Adaptive Rate Error Control Through the Use of Diversity Combining and Majority-Logic Decoding in a Hybrid-ARQ Protocol

Stephen B. Wicker

Abstract—**Diversity combining and majority-logic decoding are combined in this paper to create a simple but powerful hybrid-ARQ error control scheme. FEC majority-logic decoders are modified for use in type-I hybrid-ARQ protocols through the identification of reliability information within the decoding process. Diversity combining is then added to reduce the number of retransmissions and their consequent impact on throughput performance. Packet combining has the added benefit of adapting the effective code rate to channel conditions. Excellent reliability performance coupled with a simple high-speed implementation makes this majority-logic system an ideal choice for high data rate error control over both stationary and nonstationary channels.**

I. Introduction

Majority-logic techniques are used in the decoding of both block and convolutional codes to provide a modest amount of coding gain [1], [2]. The primary benefit attained through the use of these techniques is an implementational simplicity that allows for the operation of these decoders at very high data rates. This paper describes an adaptive rate error control system based on majority-logic decoding. The proposed system retains the simplicity of the original decoder, but provides a significant improvement in error protection through the incorporation of diversity combining and hybrid-ARQ techniques. In the next section a method is demonstrated for modifying majority-logic decoders for use in type-I hybrid-ARQ protocols. The modified decoder is then used in a diversity combining scheme designed to make efficient use of repeated transmissions. Analysis and simulation results are provided to show that the resulting system provides an extremely high level of data reliability on noisy channels at the expense of a minimal reduction in throughput.

II. The Modification of Majority-Logic Decoders for Use in Type-I Hybrid-ARQ Protocols

Error control techniques can generally be classified as either forward-error-correcting (FEC) or automatic-repeat-request (ARQ). In FEC schemes redundancy is added to transmitted data blocks to enable the receiver to correct some of the error patterns caused by noise on the channel. In ARQ schemes redundancy is added solely for the purpose of error detection. When one or more errors are detected in a received data block, a retransmission request is sent back to the originating transmitter. ARQ schemes provide a higher level of reliability than FEC schemes because a fixed amount of redundancy can detect approximately twice as many errors as it can correct. The cost for this increased reliability is exacted in the form of reduced throughput. Retransmission requests cause the throughput performance of ARQ schemes to be somewhat less than that of comparable FEC schemes.

The positive attributes of both FEC and ARQ techniques can be obtained by combining the techniques within a single *hybrid-ARQ* protocol [2], [3]. One of the more popular hybrid approaches involves the encoding of transmitted datablocks for both error detection and error correction. The error correction capability is used to resolve those error patterns most frequently caused by noise on the channel, thus reducing the number of retransmission requests. The residual detection capacity is used to detect the less frequently occurring error patterns, thus providing reliability performance approaching that of pure ARQ schemes. These hybrid protocols are commonly referred to as *type-I hybrid-ARQ* protocols [3].

Standard FEC codecs can be modified for use in type-I hybrid-ARQ protocols through the identification of reliability information within the decoding algorithm. This information is used to estimate the reliability of decoded data packets and to determine whether or not a retransmission request is appropriate. An excellent example of this type of modification has been provided by Drukarev and Costello [8]. They showed that computation time and path metric slope are directly related to decoded data packet reliability in sequential decoders. This information is used in the "time out" and "slope control" algorithms, respectively, to create two type-I hybrid-ARQ protocols based on the sequential decoding of convolutional codes. It has also been shown that both block and convolutional majority-logic decoders can be modified for use in type-I hybrid-ARQ protocols [5], [6], [15]. It is this particular modification that is pursued and extended in this paper.

Majority-logic decoding was first introduced by Reed in 1954 [16] and then given a thorough development by Massey in 1963 [1]. The common focus of majority-logic techniques is the use of selected syndrome bits in a voting scheme to resolve errors in the information portion of received data blocks. Majority-logic techniques can be used with both block and convolutional codes to achieve a moderate amount of coding gain. The primary benefit lies in the implementation of the decoder. The architecture of majority-logic decoders is highly pipelined and consists of two or more parallel shift registers and a few simple gates. These decoders may thus be used in applications whose data rate requirements preclude the use of other error control techniques.

