Analysis of a Memory and an Incremental Redundancy ARQ Schemes Over a Nonstationary Channel

Samir Kallel
UBC Department of Electrical Engineering
ANALYSIS OF A MEMORY AND AN INCREMENTAL REDUNDANCY ARQ SCHEMES OVER A NONSTATIONARY CHANNEL

by

S. Kallel

Department of Electrical Engineering
University of British Columbia
2356 Main Mall, B.C.
Vancouver, Canada, V6T 1W5
Tel: (604)-228-2653

Abstract

Recently, a type II ARQ scheme using convolutional coding has been analyzed over a nonstationary channel. It has been shown that the type II ARQ scheme performs very well on this type of channel. In this paper we analyze a generalized type II ARQ scheme using punctured convolutional coding on a two-state Markov model of a nonstationary channel. A simple ARQ scheme with memory is also analyzed. It is shown that the simple memory ARQ scheme offers a substantial throughput improvement over a conventional ARQ scheme at severe channel conditions. Furthermore, it is shown that the generalized type II ARQ scheme yields a better performance than the conventional type II ARQ scheme under all channel conditions, thus making it attractive for use over time-varying channels. ¹

¹This research was supported by the Natural Sciences and Engineering Research Council of Canada.
1 Introduction

Basically, there are two techniques to control errors in data communications [1]: Automatic-Repeat-Request (ARQ) schemes and Forward-Error-Correction (FEC) schemes. In an ARQ scheme, a high rate block code is used for error detection with a certain retransmission protocol. In an FEC scheme, a block code or a convolutional code is used for error correction.

ARQ schemes are simple to implement and offer very high reliability. Unfortunately their throughput falls off rapidly as the channel error rate increases. On the other hand, FEC schemes provide constant throughput but suffer a decrease in reliability at high channel error rates. A hybrid FEC/ARQ scheme which judiciously combines ARQ and FEC can offer performance superior to either scheme alone [1].

In order to enhance the performance of ARQ and hybrid FEC/ARQ systems, several authors have suggested that the decoder should make use of the erroneously received packets and combines them with their repeated copies to help in recovering the transmitted message [2]-[6]. Others have proposed that instead of repeating the same data packet which is received in error, the transmitter should provide additional incremental redundancy bits [7]-[15].

A well known hybrid FEC/ARQ scheme which is based on the later idea of incremental redundancy is the so called type II hybrid ARQ scheme [7]-[10]. A type II ARQ scheme uses a rate 1/2 invertible block or convolutional code. A data packet is first transmitted along with parity bits for error detection, and parity bits for error correction are sent to the receiver only when they are needed. The disadvantage of the type II ARQ scheme is that the redundancy bits for error correction are sent to the receiver all at once even if they are not all needed, thus reducing the channel use efficiency. A generalized type II hybrid ARQ scheme using punctured convolu-
Convolutional coding has been recently proposed in which redundancy bits are provided by the transmitter, a few at a time, sequentially, as needed [13]-[15]. The generalized type II hybrid ARQ scheme has been analyzed on a stationary channel [13]-[15]. It has been shown that the generalized type II ARQ scheme offers a better throughput than the conventional type II ARQ scheme.

In a recent paper, it has been shown that a type II ARQ scheme using convolutional coding performs very well on a nonstationary channel [16]. In this paper, we analyze the generalized type II ARQ scheme [15] on a nonstationary channel. A simple ARQ scheme in which packets received with errors are combined in an optimum manner with their repeated copies is also analyzed over the nonstationary channel. Throughput results are compared to that of a basic ARQ scheme.

A nonstationary channel model is presented in section 2. Section 3 describes the different ARQ schemes to be analyzed. A throughput analysis of the different ARQ schemes is given in section 4. Numerical results are finally provided in section 5, followed by some conclusions in section 6.

2 Description of the channel model

The channel model considered in this paper is the two state Markov channel model [16]-[18] (see Figure 1). In Figure 1, state 0 is the quiet state where the bit error rate (BER) is \( \epsilon_0 \), state 1 is the noisy state where the BER is \( \epsilon_1 \gg \epsilon_0 \), \( p \) is the transition probability from state 0 to state 1 and \( \hat{p} \) is the transition probability from state 1 to itself.

