A novel adaptive hybrid ARQ scheme for wireless ATM networks *

Inwhee Joe

Network Research Group, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

This paper describes the design and performance of a novel adaptive hybrid ARQ scheme using concatenated FEC codes for error control over wireless ATM networks. The wireless links are characterized by higher, time-varying error rates and burstier error patterns in comparison with the fiber-based links for which ATM was designed. The purpose of the hybrid ARQ scheme is to provide a capability to dynamically support reliable ATM-based transport over wireless channels by using a combination of our ARQ scheme (called SDLP) and the concatenated FEC scheme. The key ideas in the proposed hybrid ARQ scheme are to adapt the code rate to the channel conditions using incremental redundancy and to increase the starting code rate as much as possible with the concatenated FEC, maximizing the throughput efficiency. The numerical results show that our proposed scheme outperforms other ARQ schemes for all SNR values.

1. Introduction

In recent years, wireless ATM (Asynchronous Transfer Mode) has emerged as a solution for mobile multimedia by supporting ATM-based transport in a seamless manner [15]. The attempt to extend ATM to wireless links immediately identifies a fundamental difference in the way that ATM will be used. That is, ATM will be subject to transmission links that are unreliable wireless links. Because of the fading effects and interference, the wireless links are characterized by burstier error patterns, and higher and time-varying error rates when compared with the reliable fiber-based links for which ATM was designed. As a result, such a difference demands the use of error control schemes to insulate the ATM network layer from wireless channel impairments.

There are two basic categories of error control schemes: ARQ (Automatic Repeat Request) and FEC (Forward Error Correction) schemes. ARQ schemes provide high reliability at good and moderate channel qualities. However, if the channel error rate is high like in a wireless channel, the throughput performance drops rapidly due to the increased frequency of retransmission. In order to counter this effect, hybrid ARQ schemes are used by combining FEC with ARQ schemes. In general, hybrid ARQ schemes are classified into type I and type II schemes. In the type I hybrid scheme, each packet is encoded for both error detection and error correction. At the receiver, the FEC portion first attempts to correct the frequent error patterns. If an uncorrectable error pattern is detected, then the receiver requests a retransmission using ARQ.

Error statistics on the wireless channel are time-varying. Efficient error control for a time-varying channel can be realized with an adaptive coding scheme. The type II hybrid scheme is based on an adaptive coding scheme with two code rates. When the channel condition is good, the high code rate is used for error detection. In response to the retransmission request, redundant parity bits are sent to the receiver and combined with the previous error packet to produce the low code rate for error correction. This scheme is generalized with code combining to offer a large number of code rates instead of two as in the type II hybrid case.

We envision a system architecture for quality-critical traffic (e.g., data or image) over wireless ATM that includes a backbone network consisting of ATM switches, as shown in figure 1. Wireless workstations (mobile or fixed) communicate with stationary hosts through the backbone network where some portions could be wireless. Stationary hosts are connected to the backbone network by wired lines. In this architecture, a wireless link refers to either a link between a wireless workstation and an ATM switch or a link between ATM switches.

As an operational example, the following scenario will apply. User A is the originator of the request for information, residing on a workstation with wireless access to the ATM switch. User B is the source of the information, residing in a stationary host such as a command center. To perform the mission, user A communicates with user B by exchanging a set of multimedia messages.

In this paper, we propose a novel adaptive hybrid ARQ scheme for error control focusing on quality-critical traffic over wireless ATM networks. The hybrid ARQ scheme is based on our ARQ protocol, SDLP, combined with the

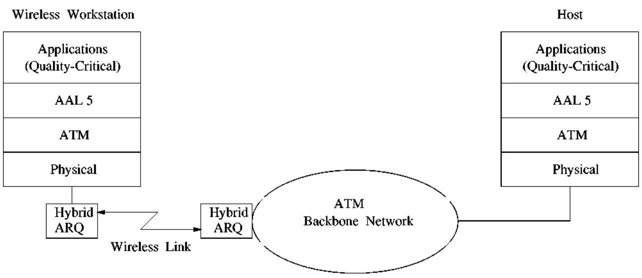

Figure 1. System architecture for quality-critical traffic over wireless ATM.

* This paper was presented in part at the ACM/IEEE MobiCom ’97 Conference, Budapest, Hungary, September 1997.

concatenated FEC scheme. In the next section, we analyze the performance of the concatenated FEC approach for wireless ATM, followed by a detailed description of our ARQ protocol featuring variable packet size and periodic status message. In section 4, we describe the proposed hybrid ARQ scheme which is adaptive to channel conditions using incremental redundancy. In section 5, we present throughput efficiency of the proposed scheme and the numerical results over a wireless channel model. Finally, we conclude the paper by highlighting our contribution.

2. The concatenated FEC approach for wireless ATM

In wireless ATM, channel errors will be typically beyond the error correcting capability of the ATM HEC (Header Error Control), making it necessary for more powerful FEC schemes. Here, we consider two possible FEC approaches that can be used for wireless ATM:

- convolutional coding with Viterbi decoding,
- concatenated FEC scheme.

Convolutional coding provides substantial coding gain with respect to an uncoded system. However, the gain is not sufficient for wireless ATM. Moreover, the Viterbi decoder errors are typically beyond the HEC's correction capability due to the burstiness.

As shown in figure 2, the concatenated FEC scheme consists of a convolutional inner code and an RS (Reed–Solomon) outer code with interleaving [3,17]. The RS code is used as an outer code to correct burst errors from the Viterbi decoder and offers as much performance gain as desired for wireless ATM.