The requisite reliability information is contained within the "voting" portion of the majority-logic decoding scheme. Consider the case of the majority-logic decoding of rate 1/2 systematic convolutional codes [1], [2]. Codewords consist of an information sequence $v^{(1)}$ interleaved with a parity sequence $v^{(2)}$. Both sequences are used by the receiver to compute a syndrome sequence s. The first step in majority-logic decoding is the estimation of the first information sequence error bit $e_0^{(1)}$.

Paper approved by the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received March 30, 1989; revised November 21, 1989. This work was supported by a grant from the AT&T Foundation. This paper was presented at the 1989 IEEE MILCOM, Boston, MA, October 17, 1989 under the title, "An Adaptive Rate Coding System Based on the Use of Majority-Logic Decoding in a Hybrid-ARQ Protocol."

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

IEEE Log Number 9042394.

0090-6778/91/0300-0380$01.00 ©1991 IEEE

$e_0^{(1)}$ affects only those syndrome bits corresponding to the first constraint length $[r]_m$ of the interleaved received sequence. Let $[H]_m$ be the correspondingly truncated version of the parity check matrix H. H is composed of shifted versions of the impulse responses $g^{(1)}$ and $g^{(2)}$ of the convolutional encoder [2]. The syndrome bits affected by $e_0^{(1)}$ can now be expressed in the following manner:

$$\begin{aligned}[s^T]_m &= [H]_m[r^T]_m \\ &= [H]_m[e^T]_m \\ &= \begin{bmatrix} g_0^{(2)} & & & \\ g_1^{(2)} & g_0^{(2)} & & \\ \vdots & \vdots & \ddots & \\ g_m^{(2)} & g_{m-1}^{(2)} & \cdots & g_0^{(2)} \end{bmatrix} \begin{bmatrix} e_0^{(1)} \\ e_1^{(1)} \\ \vdots \\ e_m^{(1)} \end{bmatrix} + \begin{bmatrix} e_0^{(2)} \\ e_1^{(2)} \\ \vdots \\ e_m^{(2)} \end{bmatrix}. \end{aligned} \quad (1)$$

This expression shows that each syndrome bit consists of the sum of several information sequence error bits and a single parity sequence error bit. A set of check sums orthogonal on the first information sequence error bit $e_0^{(1)}$ can now be selected from among these syndromes. A set of check sums is said to be orthogonal on $e_0^{(1)}$ if all of the check sums contain (or "check") the bit $e_0^{(1)}$ while all other error bits are checked by at most one of the check sums in the set. The orthogonal set of check sums is used to estimate the value of $e_0^{(1)}$ in the following manner [1], [2].

A. Majority-Logic Decoding Rule

Let J be the number of check sums orthogonal on information sequence error bit e. Let η be the number of check sums in this set that have the value 1. If $\eta > \lfloor J/2 \rfloor$, then e is assumed to have a value of one. Otherwise e is assumed to have a value of zero.

Let l be the total number of error bits checked by the J check sums. Of these l bits, one bit is common to all of the check sums. This is the information sequence error bit being estimated by the majority-logic decoding process. The other $l - 1$ error bits are each contained in exactly one of the check sums. Of these $l - 1$ b, J are parity sequence error bits and $l - J - 1$ are information sequence error bits. l is commonly called the *effective constraint length* of the majority-logic decodeable block or convolutional code.

The above decoding rule is implemented in an FEC decoder through the use of a majority-logic gate. A generalized majority-logic gate examines a set of binary inputs and provides an output of one if K or more of the inputs have a value of one. If $K - 1$ or fewer of the inputs have a value of one then the output of the gate is zero. Clearly, an FEC majority-logic decoder will use a J input majority-logic gate with the constant K set at $\lfloor J/2 \rfloor + 1$.

The extent of the majority at the input of the majority-logic gate provides the reliability information necessary for the type-I hybrid-ARQ modification. This information is utilized in the following modified majority-logic decoding rule [5], [6], [15].

B. Modified Majority-Logic Decoding Rule

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.[1]

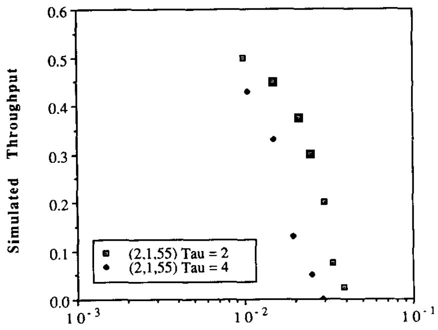

Fig. 1. Simulated throughput performance for a (2, 1, 55) convolutional code in a hybrid-ARQ protocol using majority-logic decoding [6]. Packet length = 1000.