From Figure 1, the average burst length [16]-[17] is,

\[
\bar{b} = \frac{1}{1 - \hat{p}} \quad (1)
\]
the average BER is
\[ \tau = \frac{(1 - \hat{p})\epsilon_0 + p\epsilon_1}{1 - \hat{p} + p} \]  
\hspace{1cm} (2)

and the duty cycle of the noisy bursts is
\[ p_1 = \frac{p}{1 - \hat{p} + p} \]  
\hspace{1cm} (3)

We consider the two cases of diffuse burst channel [18],[16] (i.e. large duty cycle \( p_1 \) and low ratio \( \rho = \epsilon_1/\epsilon_0 \)), and dense burst channel (i.e. low duty cycle \( p_1 \) and high intensity \( \rho \)). Following [16], we let

\[ \epsilon_0 = \tau p_1 \]  
\hspace{1cm} (4)

From (2) and (4), we have [16]

\[ \epsilon_1 = \begin{cases} 
\frac{(\tau/p_1) - (1 - p_1)\tau}{1/2} & \text{for } p_1 > \left[ \tau + 1/2 - \sqrt{(1/2 - \tau)(1/2 + 3\tau)} \right] / 2\tau \\
1/2 & \text{otherwise}
\end{cases} \]  
\hspace{1cm} (5)

Thus the burst channel model is completely described by \( \tau, p_1, \) and \( b \). Given these three parameters, \( \hat{p}, p, \epsilon_0 \) and \( \epsilon_1 \) can be determined from respectively (1), (3), (4), and (5). The particular case \( p_1 = p = \hat{p} \) corresponds to the two-state block interference (BI) channel model [19]. The BI channel model is completely characterized by \( p_1 \) and \( \tau \).

We assume that one time frame in the model corresponds to the transmission of one data sequence. Let \( b \) be the number of data sequences that can be transmitted during one channel roundtrip delay period. In the following analysis of the ARQ schemes, we need to have the channel \( b \)-step transition probabilities. We denote the probability of being in state \( s_j, (s_j=0 \text{ or } 1) \), \( b \) time frames after being in state \( s_{j-1}, (s_{j-1}=0 \text{ or } 1) \), by \( P_{s_{j-1}s_j}^{(b)} \). From [16]-[17], the four possible channel \( b \)-step transition probabilities are
\[ P_{01}^{(b)} = \frac{p}{1 + p - \hat{p}} - \frac{p}{1 + p - \hat{p}}(\hat{p} - p)^b \]  
\[ P_{11}^{(b)} = \frac{p}{1 + p - \hat{p}} + \frac{(1 - \hat{p})}{1 + p - \hat{p}}(\hat{p} - p)^b \]  
\[ P_{10}^{(b)} = 1 - P_{11}^{(b)} \]  
\[ P_{00}^{(b)} = 1 - P_{01}^{(b)} \]

3 Description of the ARQ schemes

In this section, the different ARQ schemes analyzed later are briefly reviewed. The following assumptions are made throughout this paper. A noiseless feedback channel is available so that the receiver can reliably inform the transmitter of the packets successfully decoded. Whenever the transmitter does not have any repeats to send, it transmits new packets. An infinite receiver buffer is available to store successful packets following a packet which is detected in error.

**Scheme 1**

This is the conventional type 0 ARQ scheme [1]. Each transmitted L-bit data packet contains \( k \) information bits and \( n_r \) parity check bits used only for error detection purposes. Whenever a data packet is detected in error, that packet is discarded and replaced by its repeated copy.