In this section, we analyze the performance of the concatenated scheme on an AWGN (Additive White Gaussian Noise) Rayleigh fading channel, and then compare with convolutional coding under the same conditions. In particular, a rate-$1/2$, constraint-length 7 convolutional code with QPSK (Quadrature Phase Shift Keying) modulation is used as an inner code, since it is an industry standard in digital cellular systems. Since the performance of Viterbi decoding deteriorates in presence of burst errors, ideal interleaving is assumed to spread burst errors resulting from the wireless channel. The probability of bit error at the Viterbi decoder output, $P_{\mathrm{b,Vit}}$, can be derived by using the transfer function $T(\cdot)$ of convolutional codes and the union bound [4]:

$$P_{\mathrm{b,Vit}} \leqslant \frac{1}{2}\frac{\partial T(D_1, D_2, I)}{\partial I}\bigg|_{I=1}, \tag{1}$$

where

$$D_1 = \frac{1}{1 + E_\mathrm{b}/2N_0} \quad \text{and} \quad D_2 = \frac{1}{1 + E_\mathrm{b}/N_0}.$$

If this convolutional coding scheme is used below the ATM layer, then the resulting probability of bit error at the Viterbi decoder output, $P_{\mathrm{b,Vit}}$, is the input probability of bit error to the ATM HEC function. Assuming that ideal interleaving randomizes error bursts from the Viterbi decoder, the resulting cell loss probability (CLR) is presented in figure 3. For comparison, the uncoded case for the ATM HEC only is also shown in figure 3. It is obvious that the convolutional codes with Viterbi decoding provide considerable coding gain relative to the uncoded case.

However, their performance is still not sufficient for wireless ATM. When the Viterbi decoder suffers a decoding error, the resulting codeword usually differs from the transmitted word by a few consecutive trellis branches. As a result, although the input to the Viterbi decoder is corrupted by random noise, the output of the decoder tends to have burst errors. This burst will typically be beyond the error correcting capability of HEC, causing a cell loss. Hence, the RS code with its inherent burst-error correcting capability is used to deal with burst errors out of the Viterbi decoder.

Since the constraint length is chosen to be 7 for convolutional coding, the most frequently occurring error bursts at the output of the Viterbi decoder will have length 7 or 8. Thus, 8-bit symbols are chosen for the RS code to handle these burst errors with a single RS symbol. The RS codeword takes the form of (n, k) over GF(2^8), where n is the number of code symbols and k is the number of information symbols. Since the minimum distance of this code

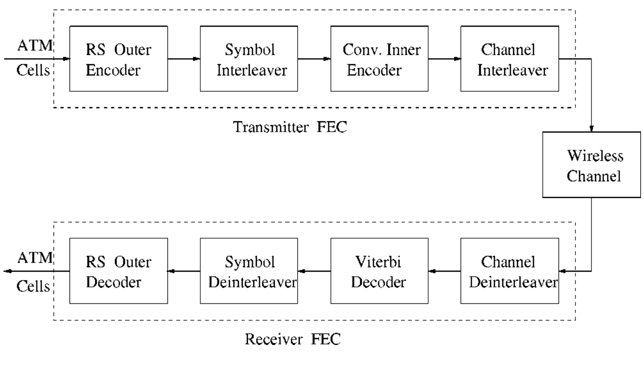

Figure 2. Concatenated FEC scheme for wireless ATM.

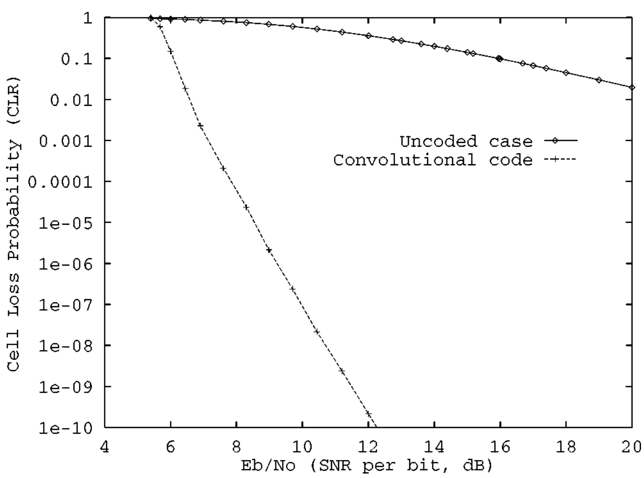

Figure 3. ATM HEC CLR for rate-$1/2$, constraint-length 7 convolutional coding on AWGN Rayleigh fading channel.

is $n - k + 1$, it can correct i symbol errors and j symbol erasures at the same time as long as $(2i + j) \leqslant (n - k)$ is satisfied. The word error rate for a codeword of n symbols at the output of the RS decoder that is capable of correcting up to t symbol errors can be written as

$$P_{\mathrm{w}} = \sum_{2i+j>2t}^{i+j=n} \binom{n}{i,j} e^i s^j (1 - e - s)^{n-i-j}, \quad i, j \geqslant 0, 1, 2, \ldots, \tag{2}$$

where e denotes the symbol error rate and s the symbol erasure rate before the RS decoder. For the concatenated coding scheme, the RS decoder processes the output stream of the Viterbi decoder. In this case, there are no symbol erasures ($s = 0$) and further the symbol error rate e at the input of the RS decoder is the same as the symbol error rate at the Viterbi decoder output, $P_{\mathrm{s,Vit}}$. With ideal symbol interleaving between the Viterbi decoder and the RS decoder, equation (2) can be simplified to

$$P_{\mathrm{w}} = \sum_{i=t+1}^{n} \binom{n}{i} P_{\mathrm{s,Vit}}^i (1 - P_{\mathrm{s,Vit}})^{n-i}. \tag{3}$$

When i (more than t) symbol errors occur, the RS decoder cannot correct them due to a decoder failure. Therefore, the symbol error rate after the RS decoder can be estimated by noting that i out of n symbols in an RS codeword will be delivered in error on a decoder failure. Furthermore, if about half of 8 bits in an errored RS symbol are assumed to be in error, then the probability of bit error at the RS decoder output is approximated as

$$P_{\mathrm{b,RS}} \approx \frac{1}{2} \sum_{i=t+1}^{n} \frac{i}{n} \binom{n}{i} P_{\mathrm{s,Vit}}^i (1 - P_{\mathrm{s,Vit}})^{n-i}. \tag{4}$$