This rule clearly defines a type-I hybrid-ARQ protocol. The most frequent error patterns will be those that cause almost all of the J check sums to be either one or zero. When these patterns occur, decoding will proceed normally. The less frequent, higher weight error patterns will cause the majority among the J check sums to be less pronounced. These are the patterns most likely to cause decoder errors, and so these will be the patterns that trigger retransmission requests.

The performance of this scheme with block and convolutional codes is quite good at bit error rates up to approximately 10^{-2} [6], [15]. At this point the number of retransmissions becomes so high as to make communication impossible. Fig. 1 contains simulation data for a (2, 1, 55) convolutional code [6]. This self-orthogonal code has an effective constraint length of 53 [2, p. 418]. It should be noted that throughput performance degrades rapidly once the channel bit error rate exceeds 10^{-2} even for small values of τ.

In the next section it is shown that the frequency of retransmissions can be kept low even at very high channel bit error rates through the incorporation of diversity combining of multiple transmissions. The resulting system will thus retain the reliability benefits of a hybrid-ARQ protocol while minimizing the reduction in throughput associated with such protocols. It is also shown that the adaptive nature of packet combining makes system throughput degradation much more graceful at high channel bit error rates.

III. Packet Combining in a Majority-Logic-Based Hybrid-ARQ System

Type-I hybrid-ARQ protocols are an effective form of error control for stationary channels that experience occasional bursts of noise. The FEC portion of the protocol protects against the influence of ambient noise while the ARQ portion takes care of the bursts. If the bursts are relatively infrequent, then it is highly improbable that a particular data packet will have to be transmitted more than twice. Problems arise, however, when the noise level increases for a period of time exceeding a single packet length. The type-I protocol's only course of action is to continually retransmit in the hope of getting a packet through to the

[1] The use of ceiling and floor functions maximizes flexibility in setting the retransmission region and also ensures that a value of $\tau = 0$ will be equivalent to FEC operation regardless of the value of J.

receiver with a correctable error pattern. As shown in Fig. 1, the result can be a catastrophic reduction in throughput. There are two basic methods by which a type-I protocol can be modified to handle this problem. The first is to monitor retransmission statistics and change code rate accordingly [17], [18]. This has the effect of increasing the amount of FEC error control within the system, bringing the retransmission rate back within acceptable limits. The second method involves combining multiple received packets to create a more reliable version of the transmitted data. This second alternative is pursued in the remainder of this paper.

The basic premise underlying packet combining is that a received packet always contains at least a small amount of useful information. This information can be used in conjunction with other received copies of the packet to obtain an estimate of the transmitted data that is more reliable than that obtainable from any of the packets by themselves. There are two basic approaches to combining multiple received packets: *code combining* and *diversity combining*. Code combining involves taking N packets encoded at rate R and creating a single packet encoded at rate R/N. The combining is thus done at the codeword level and not at the bit or symbol level. This approach is thoroughly examined in a paper by Chase [12]. Type-II hybrid-ARQ protocols can be viewed as a truncated form of code combining. In Lin and Yu's scheme the second transmission contains parity bits for the first [4]. The two packets are joined together within the received to create a single rate $1/2$ systematic codeword.

Diversity combining differs from code combining in that multiple copies of a packet encoded at rate R are combined bit by bit to create a single codeword from the original rate R code. Each bit in the resulting packet is made more reliable through the receipt of multiple copies of each bit (repetition redundancy [11]). Though not as powerful as code combining [12], diversity combining is much simpler to implement.

The combining of the individual bit can be performed using either soft or hard decision techniques. In the soft-decision approach a single estimate of the transmitted packet is maintained in a register along with reliability information for each bit in the estimate [9], [10]. Subsequent received copies are weighted accordingly and added to the estimate. Once the estimated packet is recognized to have a correctable error pattern, the packet is acknowledged, decoded, and passed along to the data sink.

Hard-decision diversity combining is not as efficient as soft-decision combining for the obvious reason that the limiting process removes information from the received packets. It should be noted however that hard-decision techniques have the benefit of reduced complexity and increased speed. A soft-decision diversity combiner appended to the type-I majority-logic protocol discussed earlier in this paper would reduce the speed at which the system could operate and essentially eliminate the benefit of using the majority-logic decoder in the first place. If the throughput degradation shown in Fig. 1 is to be postponed and made more graceful without reducing the simplicity and speed of the overall system, then a hard decision majority-logic combining approach should be adopted.