**Scheme 2**

Scheme 2 is obtained by modifying Scheme 1 as follows. Instead of discarding those packets detected in error, the receiver combines them with their repeated copies in an
optimum manner. Each copy is weighted by a reliability factor. Assuming a perfect knowledge of the channel state by the receiver, it can be shown [5] that the weighting reliability factor is

\[ w_i = \log[(1 - \epsilon_i)/\epsilon_i] \]  

(10)

Using \( d \) copies of a data packet, the decision rule on each bit is thus

\[ \min_x \sum_{j=1}^{d} \log[(1 - \epsilon_i)/\epsilon_i] \gamma_j^{(i)} \oplus x \]  

(11)

where \( \gamma_j^{(i)} \) is the symbol received in channel state \( i \) to the \( j \)-th repetition of symbol \( x \) (\( x=0 \) or 1). Note that for \( \epsilon_0 = \epsilon_1 \), (11) becomes a simple majority voting rule.

Let \( d_0 \) copies among \( d \) be transmitted in channel state 0, and the other \( d_1 = d - d_0 \) copies be transmitted in channel state 1. Let us assume that for a given position, among the \( d_0 \) repeated bits \( e_0 \) are received in error and among the other \( d_1 \) repeated bits \( e_1 \) are received in error. Define the ration \( w \) as

\[ w = \frac{\log[(1 - \epsilon_1)/\epsilon_1]}{\log[(1 - \epsilon_0)/\epsilon_0]} \]  

(12)

then from (11), and using (12), the bit error rate with \( d \) repetitions, \( P_d(d_0, d_1) \), is given by

\[ P_d(d_0, d_1) = Pr\{d_0 + d_1 w \leq 2(\epsilon_0 + \epsilon_1 w)\} \]  

(13)

Thus,

\[ P_d(d_0, d_1) = \sum_{e_0} \left( \begin{array}{c} d_0 \\ e_0 \end{array} \right) \epsilon_0^{e_0}(1 - \epsilon_0)^{(d_0-e_0)} \sum_{e_1} \left( \begin{array}{c} d_1 \\ e_1 \end{array} \right) \epsilon_1^{e_1}(1 - \epsilon_1)^{(d_1-e_1)} \]  

(14)

where the summation in (14) is taken over all possible combinations of \( e_0 \) and \( e_1 \) which fulfil the inequality in (13). For the combinations of \( e_0 \) and \( e_1 \) yielding to the equality in (13), a factor of 1/2 has to be added in (14). Note that for \( d_0 = 0 \) or \( d_1 = 0 \), corresponding to receiving all \( d \) bits in channel state 0 or 1, (14) becomes equivalent to the usual bit error rate using a majority voting rule on \( d \) repetitions over a BSC with cross-over probability \( \epsilon_0 \) or \( \epsilon_1 \).
Scheme 3

Scheme 3 is a generalized type II hybrid ARQ scheme that uses a family of rate compatible convolutional (RCC) codes [13]-[15]. The RCC codes are obtained from a starting best known high rate \((V - 1)/V\) punctured convolutional code derived from a best known rate 1/2 code [15]. The construction of the family of RCC codes is as follow: Starting with the high rate \((V - 1)/V\) punctured convolutional code, lower rate compatible codes are obtained by simply completing the \((V - 2)\) bits that have been initially deleted from the original rate 1/2 code to form the starting rate \((V - 1)/V\) code, a few at a time, sequentially, up to rate \((V - 1)/2(V - 1) = 1/2\) code. Then, from the rate \((V - 1)/2(V - 1)\) code, lower rate compatible codes are obtained, by simply duplicating the \(2(V - 1)\) code bits, a few at a time, sequentially, without any limit [15]. The obtained codes of rates less than 1/2 are repetition codes [15]. A repetition code can be represented by a matrix called "repetition matrix" [15]. A repetition matrix has the same dimensions as a perforation matrix. However, each element of a repetition matrix is equal or greater than one, indicating the number of duplications of the corresponding code symbol. As an example, consider the starting rate 3/4 punctured convolutional code of memory \(m = 6\) and perforation matrix \(P_0\) given [20] by

\[
P_0 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
\]  

(15)

With an increment of one bit at a time, the RCC codes of rates 3/5 and 3/6 obtained from the above code have perforation matrices \(P_1\) and \(P_2\), given [15] respectively by

\[
P_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}
\]  

(16)

\[
P_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\]  