Since one RS symbol consists of 8 bits from the Viterbi decoder, the symbol error rate at the Viterbi decoder output, $P_{\mathrm{s,Vit}}$, can be bounded as

$$P_{\mathrm{s,Vit}} < 8P_{\mathrm{b,Vit}}. \tag{5}$$

The symbol errors at the input of the RS decoder are assumed to be independent by using a symbol interleaver between two decoders, assuring that a given error burst from the Viterbi decoder affects no more than one symbol in an RS codeword. The evaluated results from equation (4) are presented in figure 4 for the concatenated code of a rate-$1/2$, constraint-length 7 convolutional inner code and an $(n, 53)$ RS outer code for various error correcting capabilities ($t = 2$, 3, 4, 5, 8 symbols) on an AWGN Rayleigh fading channel with QPSK modulation. For comparison, the convolutional code alone case ($t = 0$) with the rate-$1/2$ and constraint-length 7 is also shown in figure 4. The information size of the RS code is chosen to be 53 symbols, which corresponds to one ATM cell. The results show that the more error-correcting capability, the steeper performance curve with more coding gain, and consequently the concatenated coding scheme can offer sufficient error performance for wireless ATM. For example, with an

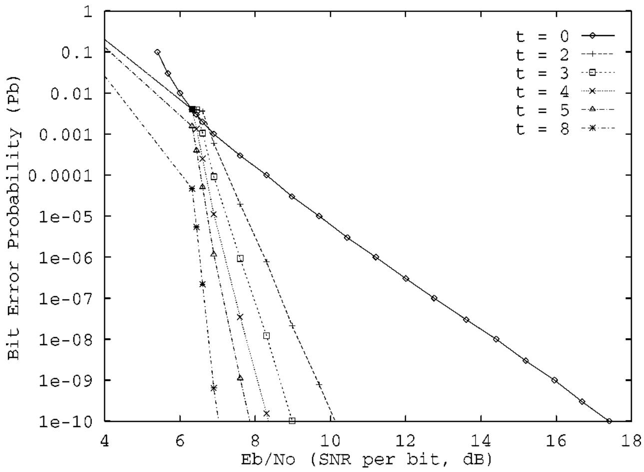

Figure 4. BER performance of the concatenated scheme on AWGN Rayleigh fading channel.

8-correcting RS code the coding gain is about 10.4 dB at the bit error rate of 10^{-10} compared to the convolutional code alone case ($t = 0$) and the overall code rate is 0.384 (i.e., $1/2 \cdot 53/(53 + 16)$).

Based on the performance evaluation, the concatenated scheme has been selected as our FEC approach for wireless ATM.

3. The new ARQ scheme for wireless links: SDLP

Recently, several ARQ protocols have been proposed for reliable data transport. In 1992, ETSI (European Telecommunications Standards Institute) developed a new ARQ protocol, RLP (Radio Link Protocol), for GSM (Global System for Mobile Communications) [5], which is based on the timer-based ARQ scheme, leading to redundant retransmissions. Secondly, another protocol called AIRMAIL for asymmetric wireless links has been presented in [2] with lower throughput due to the fixed packet length.

Finally, the SSCOP (Service Specific Connection Oriented Protocol) has been released for ATM networks [7]. It uses a new ARQ protocol based on the selective-repeat (SR) ARQ protocol. Even if the packet length is variable in SSCOP, it does not apply the concept of the optimal packet length in the sense of maximizing the throughput efficiency. Moreover, it eliminates the timeout mechanism by using three different packet types, i.e., periodic polling packet from the transmitter, acknowledgment (ACK) packet, and negative acknowledgment (NAK) packet, while our SDLP protocol combines these three types of packets into one periodic status message from the receiver, which is simpler to implement.

Here we propose an efficient data link protocol, based on the retransmission to provide reliable ATM-based transport in response to fluctuating lossy links.

3.1. Protocol description

The "variable packet size" and "periodic status message" are the key ideas in our new ARQ protocol, which minimizes the processing overhead and maximizes the throughput efficiency over wireless links with high and variable error rates. The throughput efficiency is very sensitive to the packet size, i.e., when a packet is too large, there is an increased possibility of retransmission, while a small packet is inefficient because of the large overhead bits with respect to the actual data. Therefore, the packet size should be chosen adaptively based on the time-varying conditions of the wireless channel (refer to [1] for details).

In timer-based ARQ schemes, retransmissions are triggered by both transmission timers and in response to explicit NAKs from the receiver. Since these two events occur independently, the efficiency of these protocols suffers due to redundant retransmissions, especially when the timer value is small [12]. All classical ARQ schemes have transmission timers, resulting in redundant retransmissions and longer recovery from the timeout mechanism, not to speak of the complexity of maintaining timers. Instead, the SDLP protocol relies on periodic status messages from the receiver, preventing redundant retransmissions.

The idea of "periodic status message", where the receiver sends its status to the transmitter on a periodic basis as in the SNR protocol [13] instead of using explicit acknowledgments or NAKs, simplifies the protocol and further eliminates the dependence on the error prone medium. Even when the receiver loses packets due to the channel errors, it can still send periodic status messages because they are sent periodically by a local timer and not by the event of receiving a packet. Likewise, although a status message gets lost, a subsequent message will always follow.