A. Majority-Logic Diversity Combining

Majority-logic diversity combining is the use of multiple copies of each transmitted bit in a voting scheme to obtain a single more reliable version of each bit. To prevent the occurrence of ties, a special retransmission protocol is adopted. It

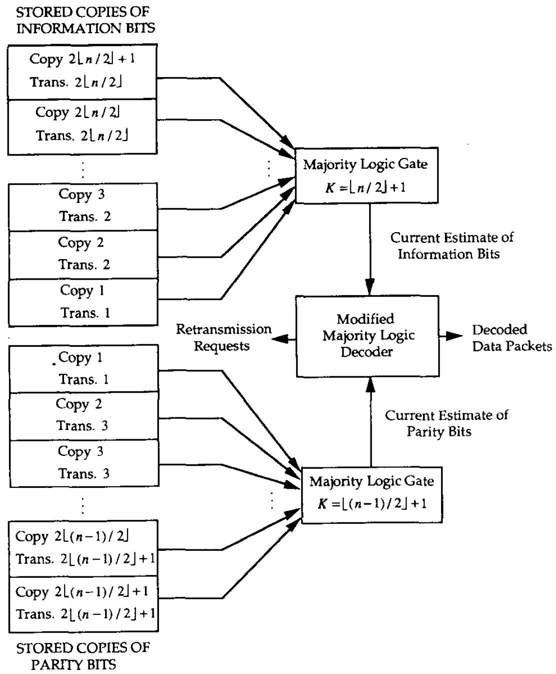

Fig. 2. Diversity combining majority-logic hybrid-ARQ decoder.

shall be assumed that the packets are encoded using rate $1/2$ systematic block or convolutional codes,[2] though the following can be readily modified for use at other code rates. The transmission/retransmission protocol shall be as follows:

Transmission #1: Normally encoded data packet.

Transmission #2: Two copies of the information bits.

Transmission #3: Two copies of the parity bits.

Subsequent even numbered transmissions: Two copies of the information bits.

Subsequent odd numbered transmissions: Two copies of the parity bits.

This protocol has been designed so that all transmissions will contain the same number of bits. It should also be noted that at any given moment the receiver will have an odd number of copies of the information portion of the codeword and an odd number of copies of the parity portion of the codeword. It is thus impossible to have a tie in the bit by bit voting procedure. Finally, the selection of the information bits for the first of the retransmissions is not arbitrary. In (1) it was shown that each of the syndromes is the sum of several information bits and a single parity bit. The syndrome is thus made reliable more quickly by increasing the reliability of the information bits first.

Fig. 2 contains a system block diagram for a diversity combining majority-logic hybrid-ARQ decoder for rate $1/2$ codes. Simplicity and speed have been retained at the expense of additional memory. The throughput performance of the system is determined in the next section. The buffering requirements are considered in the final section.

B. Reliability and Throughput Performance

The retransmission process alternates between the transmission of copies of the information bits and copies of the parity

[2] Good convolutional codes for this purpose can be found in Lin and Costello [2, ch. 13]. Majority-logic decodable block codes from Euclidean and projective geometry can be found in Hartman *et al.* [19].

bits. As the number of copies being combined increases, the reliability of the individual bits in the estimate increases. If the information bits are the first to be retransmitted, n successive transmissions will result in the receiver having $(2\lfloor n/2 \rfloor + 1)$ copies of the information bits and $(2\lfloor (n-1)/2 \rfloor + 1)$ copies of the parity bits.