(17)
The code of rate 3/6 corresponds to the original rate 1/2 code of memory \( m = 6 \) which has been punctured to form the starting rate 3/4 code. From this code, lower RCC codes are obtained by simply duplicating the 6 code bits, sequentially, a few at a time, without any limit. To illustrate, the repetition matrices \( Q_3 \) and \( Q_4 \) for rates 3/7 and 3/8 repetition codes are respectively given \([15]\) by

\[
Q_3 = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{(18)}
\]

\[
Q_4 = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{(19)}
\]

Let \( C_n, n=1,2,... \), denote the RCC codes of decreasing rates to be used in the generalized type II ARQ scheme. To each \( k \) bit data packet to be transmitted, \( n_p \) parity bits for error detection are appended together with \( m \) known bits corresponding to the memory of the convolutional encoder. The sequence of \( L = (k + n_p + m) \) is then encoded with the original rate 1/2 encoder. The packet is said to be at \textbf{level 1} and only code bits in the starting code \( C_1 \) are sent to the receiver. At the receiver end, after decoding for error correction using the Viterbi algorithm with code \( C_1 \), error detection operation is performed. If the presence of errors is detected, the receiver informs the transmitter via the feedback channel and the packet moves to \textbf{level 2}. At \textbf{level 2}, incremental code bits yielding code \( C_2 \) are transmitted and decoding is started again using code \( C_2 \). Should the presence of errors still remain, then the packet moves to \textbf{level 3} and additional incremental bits yielding code \( C_3 \) are provided by the transmitter. This procedure of transmitting incremental redundancy bits according to the family of RCC codes is continued until decoding finally succeeds. With a starting perforation matrix of the form,

\[
P_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{(20)}
\]

which means that at \textbf{level 1}, one of the two sequences at the output of the two
adders of the rate $1/2$ encoder is sent to the receiver, the above ARQ scheme becomes equivalent to the type II ARQ scheme [8] with code combining [10].

Here again, the decoder weights each received sequence by a reliability factor. At a given level $n$ of the generalized type II hybrid ARQ scheme, the sequence used for the decoding is obtained by interlacing the initial received encoded sequence in $C_1$ with the $(n-1)$ subsequent received sequences yielding code $C_n$. Let this sequence be of length $N$ bits and be denoted $Y$. Let $N_0$ bits among $N$ be received while the channel is in state $0$, and the other $N_1 = N - N_0$ bits be received while the channel is in state 1. A maximum-likelihood decoder (Viterbi decoder) will select the codeword (data packet) $X^{(k)}$ which maximizes the likelihood function. This function can be written as

$$ \max_k \{ P[Y|X^{(k)}] = (1 - \epsilon_0)^{(N_0-d_0^{(k)})/\epsilon_0} (1 - \epsilon_1)^{(N_1-d_1^{(k)})/\epsilon_1} \} \quad (21) $$

where $d_0^{(k)}$ and $d_1^{(k)}$ are the number of bit disagreements among respectively the $N_0$ and $N_1$ bits for codeword $X^{(k)}$. Taking the logs of both sides yields

$$ \max_k \{ \log P[Y|X^{(k)}] = N_0 \log(1-\epsilon_0) + N_1 \log(1-\epsilon_1) - d_0^{(k)} \log[(1-\epsilon_0)/\epsilon_0] - d_1^{(k)} \log[(1-\epsilon_1)/\epsilon_1] \} \quad (22) $$

Finally by dropping the terms which are not functions of $k$, we obtain

$$ \max_k \{ \log P[Y|X^{(k)}] \propto \min_k \{ d_0^{(k)} \log[(1-\epsilon_0)/\epsilon_0] + d_1^{(k)} \log[(1-\epsilon_1)/\epsilon_1] \} \} \quad (23) $$

An alternative way to write (23) is

$$ \min_k \sum_{j=0}^{N} \log[(1-\epsilon_j)/\epsilon_j] y_j^{(f)} \oplus x_{kj} \quad (24) $$

1
where \(i\) (\(i = 0\) or \(1\)) indicates the channel state for the transmitted symbol \(x_{k,i}\) which is received as \(y_{j}^{(i)}\). As would be expected, the weighting reliability factor with Scheme 3 is \(\log[(1 - \epsilon_i)/\epsilon_i]\), as in Scheme 2.