3.1.1. Data packet format

The format of the data packet is shown in figure 5(a). *TI* is the traffic type indicator. Since the protocol is expected to carry different types of traffic with different requirements, the *TI* field (e.g., 2 bits) is used to indicate whether a packet is data or control, and its traffic type such as quality critical (e.g., data and still images) or time critical traffic (e.g., voice and video). The advantage of including *TI* in the header is that the link layer can identify the traffic type without reference to ATM virtual circuit (VC) level information. Each data packet has a sequence number *seq* (e.g., 8 bits) in order to detect out-of-sequence packets at the receiver. *SC* is the segment counter (e.g., 4 bits) used for segmentation and reassembly of the ATM cells. *HO* (e.g., 1 bit) is used to indicate the packets being sent during handoff. When there is no more data to transmit, the end of text (*EOT*) bit [2] is set in the last packet. The *payload* field contains information bits of the variable size passed down from the ATM layer. The 2-byte *CRC* field provides error detection for the data packet.

2 (bits)	8	4	1	1		16
TI	seq	SC	HO	EOT	Payload (variable size)	CRC

(a) Data Packet Format

2 (bits)	8	4		16
TI	MSN	m	BMAP	CRC

(b) Control Packet Containing Receiver's State

Figure 5. SDLP packet formats.

3.1.2. Control packet format

Status messages are sent from the receiver to the transmitter through control packets on a periodic basis using a local timer. Especially when a packet with the *EOT* bit set is received, the receiver sends its status message immediately without waiting until the next transmission point. This allows for a fast response when small amounts of data are being sent. The period of sending status messages should be chosen so as not to waste much bandwidth for control packets, while at the same time offsetting the effect of a noisy channel.

As shown in figure 5(b), the control packet has the following fields: *TI* is the traffic type indicator, *MSN* is the maximum sequence number of the packet below which every packet has been received correctly at the receiver, m denotes the current frequency of control packets, *BMAP* is the bit map indicating packets outstanding between *MSN* and $MSN + W - 1$, where W is the window size, and *CRC* is used for error detection.

3.1.3. Control packet transmissions

Let T_i be the time interval in which the control packets will be sent from the receiver:

$$T_i = \max\left\{\frac{\mathrm{RTD}(L_i)}{m_i}, \delta\right\}, \quad i = 0, 1, 2, \ldots, \tag{6}$$

where RTD dependent on L_i is an estimate of the round trip delay between sender and receiver, L_i is the variable packet length in bits, m_i is a parameter used to adjust the time interval of control packet transmission according to the channel condition, and δ is the average packet interarrival time on the wireless channel. The round trip delay RTD consists of four components:

$$\mathrm{RTD}(L_i) = T_{\mathrm{tx}}(L_i) + 2T_{\mathrm{prop}} + T_{\mathrm{proc}}(L_i) + T_{\mathrm{cp}}, \tag{7}$$

where T_{tx} is the transmission time of the data packet dependent on the packet length L_i and computed by

$$T_{\mathrm{tx}}(L_i) = \frac{L_i}{R} \tag{8}$$

with the data rate R in bits/s of the wireless channel. T_{prop} is the propagation delay on the wireless link (measurable), T_{proc} is the processing time of the data packet which includes codec and interleaving (measurable), and T_{cp} is the transmission time of the control packet. Once the packet length L_i is determined based on the channel condition, the round trip delay RTD is fixed and can be obtained from equation (7).

The initial value m_0 is determined by considering a tradeoff between bandwidth and response time at the worst case of the channel bit-error-rate (BER), thereby setting the initial conditions of m_i and channel BER. In order to adjust the value T_i to the varying channel condition, m_i will be recalculated periodically, based on the following two factors: channel activity and channel BER. The calculation period is chosen to be RTD, because m_i is the value defined per RTD and hence can be reinitialized every RTD unit of time. For channel activity, one bit is used to indicate whether the channel is active or not by counting the number of packets received during the last RTD interval. If no packets arrive, the channel activity bit is set to 0. The ultimate goal of the T_i adjustment is to minimize the bandwidth to be wasted by control packets when the channel is idle or when the channel BER becomes better. In particular, when the channel condition improves, the frequency of control packets can be reduced by acknowledging as many packets as possible at a time. To increase the control packet interval T_i when the channel is idle or the channel condition gets better, m_i is used to slow down the frequency of control packets, i.e.,

$$T_{i+1} = \max\left\{\frac{\mathrm{RTD}(L_{i+1})}{m_{i+1}}, \delta\right\}, \tag{9}$$

where $m_{i+1} = m_i/2$. However, T_i is never increased beyond a certain maximum:

$$T_i \leqslant \max\left\{\frac{\mathrm{RTD}(L_i)}{\min(m_i)}, \delta\right\}, \tag{10}$$

where $\min(m_i)$ is the minimum allowable value of m_i. This constraint on the maximum value of T_i is necessary to avoid deadlocks even under worst case channel error failures and to guarantee normal protocol operation. Since m_i represents how many control packets to send in a round trip delay, the minimum value of m_i will be one so that the transmitter can have at least one acknowledgment within a window interval for advancing the window.

On the other hand, activity on the channel or the degraded channel condition makes the control packet interval T_i return to the original state by restoring the initial value m_0. For time critical traffic, T_i can be also reduced to obtain a better response time. In that case, we should consider a tradeoff between the frequency of status messages and the expensive bandwidth.

In order for the receiver to control the flow of information from the transmitter, the sliding window mechanism is used in our protocol. The window size W is measured in packets:

$$W \geqslant \max\left(\frac{\mathrm{RTD}(L_i)}{T_{\mathrm{tx}}(L_i)}\right). \tag{11}$$

Even if the round trip delay RTD and the transmission time T_{tx} depend on the packet length L_i, the window size will be fixed, based on the worst case calculation, in order to make the implementation simpler. The buffer size should be larger than the window size to accommodate newly arriving data packets during multiple retransmissions at the transmitter and to take care of out-of-sequence packets at the receiver. Usually, the buffer size is a multiple of the window size. Once the window size is obtained from equation (11), the buffer size and the *BMAP* size can be determined accordingly. Given a sequence number modulus N, the upper bound of the window size can be expressed by

$$W \leqslant \frac{N}{2}. \tag{12}$$

Since the 8-bit sequence number is used in the packet format, it leads to the sequence number modulus $N = 256$ and the upper bound on the window size comes to 128 in our protocol.