The bits in the copies of each sequence are combined through a simple voting procedure: the estimate for each sequence bit will be determined by whichever value is in the majority amongst all of the copies of that sequence bit. As noted earlier, a majority is guaranteed because there is always an odd number of copies of each of the sequences. It is assumed that the communication channel is binary symmetric with bit error probability p. The bit error probability $p_{\text{inf},n}$ of the bits in the estimate of the information sequence after the nth transmission is thus

$$p_{\text{inf},n} = \sum_{k=\lfloor n/2 \rfloor + 1}^{2\lfloor n/2 \rfloor + 1} \binom{2\lfloor n/2 \rfloor + 1}{k} p^k (1-p)^{2\lfloor n/2 \rfloor - k + 1}. \tag{2}$$

The bit error probability $p_{\text{par},n}$ of the bits in the estimate of the parity sequence after the nth transmission is

$$p_{\text{par},n} = \sum_{k=\lfloor (n-1)/2 \rfloor + 1}^{2\lfloor (n-1)/2 \rfloor + 1} \binom{2\lfloor (n-1)/2 \rfloor + 1}{k} \cdot p^k (1-p)^{2\lfloor (n-1)/2 \rfloor - k + 1}. \tag{3}$$

An upper bound can be obtained for $P_{e,n}$, the probability of decoder error on the nth transmission, in the following manner. The majority-logic decoding rule will make an incorrect error bit estimate if enough of the $l-1$ error bits in the orthogonal set of check sums (the bit being estimated is excluded) have a value of one. Given that a retransmission is requested if the number of check sums with a value of one is greater than $\lfloor J/2 \rfloor - \lceil \tau/2 \rceil$ while less than $\lfloor J/2 \rfloor + \lfloor \tau/2 \rfloor + 1$, the following conservative assumption is made. A decoder error will be made whenever there are $\lfloor J/2 \rfloor + \lfloor \tau/2 \rfloor + 1$ or more errors among the other $l-1$ error bits in the orthogonal set of check sums. Though these errors may be distributed in any manner among the J parity sequence error bits and the l-J-1 information sequence error bits, the worst case assumption implies that the erroneous bits are in the worst possible positions within the check sums. The following are the upper bound results:

$$P_{e,n} \le \sum_{e=\lfloor J/2 \rfloor + \lfloor \tau/2 \rfloor + 1}^{l-1} \left\{ \sum_{j=0}^{e} \binom{J}{j} \binom{l-J-1}{e-j} p_{\text{inf},n}^{e-j} p_{\text{par},n}^{j} \times (1-p_{\text{inf},n})^{l-J-1-e+j} (1-p_{\text{par},n})^{J-j} \right\}. \tag{4}$$

The probability of a retransmission request being generated by the packet estimate after the nth transmission $(P_{r,n})$ can be bounded through the use of similar assumptions. It is assumed that any combination of $\lfloor J/2 \rfloor - \lceil \tau/2 \rceil + 1$ or more errors among the $l-1$ bits used in the estimation process will cause the generation of a retransmission request. Let $\psi_{e,n}$ be the probability of there being exactly e error bits with a value of one among a specified set of $l-1$ error bits in the estimated packet. $\psi_{e,n}$ is exactly computed as

$$\psi_{e,n} = \sum_{j=0}^{e} \binom{J}{j} \binom{l-J-1}{e-j} p_{\text{inf},n}^{e-j} p_{\text{par},n}^{j} \cdot (1-p_{\text{inf},n})^{l-J-1-e+j} (1-p_{\text{par},n})^{J-j}. \tag{5}$$

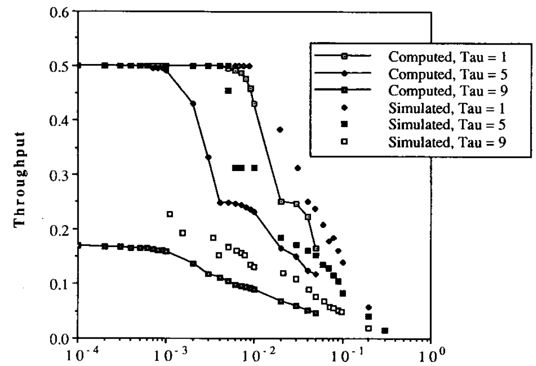

Fig. 3. The lower bound on throughput and simulation results for a (2, 1, 55) convolutional code in a diversity combining majority-logic hybrid-ARQ protocol. Packet Length = 2000.