Let an encoded sequence in \(C_1\) be transmitted in channel state \(s_1\), and let the \((n - 1)\) additional sequences yielding code \(C_n\) be respectively transmitted in channel states \(s_2, s_3, ..., s_n\). Following [21], the probability of a decoding error event of Viterbi decoding with code \(C_n\), \(P(E(n)|s_1, s_2, ..., s_n)\), given that the channel states for the \(n\) successive transmissions are respectively \(s_1, s_2, ..., s_n\), can be upper bounded as

\[
P(E(n)|s_1, s_2, ..., s_n) \leq \sum_{d=d_{\text{free}}^{(n)}}^{\infty} \sum_{d(1)=0}^{d_1} \sum_{d(2)=0}^{d_2} \sum_{d(n)=0}^{d_n} a_d^{(n)}(d(1), d(2), ..., d(n)) \cdot P_d(\sum_{s_j=0} d(j), \sum_{s_j=1} d(j))
\]

In (25), \(d_{\text{free}}^{(n)}\) is the free distance of code \(C_n\), and \(a_d^{(n)}(d(1), d(2), ..., d(n))\) is the number of paths of weight \(d = (d(1) + d(2) + ... + d(n))\), where \(d(1)\) is the contribution of code \(C_1\), \(d(2)\) is the contribution of the added symbols to \(C_1\) yielding code \(C_2\), \(d(3)\) is the contribution of the added symbols to \(C_2\) yielding code \(C_3\) and so on. The term \(P_d(\sum_{s_j=0} d(j), \sum_{s_j=1} d(j))\) in (25) is the probability that a wrong path at distance \(d\) from the correct path is selected, where \(\sum_{s_j=0} d(j)\) is the total number of positions among the \(d\) positions that are transmitted in channel state 0, and \(\sum_{s_j=1} d(j)\) is the total number of positions transmitted in channel state 1. \(P_d(\sum_{s_j=0} d(j), \sum_{s_j=1} d(j))\) is equivalent to \(P_d(d_0, d_1)\), given by (14), with \(\sum_{s_j=0} d(j) = d_0\) and \(\sum_{s_j=1} d(j) = d_1\). A computer program to determine for any given RCC code \(C_n\) the weight spectra coefficients \(a_d^{(n)}(d(1), d(2), ..., d(n))\) has been developed. Thus, for any set of transmission channel states, and for any given RCC code \(C_n\), the bound (25) on the probability of a decoding error event \(P(E(n)|s_1, s_2, ..., s_n)\) can be calculated.
4 Throughput analysis

In this section, the throughputs of the three ARQ schemes described above are determined. The throughput \( \eta \) is defined as the average number of information bits accepted by the receiver per transmitted channel symbol. It will be assumed that the undetected error probability is negligible.

Let \( S(n)|s_1, s_2, \ldots, s_n \) denote the event \{ decoding success at level \( n \) of the ARQ scheme, i.e. after receiving \( n \) sequences for the same data packet, given that the successive channel states for the \( n \) transmissions are respectively \( s_1, s_2, \ldots, s_n \}, and let \( F(n)|s_1, s_2, \ldots, s_n \) denote the event \{ decoding failure at level \( n \} \).

