psi_1(j) = \sum_{i=0}^{n \choose 2} \sum_{w=0}^{n-u} \sum_{x=0}^{n-j-w-2u} \sum_{y=0}^{n-j-v-2u} \binom{n-j-w-x-2u-2v}{j-x} \binom{n-j-v-y-2u-2v}{j-y} \binom{n-j-w-x-y-2u-2v}{j-x-y} \times (j-x-y) (n-w-x-y-2u-2v) (n-j-w-x-y-2u-2v) \times p_e^{j+y+j-y-2u} p_s^{j+y}(1-p_e-p_s)^{n-u-j-w} \]

\[ \psi_1 = \sum_{j=d_{\text{min}}}^{n} A_j \psi_1(j) \]  

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 \( \lambda_e = 0.1 \) (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 VI, 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.
VI. EXAMPLES

In this section several examples of the proposed protocol are examined. In addition, the qualitative effects of the decoding sphere sizes are considered. This section concludes with the 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 (C3) is decomposed into a pair of (8, 4) punctured MDS codes (C1 and C2). 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 D1 and D2 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 shows that the type-II protocol offers improved throughput performance at lower signal to noise ratios. On a nonstationary channel, the type-II protocol offers more graceful throughput degradation as the channel deteriorates. Figure 7 shows an 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)).

In the next example a (64,24) MDS code (C3) is decomposed into a pair of (32, 24) punctured codes (C1 and C2). 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 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 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, $d_{e3}$, introduces the same affect. The optimal setting for $d_e$ and $d_{e3}$
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 FOR 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 \(D_f\) 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 \(d_{\text{min}}\) away (the code is assumed to be linear). There must thus be a minimum of \((d_{\text{min}} - \lfloor d_e/2 \rfloor)\) 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.

\[
P_E \leq \left\{ \begin{array}{ll}
1 & \text{nonerasure} \\
1 - \sum_{v=0}^{d_{\text{min}} - \lfloor d_e/2 \rfloor} \binom{n}{v} p_e^v (1 - p_e)^{n-v} & \text{erasure}
\end{array} \right. \\
\times \sum_{w=0}^{d_e} \binom{n}{w} p_e^w (1 - p_e)^{n-w}
\]

(20)

The probability of retransmission is upper bounded by the probability that \((2e + s) > d_e\).
Figure 11: Generic graph for the direct combination type-II hybrid-ARQ protocol

![Figure 11: Generic graph for the direct combination type-II hybrid-ARQ protocol](image)

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

<table>
<thead>
<tr>
<th>Branch label</th>
<th>Throughput label</th>
<th>Reliability label</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$T_{n_{1}}$</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>$p_{E_{1}}^{(1)} \cdot T_{n_{2}}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
<tr>
<td>c</td>
<td>$1 - p_{E_{1}}^{(1)}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
<tr>
<td>d</td>
<td>$1 - p_{E_{1}}^{(1)}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
<tr>
<td>e</td>
<td>$p_{E_{1}}^{(1)} \cdot T_{n_{1}}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
<tr>
<td>f</td>
<td>$1 - p_{E_{1}}^{(1)}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
<tr>
<td>g</td>
<td>$1 - p_{E_{1}}^{(1)}$</td>
<td>$p_{E_{1}}^{(1)}$</td>
</tr>
</tbody>
</table>

The transfer function is as follows:

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

The reliability function is easily determined as

$$P(E) = \frac{p_{E_{1}}^{(1)} + p_{E_{2}}^{(1)} p_{E_{1}}^{(1)} - n_{1} p_{E_{1}}^{(1)}}{n_{1} + n_{2} p_{E_{1}}^{(1)} + n_{1} p_{E_{1}}^{(1)} p_{E_{1}}^{(1)} - n_{1} p_{E_{1}}^{(1)}}$$

and the throughput is

$$\eta = \frac{k \left( 1 - p_{E_{1}}^{(1)} \right)^{2}}{n_{1} + n_{2} p_{E_{1}}^{(1)} + n_{1} p_{E_{1}}^{(1)} p_{E_{1}}^{(1)} - n_{1} p_{E_{1}}^{(1)}}.$$

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