During the course of decoding a packet of length N, the modified decoder will make estimates of each of the $N/2$ error bits in the information sequence. Each one of these estimates affords the opportunity for the generation of a retransmission request. Using a union bound approach, the following upper bound is obtained for the probability of a retransmission request on the nth transmission.

$$P_{r,n} \le \frac{N}{2} \sum_{e=\lfloor J/2 \rfloor - \lceil \tau/2 \rceil + 1}^{l-1} \psi_{e,n}. \tag{6}$$

Given $P_{e,n}$ and $P_{r,n}$ it is possible to obtain expressions for overall system reliability and throughput. We begin with the latter. Let T be the expected number of transmissions necessary for a packet to be accepted by the receiver. T is calculated in the following manner:

$$\begin{aligned} T &= 1(1-P_{r,1}) + 2(1-P_{r,2})P_{r,1} + 3(1-P_{r,3})P_{r,2}P_{r,1} + \cdots \\ &= \sum_{j=1}^{\infty} j(1-P_{r,j}) \prod_{k=0}^{j-1} P_{r,k} \end{aligned} \tag{7}$$

where $P_{r,0}$ is defined to be 1.

The terms in the above expression get very small very quickly, making it possible to obtain useful values from a truncated summation. It shall be assumed that a selective-repeat hybrid-ARQ protocol is being used [2], [3], though the following can be easily modified for use with stop-and-wait and go-back-N protocols. The throughput of the selective-repeat system is $\upsilon = (k/n)T^{-1}$ where k/n is the rate of the code being used. Fig. 3 shows the computed upper bound on throughput for the same (2, 1, 55) convolutional code used to generate Fig. 1. Several values of the retransmission threshold τ are used, and the packet length has been increased to 2000 b. At throughput values of 0.25 and 0.17 (corresponding to $T = 2$ and $T = 3$) the curves flatten out momentarily, reflecting the added error correction capability made available by the additional copies of the information and parity sequences. Herein lies the adaptive nature of the system. As the ambient noise level on the channel varies, the number of packets combined at the receiver for each data packet varies as well. These curves also suggest a graceful degradation in system throughput performance as the noise level grows. Simulation data has been included in Fig. 3 to verify the tightness of the lower bound on throughput. It is shown that the lower bound is

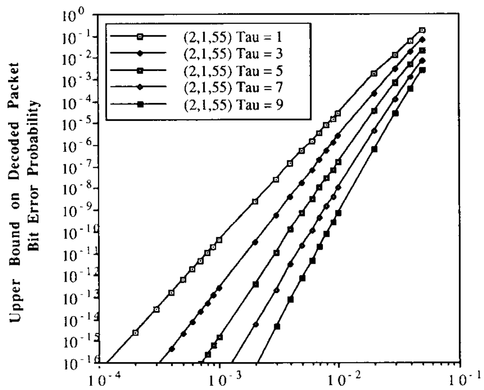

Fig. 4. The upper bound on decoded packet bit error probability for a (2, 1, 55) convolutional code in a diversity combining majority-logic hybrid-ARQ protocol. Packet length = 2000.

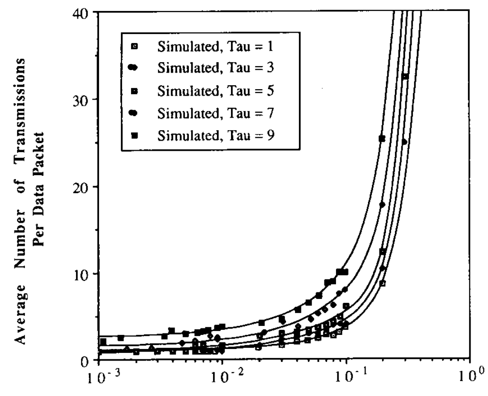

Fig. 5. Average number of transmissions per data packet for the majority-logic diversity combiner using a (2, 1, 55) convolutional code. Packet length = 2000.

not only tight, but also clearly reflects the adaptive nature of the system. The addition of majority-logic combining has thus extended the performance of the type-I hybrid-ARQ protocol to very high bit error rates while making throughput degradation much more gradual as the channel degrades.

The probability of bit error within accepted packets $P(E)$ is obtained in a similar manner.

$$P(E) = P_{e,1} + P_{e,2}P_{r,1} + P_{e,3}P_{r,2}P_{r,1} + \cdots = \sum_{j=1}^{\infty} P_{e,j} \prod_{k=0}^{j-1} P_{r,k}. \quad (8)$$

A useful closed-form upper bound can be derived from this expression.