A state table for all outstanding packets is maintained at the transmitter and is updated whenever a control packet arrives from the receiver. The entries in the state table consist of *State*[*seq*, *count*, *ack*], where *seq* is the sequence number of an outstanding packet, *count* is the retransmission count, and *ack* indicates whether a packet has been acknowledged or not. When a packet is transmitted, the initial value of retransmission *count* for that packet is set to

$$\textit{count} = \lceil m_i \rceil, \tag{13}$$

where m_i comes from the latest received control packet. Every time a control packet is received, the transmit buffer is compared with the received *BMAP* field. Packets that have been acknowledged are removed from the state table. At the same time, the retransmission *count* of all other packets in the state table is decreased by one, because it is expressed in the unit of the control time interval. A packet is retransmitted only if the retransmission *count* goes to zero, because it means that the round trip delay has elapsed since a particular packet was transmitted. Since retransmission has priority over transmission of a new packet, the retransmission packet is sent immediately with the retransmission *count* reset to the initial value.

4. The adaptive hybrid ARQ scheme with concatenated FEC

Since the convolutional code is used as an inner code in the concatenated FEC scheme, code combining can easily be implemented with incremental redundancy by imposing the rate compatibility condition on the convolutional code. For wireless ATM, we propose an efficient hybrid ARQ scheme adapting to channel conditions based on the incremental redundancy [8].

4.1. The rate compatible convolutional codes

A family of convolutional codes is defined as *rate compatible* [9] if all the code bits of the higher rate codes are included in the lower rate codes. These rate compatible convolutional (RCC) codes are obtained from an original rate-$1/2$ convolutional code by using "rate tables" for puncturing or repetition. The rate table indicates which of the bits are to be punctured or repeated prior to transmission for each code rate. Each entry of a rate table is zero or a positive integer. A zero denotes puncturing and a positive integer denotes the number of repetitions of the corresponding code bit. These tables operate on each group periodically with a period of P to generate a family of RCC codes with code rates

$$R = \frac{P}{P+i}, \quad i = 0, 1, 2 \ldots . \tag{14}$$

Each table consists of two rows and P columns. The highest rate RCC code (i.e., P/P) is obtained from a rate-$1/2$ convolutional code by puncturing P bits from each group of $2P$ code bits according to the rate table. For example, the rate table A_0 of a rate-$8/8$ punctured convolutional code, obtained from a rate-$1/2$ code with constraint-length 7, is given by [18]

$$A_0 = \begin{bmatrix} 1\,1\,1\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,1 \end{bmatrix}, \tag{15}$$

where P is chosen to be eight bits for byte processing.

Starting with the rate P/P punctured code, RCC codes of lower rates can be obtained by simply inserting bits at the punctured bit positions. When all P bits are inserted, the original rate-$1/2$ code is recovered. As an example, an RCC code of rate-$8/12$ can be obtained by adding four more bits to the A_0 table above according to the following rate table [9]:

$$A_1 = \begin{bmatrix} 1\,1\,1\,1\,0\,1\,0\,1 \\ 1\,0\,1\,0\,1\,1\,1\,1 \end{bmatrix}. \tag{16}$$

Similarly, RCC codes of rates lower than $1/2$ can be obtained by repeating some bits specified by the rate table, depending on a given code rate. There is no limit to the lowest achievable code rate with RCC codes. In summary, RCC codes can provide an efficient way of implementing incremental redundancy using the same encoder/decoder. Moreover, these codes can also provide variable rate error protection through a large selection of code rates, which are particularly useful for multimedia applications with different QoS (Quality of Service) requirements.

4.2. The hybrid ARQ scheme

The proposed hybrid scheme is not to repeat information bits in response to the retransmission request, but to transmit additional parity bits only using incremental redundancy of RCC codes until the combined code is powerful enough for error correction. The key idea of the scheme is to adapt the code rate to the channel conditions, maximizing the throughput efficiency. The starting code rate and the number of incremental redundancy bits are to be chosen according to the channel condition at that time. Since the concatenated FEC scheme is used here and RS codes have small overhead, RS codes will be included as much as possible so as to reduce the large overhead of the rate-$1/2$ convolutional code. As a result, when compared with the convolutional coding only case, the starting code rate of the concatenated FEC scheme increases at a given channel condition, leading to higher throughput.

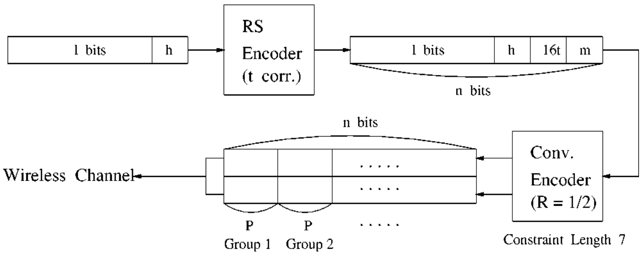

Figure 6. Encoding scheme with concatenated FEC.

The hybrid ARQ scheme with the concatenated FEC scheme is similar to the one proposed by Kallel [11]. The concatenated FEC scheme is combined with our own ARQ protocol, SDLP, given in section 3, to yield the hybrid ARQ scheme. Let the window size be W and the receiver buffer size be sW, where s is a positive integer. As shown in figure 6, to each l-bit information payload, h control bits are added per packet, including traffic type and *CRC* bits, $16t$ bits from the RS outer encoder, and then m tail bits for trellis termination, where m is the memory size of the convolutional encoder. The error correcting capability t of RS codes is determined such that the starting code rate R_0 of the convolutional encoder increases as much as possible under the current channel condition, maximizing the throughput efficiency. The sequence of n bits $(l + h + 16t + m)$ is then encoded with the rate-$1/2$, constraint-length 7 convolutional encoder and transmitted using the following procedure. Let r_i be the number of incremental redundancy bits at step i.