The average number of transmissions per correctly received packet, given the set of channel states for the successive transmissions are respectively \( s_1, s_2, \ldots, s_n, \ldots \), is given by

\[
E[Tr|s_1, s_2, \ldots, s_n, \ldots] = 1 + Pr\{F(1)|s_1\} + Pr\{F(1), F(2)|s_1, s_2\} + \ldots
+ Pr\{F(1), F(2), \ldots, F(n)|s_1, s_2, \ldots, s_n\} + \ldots
\]

By averaging (26) over all possible sets of transmission channel states, \( E[Tr] \) can be obtained, and is given by

\[
E[Tr] = 1 + \sum_{s_1=0}^{1} P_{s_1} Pr\{F(1)|s_1\}
+ \sum_{s_1=0}^{1} \sum_{s_2=0}^{1} P_{s_1} P_{s_1^{(b)}} Pr\{F(1), F(2)|s_1, s_2\} + \ldots
+ \sum_{s_1=0}^{1} \sum_{s_2=0}^{1} \ldots \sum_{s_n=0}^{1} P_{s_1} P_{s_1^{(b)}} \ldots P_{s_{n-1}^{(b)}} Pr\{F(1), F(2), \ldots, F(n)|s_1, s_2, \ldots, s_n\} + \ldots
\]

In (27), \( P_{s_1} \) is the probability that the channel starts in state \( s_1 \), \( P_{s_1} = p_1 \) if \( s_1 = 1 \), and \( 1 - p_1 \) if \( s_1 = 0 \), and \( P_{s_{j-1}^{(b)}, s_j} \) is the channel \( b \)-step transition probability from state \( s_{j-1} \) to state \( s_j \), which is given by (6), (7), (8) or (9), depending on the values of \( s_j \) and \( s_{j-1} \).
Scheme 1

For Scheme 1, each term \( Pr\{F^{(1)}, F^{(2)}, \ldots, F^{(n)}|s_1, s_2, \ldots, s_n\} \), for \( n = 1, 2, \ldots \), reduces to

\[
Pr\{F^{(1)}, F^{(2)}, \ldots, F^{(n)}|s_1, s_2, \ldots, s_n\} = Pr\{R_d|s_1\} Pr\{R_d|s_2\} \cdots Pr\{R_d|s_n\}, n = 1, 2, \ldots
\]  

(28)

where \( R_d|s_n \) is the event that the \( n \)-th received copy contains errors, given that the channel state for the transmission of that copy is \( s_n \),

\[
Pr\{R_d|s_n\} = \begin{cases} 
1 - (1 - \epsilon_0)^L & \text{if } s_n = 0 \\
1 - (1 - \epsilon_1)^L & \text{if } s_n = 1 
\end{cases}
\]  

(29)

Let the vector \( V \) be defined as

\[
V = \begin{bmatrix}
P_0 Pr\{R_d|0\} \\
P_1 Pr\{R_d|1\}
\end{bmatrix}
\]  

(30)

and let a row vector of ones be denoted as \( \mathbf{1} = [1, 1] \). Let the matrix \([G]\) be defined as

\[
[G] = \begin{bmatrix}
P_0^{(b)} Pr\{R_d|0\} & P_0^{(b)} Pr\{R_d|0\} \\
P_1^{(b)} Pr\{R_d|1\} & P_1^{(b)} Pr\{R_d|1\}
\end{bmatrix}
\]  

(31)

Using (30) and (31), (28) can be expressed as

\[
E[Tr] = 1 + \mathbf{1} \sum_{n=1}^{\infty} [G]^n V
\]  

(32)

Since \([G]\) is a stochastic matrix, (32) reduces to

\[
E[Tr] = 1 + \mathbf{1}[I_{2x2} - [G]]^{-1} V
\]  

(33)

where \( I_{2x2} \) is a two by two identity matrix. The throughput for Scheme 1 is given by

\[
\eta = \frac{1}{E[Tr]} \frac{k}{k + n_p}
\]  

(34)
Scheme 2

For Schemes 2 and 3, the evaluation of $E[Tr]$ given by (27) is complicated by the statistical dependencies among $\{F^{(1)}, F^{(2)}, F^{(3)}, \ldots, F^{(n)}|s_1, s_2, \ldots, s_n\}$. It can be shown [15] that

$$
\begin{align*}
Pr\{F^{(1)}|s_1\}, Pr\{F^{(2)}|s_1, s_2\}, \ldots, Pr\{F^{(n)}|s_1, s_2, \ldots, s_n\} & \leq \\
Pr\{F^{(1)}, F^{(2)}, \ldots, F^{(n)}|s_1, s_2, \ldots, s_n\} & \leq \\
Pr\{F^{(n)}|s_1, s_2, \ldots, s_n\}
\end{align*}
$$