$$\begin{aligned} P(E) &= P_{e,1} + P_{e,2}P_{r,1} + P_{e,3}P_{r,2}P_{r,1} + \cdots \\ &\le P_{e,1}(1 + P_{r,1} + P_{r,2}P_{r,1} + \cdots) \\ &\le P_{e,1}(1 + (1 - P_{r,1}) + 2(1 - P_{r,2})P_{r,1} \\ &\quad + 3(1 - P_{r,3})P_{r,2}P_{r,1} + \cdots)) \\ &= P_{e,1}(1 + T). \end{aligned} \quad (9)$$

Fig. 4 shows the computed upper bound for decoded packet bit error rate for the (2, 1, 55) convolutional code used in the previous figures. The slope of the curves increases rapidly with τ, indicating the improvement in reliability obtained by widening the region of syndrome values that cause the generation of retransmission requests. The amount of error protection provided by this system is clearly quite substantial.

C. Buffering Considerations

The majority-logic combining scheme maintains the principal advantage of majority-logic decoders at the expense of additional memory. Multiple copies of received information and parity bits must be stored in the receiver until a reliable estimate of the transmitted packet can be made. The amount of memory necessary is a function of the retransmission protocol being used. There are three basic protocols: stop-and-wait, go-back-N, and selective repeat [2], [3]. They are listed here in order of increasing buffer utilization. A rigorous determination of the required buffer size must take into account several factors, the most important of which are forward and return transmission, propagation, and processing delays. The role of these factors varies according to the retransmission protocol in use. The common denominator in all three cases (and their various hybrids) is T, the average number of copies of each packet that will be required to generate a reliable estimate of the transmitted packet. An expression for T was provided in (7). Fig. 5 shows the values for T obtained over one hundred simulation runs using the (2, 1, 55) convolutional code. The solid lines in the graph are third-order polynomial curves fit to the simulation data. Even for high values of τ the number of copies of each packet that must be stored remains small up to channel bit error rates on the order of 10^{-1}.

References

[1] J. L. Massey, *Threshold Decoding*. Cambridge, MA: M.I.T. Press, 1963.
[2] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[3] 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.
[4] S. Lin and P. S. Yu, "A hybrid-ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Trans. Commun.*, vol. COM-30, pp. 1701–1719, July 1982.
[5] S. B. Wicker, "An error control technique for high data rate communication networks," Proc. INFOCOM 1989.
[6] ——, "Modified majority-logic decoders for use in convolutionally encoded hybrid-ARQ systems," *IEEE Trans. Commun.*, vol. 38, pp. 263–266, Mar. 1990.
[7] Y. Wang and S. Lin, "A modified selective-repeat type-II hybrid-ARQ system and its performance analysis," *IEEE Trans. Commun.*, vol. COM-31, pp. 593–607, May 1983.
[8] A. Drukarev and D. J. Costello, Jr., "Hybrid ARQ error control using sequential decoding," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 521–535, July 1983.
[9] G. Benelli, "An ARQ scheme with memory and soft error detectors," *IEEE Trans. Commun.*, vol. COM-33, pp. 285–288, Mar. 1985.
[10] ——, "An ARQ scheme with memory and integrated modulation," *IEEE Trans. Commun.*, vol. COM-35, pp. 689–697, July 1987.
[11] P. Sindhu, "Retransmission error control with memory," *IEEE Trans. Commun.*, vol. COM-25, pp. 473–479, May 1977.
[12] D. Chase, "Code combining—A maximum-likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Trans. Commun.*, vol. COM-33, pp. 385–393, May 1985.

[13] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1984, 2nd ed., p. 48.

[14] D. E. Knuth, *The Art of Computer Programming*, *Volume 2*. Reading, MA: Addison-Wesley, 1981, 2nd ed.

[15] M. D. Rice and S. B. Wicker, "Modified majority-logic decoding of cyclic codes in hybrid-ARQ systems," *IEEE Trans. Commun.*, to be published.

[16] I. S. Reed, "A class of multiple error-correcting codes and the decoding scheme," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 38–49, Sept. 1954.

[17] S. B. Wicker, "Hybrid-ARQ Reed–Solomon coding in an adaptive rate system," in *Proc. 1989 Int. Conf. Commun.*, pp. 45.5.1–45.5.5.

[18] M. D. Rice and S. B. Wicker, "Adaptive error control for slowly varying channels," *IEEE Trans. Commun.*, to be published.

[19] C. R. P. Hartmann, J. B. Ducey, and L. D. Rudolph, "On the structure of generalized finite geometry codes," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 240–252, Mar. 1974.