Procedure

Step 0. Given the starting code rate $R_0 = P/(P + r_0)$, $r_0 \geqslant 0$, the encoded sequence is transmitted according to the rule defined by the rate table of R_0 described in section 4.1. At the receiver, the received sequence is decoded by the Viterbi decoder then by the RS decoder, and checked with the *CRC* bits for error detection. If there is no error in the sequence, then the procedure stops. Otherwise, a retransmission request is sent to the transmitter through the control packet with the corresponding bit off in the *BMAP* (figure 5), and the procedure moves to step 1.

Step i, $1 \leqslant i < s$. In response to the retransmission request, only incremental redundancy bits r_i are transmitted at this step i using the rate table with the code rate $R_i = P/(P + r_0 + \ldots + r_i)$, $r_i \geqslant 1$. The size of r_i is determined based on the channel condition at this point. There are two cases to be considered. First, if the channel condition becomes worse compared to the previous channel condition, r_i will be chosen to adapt to the current channel condition as shown in the case I of figure 7. On the other hand, if the channel condition is still the same or unusually becomes better, r_i

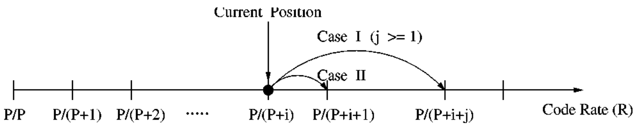

Figure 7. Adaptive code rate to the channel condition using incremental redundancy.

will be set to the minimum incremental level, which is just one bit as shown in the case II of figure 7. This is because adapting to the channel condition does not give any redundancy bits to transmit. At the receiver, the redundancy bits are combined with the previously received sequence for decoding by the Viterbi decoder and the RS decoder. The decoded sequence is checked with the *CRC* bits for error detection. If there is no error in the sequence, then the procedure stops. Otherwise, a retransmission request is sent to the transmitter through the control packet with the corresponding bit off in the *BMAP* (figure 5), and the procedure moves to step $i+1$.

Step i**,** $i \geqslant s$**.** In response to the retransmission request, incremental redundancy bits r_s are transmitted using the rate table with the code rate $R_i = P/(P + r_0 + \cdots + r_{s-1} + (i-s+1)r_s)$, $r_s \geqslant 1$. Because of buffer overflow in the worst case, incremental redundancy bits are not increased any more and W packets are chosen to be discarded from the receive buffer, based on the priority of the packets. At the receiver, the redundancy bits are combined with the previously received sequence for decoding by the Viterbi decoder and the RS decoder. The decoded sequence is checked with the *CRC* bits for error detection. If there is no error in the sequence, then the procedure stops. Otherwise, a retransmission request is sent to the transmitter through the control packet with the corresponding bit off in the *BMAP* (figure 5), and the procedure moves to step $i+1$.

Figure 8 depicts two types of packet formats for the hybrid ARQ scheme based on the SDLP packet formats given in figure 5. Each hybrid ARQ packet starts with the synchronization field (e.g., 28 bits) used to detect the beginning of the packet. The FEC parameter field (e.g., 8 bits) is used to indicate how this packet is encoded in the concatenated FEC scheme at the transmitter. This field consists of three subfields: interleaver depth (e.g., 3 bits), code rate R of the convolutional code (e.g., 3 bits), and error-correcting capability t of the RS code (e.g., 2 bits) in the concatenated FEC scheme. The FEC request field (e.g., 8 bits) has the same structure as the FEC parameter field, and is used to ask the transmitter to encode with the particular FEC parameters from the next data packet until a new FEC request arrives. The FEC parameters in the FEC request field are determined based on the current channel error statistics from the decoder. The FEC redundant bits are generated according to the FEC parameters and appended to the packet.

The synchronization sequence is transmitted without FEC encoding, while the remaining packet header is always encoded at the code rate $1/4$ of the convolutional code with the maximum value $t = 8$ of the RS code in the concatenated FEC scheme to provide a high probability of error-free decoding on noisy channels. Since the FEC parameters for the packet header are fixed unlike the rest of the packet, the receiver can decode the packet header part all the time without any prior knowledge about the FEC encoding of the packet. The $1/4$ code rate can be obtained by repeating the code sequence of the code rate $1/2$. Furthermore, since there is the FEC request field in the control packet, this hybrid ARQ scheme can also be used for an asymmetric channel environment in terms of error statistics.

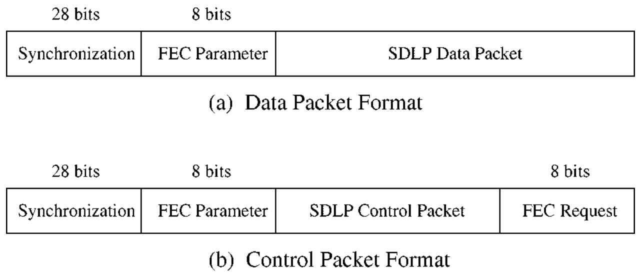

Figure 8. Packet formats for the hybrid ARQ scheme.