(35)

Substituting the two bounds (35) in (27), a lower and an upper bound on $E[Tr]$ for Scheme 2 can be obtained. Each term $Pr\{F^{(j)}|s_1, s_2, \ldots, s_j\}$ in (35) for $j=1,2,\ldots$, is given by

$$
Pr\{F^{(j)}|s_1, s_2, \ldots, s_j\} = 1 - (1 - P_{n_j}(d_0, d_1))^L
$$

(36)

where $d_0$ and $d_1$ are the number of copies among $n_j$, received in respectively channel states 0 and 1. $P_{n_j}(d_0, d_1)$ is given by (14), with $n_j = d$.

The throughput of Scheme 2 is also given by (34). Using the two bounds on $E[Tr]$, a lower and an upper bound on $\eta$ can be obtained.

Scheme 3

For Scheme 3, the probability of a decoding failure at level $j$ can be bounded [15] as

$$
Pr\{F^{(j)}|s_1, s_2, \ldots, s_j\} \leq 1 - (1 - P(E^{(j)}|s_1, s_1, \ldots, s_j))^l
$$

(37)

where $P(E^{(j)}|s_1, s_1, \ldots, s_j)$ is the probability of a decoding error event of Viterbi decoding with code $C_j$ given by (25), and where $l$ in the number of trellis levels ($l = (k + n_p)/(V - 1)$).

Assuming that the number of incremental redundancy bits per each $(V - 1)$ information bits provided by the transmitter at the successive transmissions for the same data packet is equal to $h$, then the throughput $\eta$ is given by
\[
\eta = \frac{V - 1}{V + h(E[T_r] - 1)} \frac{k}{k + n_p + m}
\]  

(38)

Using the lower bound on \( Pr\{F^{(j)}|s_1, s_2, ..., s_j\} \) given by (37), and the upper bound on \( P(E^{(j)}|s_1, s_1, ..., s_j) \) given by (25), a lower bound on the throughput of the generalized type II hybrid ARQ scheme can be calculated.

For the special case of a type II hybrid ARQ scheme [8]-[10], the throughput expression is given by

\[
\eta = \frac{1}{E[T_r]} \frac{k}{k + n_p + m}
\]

(39)

The average number of transmissions \( E[T_r] \) for the conventional type II ARQ scheme can be calculated as in [16]. Note that for the case of a type II ARQ scheme, (with or without code combining), the probability of a decoding failure at level 1, \( Pr\{F^{(1)}|s_1\} \), is given by

\[
Pr\{F^{(1)}|s_1\} = 1 - (1 - \epsilon_{s_1})^{(k+n_p+m)}
\]

(40)

### 5 Numerical Results

We present in this section numerical results of the throughput for the different ARQ schemes described above. Since we are mainly interested in comparing the different ARQ schemes presented in this paper, throughput results are given only for the case when \( p = \hat{p} = p_l \), i.e., a BI channel. Figures 2, 3, 4 and 5 show the throughput as a function of the average BER \( \bar{\epsilon} \), for respectively the values \( p_l = 0.05 \) (dense burst channel), \( p_l = 0.1 \), \( p_l = 0.25 \) (diffuse burst channel), and \( p_l = 1.0 \) (stationary channel), and for a packet length \( L = 1000 \). The code used with the type II hybrid ARQ scheme is of memory \( m = 6 \) and generator polynomials, in octal representation, (133,171) [22]. Two starting punctured convolutional codes of rates 7/8 and 3/4, both with memory \( m = 6 \) [15], are used with the generalized type II hybrid ARQ scheme, with \( h = 1 \). For the calculation of (25), only the first seven elements of the weight
spectra of each code were used in order to obtain an accurate approximation of the probability of a decoding error event of Viterbi decoding. The precision obtained in the computation of the throughput for schemes 2 and 3 ranges from 0.05% to 0.5% for high to low values of the average BER.