5. Performance evaluation

5.1. Throughput analysis

The throughput is defined as the average number of actual information bits per transmitted channel bit. Let E_i be the decoding error event for each packet occurring together at the Viterbi decoder then the RS decoder at step i. The expected value of transmitted channel bits $E[N]$ per correctly accepted l-bit information payload is given by

$$
\begin{aligned}
E[N] = {} & \frac{n}{R_0} + \frac{nr_1}{P}\Pr(E_0) + \cdots + \frac{nr_{s-1}}{P}\Pr(E_0, \ldots, E_{s-2}) \\
& + \left(\frac{nr_s}{P} + nW\right)\big[\Pr(E_0, \ldots, E_{s-1}) \\
& + \Pr(E_0, \ldots, E_s) + \cdots\big],
\end{aligned}
\tag{17}
$$

where R_0 denotes the starting code rate, r_i the incremental redundancy bits at step i, P the period of RCC codes, and W the window size.

$$n = l + h + 16t + m, \tag{18}$$

where l denotes the information payload bits, h the packet control bits, t the error correcting capability in bytes of RS codes, and m the memory size of the convolutional encoder. The throughput η is then given by

$$\eta = \frac{1}{E[N]}\frac{l}{n}. \tag{19}$$

The exact evaluation of $\Pr(E_0, E_1, \ldots)$ is difficult due to the statistical dependences among error events. However, we can obtain a lower and an upper bound using the following inequalities [9]:

$$\prod_{j=0}^{i} \Pr(E_j) \leqslant \Pr(E_0, E_1, \ldots, E_i) \leqslant \Pr(E_i). \qquad (20)$$

5.2. Numerical results

In this section, the throughput of the proposed hybrid ARQ scheme is evaluated for a wireless channel model using an AWGN Rayleigh fading channel with binary FSK (Frequency Shift Keying) modulation. The decoding error event for each packet consists of two individual events, E_{Vit} and E_{RS}, at the Viterbi decoder and the RS decoder, respectively. Since convolutional codes are used as an inner code and RS codes as an outer code in the concatenated FEC scheme, these two error events are sequential and dependent. At each procedure step, the probability of a decoding error $\Pr(E_i)$ for a particular packet corresponds to the word error rate P_{w} at the output of the RS decoder from equation (3) described in section 2. The only difference is that since FSK modulation is used here instead of QPSK for comparison with other typical ARQ schemes, the probability of bit error $P_{\text{b,Vit}}$ at the output of the Viterbi decoder is different.

Let β_d be the coefficients of the derivative of the convolutional code transfer function $T(\cdot)$. Given the free distance d_{free,R_i} and β_{d,R_i} at the code rate R_i, the probability of bit error at the output of the Viterbi decoder with the code rate R_i used at step i is given by [14]

$$P_{\text{b,Vit}}(R_i) < \frac{1}{n} \sum_{d=d_{\text{free},R_i}}^{\infty} \beta_{d,R_i} P_d, \qquad (21)$$

where n denotes the input sequence bits to the convolutional encoder and P_d the probability that a wrong path of weight d is selected. Let the average received SNR (signal-to-noise ratio) per bit be γ_{b}. With soft decision, the probability of P_d is given by [14]

$$P_d = p^d \sum_{k=0}^{d-1} \binom{d-1+k}{k} (1-p)^k, \qquad (22)$$

where $p = 1/(2 + R_0\gamma_{\text{b}})$. From these equations, lower bounds on the throughput of the proposed ARQ hybrid scheme with the concatenated FEC can now be calculated.

In order to evaluate the throughput of the hybrid ARQ scheme for the concatenated FEC, the starting code rate R_0 should be considered first as shown in equation (17). Using equation (4) for the performance results of the concatenated FEC, we can compare with the convolutional code case in terms of the code rate. As shown in figure 9, the concatenated FEC can increase the code rate by taking advantage of the small overhead of RS codes at a given channel condition. In fact, since the performance curve of the concatenated scheme is much steeper than that of the convolutional code, the higher the error performance, the larger the difference of the code rate between two coding schemes. For example, a convolutional code with soft decision and perfect channel state information can achieve the BER performance of 10^{-9} with $E_{\text{b}}/N_0 = 7.6$ dB and code rate $= 1/3$ on an AWGN Rayleigh fading channel [6]. Instead, the concatenated scheme offers the same performance with the overall code rate of about 0.421, which is much better than that of the convolutional code case.

The lower bounds on the throughput of the proposed hybrid scheme with $P = 8$, $s = 1$, optimized over all values of R_0 and R_1 for $n = 500$ and $W = 10$, are presented in figure 10. For comparison, the throughput curves for the hybrid ARQ scheme with convolutional codes [11], type II hybrid ARQ scheme [10], and classical SR-ARQ [20] are also plotted in figure 10. The results show that our proposed scheme outperforms other ARQ schemes for all SNR values. Moreover, the more error performance is required, the higher throughput is expected due to the difference of the starting code rate.

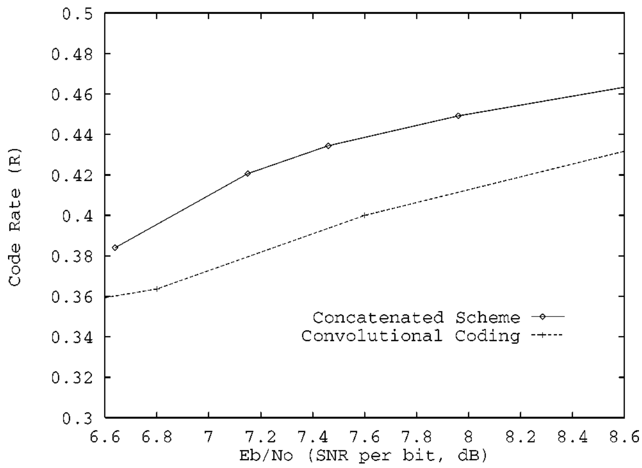

Figure 9. Comparison between concatenated scheme and convolutional coding at the BER 10^{-7} in terms of code rate.

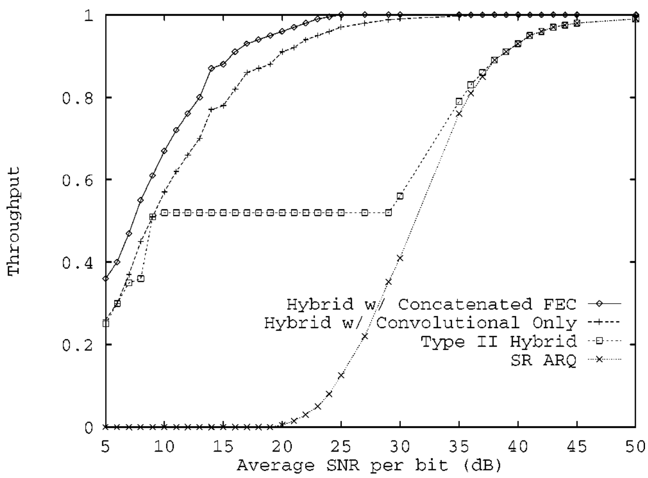

Figure 10. Throughput of hybrid ARQ scheme using concatenated FEC on an AWGN Rayleigh fading channel.

6. Conclusions

In this paper, we have discussed the design and performance of a novel adaptive hybrid ARQ scheme with the concatenated FEC to support quality-critical traffic over wireless ATM networks. The performance of a concatenated FEC scheme has been analyzed over an AWGN Rayleigh fading channel, showing that it can provide sufficient error performance for wireless ATM. The hybrid ARQ scheme is based on our ARQ scheme, SDLP, featuring variable packet size and periodic status message. The key ideas in the proposed hybrid ARQ scheme are to adapt the code rate to the channel conditions using incremental redundancy and to increase the starting code rate as much as possible with the concatenated FEC, maximizing the throughput efficiency. This strategy allows the system to be flexible and adaptive to channel conditions, especially suitable for a time-varying channel with high error rates, such as a wireless channel. The numerical results have shown that our proposed scheme outperforms other ARQ schemes for all SNR values.

Acknowledgement

I would like to give my particular thanks to my advisor, Dr. Ian F. Akyildiz, for his insightful comments and constructive suggestions.

References

[1] I.F. Akyildiz and I. Joe, A new ARQ protocol for wireless ATM networks, in: *Proceedings of IEEE International Conference on Communications ICC '98* (June 1998) pp. 1109–1113.

[2] E. Ayanoglu, S. Paul, T.F. LaPorta, K. Sabnani and R.D. Gitlin, AIRMAIL – A link layer protocol for wireless networks, Wireless Networks 1(1) (February 1995) 47–60.

[3] J.B. Cain and D.N. McGregor, A recommended error control architecture for ATM networks with wireless links, IEEE Journal on Selected Areas in Communications 15(1) (January 1997) 16–28.

[4] Y. Chen and C. Wei, Performance evaluation of convolutional codes with MPSK on Rician fading channels, IEE Proceedings 134(2) (April 1987) 166–173.

[5] European Telecommunications Standards Institute, European digital cellular telecommunications system, GSM 4.22 (May 1992).

[6] J. Hagenauer, Rate-compatible punctured convolutional codes and their applications, IEEE Transactions on Communications 36(4) (April 1988) 389–400.

[7] T.R. Henderson, Design principles and performance analysis of SSCOP: a new adaptation layer protocol, Computer Communication Review 25 (April 1995) 47–59.

[8] I. Joe, An adaptive hybrid ARQ scheme with concatenated FEC codes for wireless ATM, in: *Proceedings of ACM/IEEE MobiCom '97* (September 1997) pp. 131–138.

[9] S. Kallel and D. Haccoun, Generalized type II hybrid ARQ scheme using punctured convolutional coding, IEEE Transactions on Communications 38(11) (November 1990) 1938–1946.

[10] S. Kallel and C. Leung, Efficient ARQ schemes with multiple copy decoding, IEEE Transactions on Communications 40(3) (March 1992) 642–650.

[11] S. Kallel, Efficient hybrid ARQ protocols with adaptive forward error correction, IEEE Transactions on Communications 42(2) (February 1994) 281–289.

[12] S. Nanda, R. Ejzak and B.T. Doshi, A retransmission scheme for circuit-mode data on wireless links, IEEE Journal on Selected Areas in Communications 12(8) (October 1994) 1338–1352.

[13] A.N. Netravali, W.D. Roome and K. Sabnani, Design and implementation of a high-speed transport protocol, IEEE Transactions on Communications 38(11) (November 1990) 2010–2024.

[14] J.G. Proakis, *Digital Communication*, 2nd edition, (McGraw-Hill, 1989).

[15] D. Raychaudhuri and N.D. Wilson, ATM based transport architecture for multiservices wireless personal communication networks, IEEE Journal on Selected Areas in Communications 12(8) (October 1994) 1401–1414.

[16] C.E. Shannon, Communication in the presence of noise, Proceedings IRE 37(1) (January 1949) 10–21.

[17] Y.A. Tesfai and S.G. Wilson, FEC schemes for ATM traffic over wireless links, in: *Proceedings of IEEE International Conference on Communications ICC '96* (June 1996) pp. 948–953.

[18] Y. Yasuda et al., High-rate punctured convolutional codes for soft decision Viterbi decoding, IEEE Transactions on Communications 32(3) (March 1984) 315–319.

[19] R. Yuan, S.K. Biswas and D. Raychaudhuri, A signaling and control architecture for mobility support in wireless ATM networks, in: *Proceedings of IEEE International Conference on Communications ICC '96* (June 1996) pp. 469–477.

[20] E.J. Weldon, An improved selective-repeat ARQ strategy, IEEE Transactions on Communications 30(3) (March 1982) 480–486.

Inwhee Joe received the B.S. and M.S. degrees in electronics engineering from Hanyang University in Seoul, Korea, the M.S. degree in electrical and computer engineering from the University of Arizona, Tucson, in 1994, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, in 1998. From 1985 to 1992, he was an Engineer in the Research Center of DACOM Corporation, where his research focused on networking software, operating systems, and distributed systems. Since 1998, he has been a Member of the Network Research Group at Oak Ridge National Laboratory, Oak Ridge, TN. His current research interests include wireless ATM networks, mobile networking, multimedia networking, and performance evaluation.
E-mail: inwhee@virtue.dsrd.ornl.gov