In all cases, it can be seen that the two bounds on the throughput of Scheme 2 are quite close. Scheme 2 provides a substantial throughput improvement over Scheme 1 at low average BER. From Figures 2-5, we can see that the range of the average BER for which the use of FEC with ARQ yields an improvement in throughput over ARQ alone increases as $p_1$ increases. As might be expected, the throughput with a type II ARQ scheme is superior than with Scheme 1. The use of code combining with a type II ARQ scheme yields a throughput improvement at low $\tau$ over the conventional type II ARQ scheme, especially for relatively high values of $p_1$ (see Figs. 2-5). The advantages of the generalized type II ARQ scheme are evident from Figures 2-5. With the generalized type II ARQ scheme, the throughput increases as the starting coding rate increases, and as the channel degrades, it tends to merge with the throughput of rate 1/2 type II ARQ scheme with code combining. It should be pointed out that in a practical system, the number of incremental redundancy bits provided by the transmitter at each level of the generalized type II ARQ scheme should be increased as the channel degrades. This reduces the number of transmissions required for a given data packet, minimizing thus buffering overflow events, and also reducing the loss in throughput due to the transmission of overheads.

Figure 6 compares the throughputs of Schemes 1 and 2 for different values of $p_1$. It can be seen that in both schemes, for a given value of $\tau$, the throughput increases as the channel becomes burstier in nature.

Figure 7 shows the throughputs of both the conventional type II and the generalized type II ARQ schemes for different values of $p_1$. It might be noted that under
good channel conditions, the throughput of the type II ARQ scheme is superior to that of the generalized type II ARQ scheme. This is due to the fact that with the generalized type II ARQ scheme, some redundancy bits for error correction are incorporated in the first transmission of every data packet. However, the difference tends to become smaller as the starting coding rate used with the generalized type II increases. Also, by properly choosing an invertible starting code of rate 1, the throughput of the generalized type II ARQ scheme can be made to approach 1 for good channel conditions. As it can be seen from Figure 7, for a given value of the average BER, the throughput of the generalized type II ARQ scheme improves as \( p \) decreases.

6 Conclusions

We have analysed efficient ARQ schemes over a simple nonstationary channel. We have shown that with a simple ARQ scheme that only uses error detection, a throughput improvement can be achieved by allowing the decoder to combine packets received in error with their repeated copies. We have shown that the generalized type II hybrid ARQ scheme performs better that the conventional type II hybrid ARQ scheme, on bursty and stationary channels. With the generalized type II hybrid ARQ scheme, a significant throughput can be achieved over a wide range of channel parameters, and the throughput improves as the channel becomes burstier in nature. These results demonstrate once again the effectiveness of the generalized type II hybrid ARQ scheme over time-varying channels.
References


ACKNOWLEDGMENT

The author would like to thank H. Chow for his help in writing the program to generate the weight spectra of the convolutional codes used in this paper.
List of Figures

1. A tow-state Markov chain nonstationary channel model. 20
2. Throughput of various ARQ schemes for $p_1 = 0.05$. 21
3. Throughput of various ARQ schemes for $p_1 = 0.1$. 22
4. Throughput of various ARQ schemes for $p_1 = 0.25$. 23
5. Throughput of various ARQ schemes for $p_1 = 1.0$. 24
6. Throughput comparison of Schemes 1 and 2 for various values of $p_1$. 25
7. Throughput comparison of the conventional type II and rate 7/8 generalized type II ARQ schemes for various values of $p_1$. 26
Figure 1: A two-state Markov chain nonstationary channel model.
Figure 2: Throughput of various ARQ schemes for $p_1 = 0.05$
Figure 3: Throughput of various ARQ schemes for $p_1 = 0.1$
Figure 4: Throughput of various ARQ schemes for $p_1 = 0.25$
Figure 5: Throughput of various ARQ schemes for $p1 = 1.0$
Figure 6: Throughput comparison of Schemes 1 and 2 for various values of $p_1$
Figure 7: Throughput comparison of the conventional type II and rate 7/8 generalized type II ARQ schemes for various values of $p_1$.