An Adaptive FEC Scheme for Data Traffic in Wireless ATM Networks

Ian F. Akyildiz, *Fellow, IEEE*, Inwhee Joe, Henry Driver, and Yung-Lung Ho

***Abstract*—In this paper, a new adaptive forward-error-correction scheme (AFEC) is introduced at the link layer for TCP/IP data traffic in wireless ATM networks. The fading and interference in wireless links cause high and variable error rates, as well as bursty errors. The purpose of the AFEC scheme is to provide a dynamic error-control mechanism by using the Reed–Solomon coding to protect the ATM cell payload, as well as the payload type indicator/cell loss priority fields in the ATM cell header. In order to enhance the error tolerance in cell framing and correct delivery, the AFEC scheme functions within a new concept called LANET framing and addressing protection mechanisms. The AFEC scheme has been validated using a simulation testbed of a low-speed wireless ATM network.**

***Index Terms*—Adaptive FEC, address protection, cell scrambling, error control, LANET, Reed–Solomon coding, wireless ATM.**

I. Introduction

The fading effects and interference in wireless links cause higher and time-varying error rates, as well as burstier error patterns compared to wired links. More powerful error-control schemes are needed to insulate the ATM network layer from the wireless channel impairments, because the wireless channel errors will typically be beyond the error-correcting capability of the only error-control scheme, the header error control (HEC), at the ATM layer.

Efficient error control for time-varying wireless channels can be realized by an adaptive coding scheme. One of the well-known adaptive coding schemes is the so-called *convolutional coding*, which can adapt easily to variable channel conditions by modifying its code rate or the constraint length [2], [10]. It has been demonstrated that the convolutional coding itself is not sufficient for wireless ATM networks [7], [16]. Thus, a concatenated FEC scheme has been proposed by combining a convolutional inner code and a Reed–Solomon (RS) outer code with interleaving [7], [16]. However, this scheme may have interleaving delay and accordingly high overhead.

Recently, some adaptive FEC schemes on the transport packets have been introduced to improve reliability for real-time traffic over wide area networks including the Internet [6], [13]. The amount of redundant information is adjusted as a function of the network state, such as packet loss statistics, delay constraints, and quality of service (QoS) requirements. Since the increased redundancy at the end-to-end transport level could cause congestion to the network, their FEC schemes are coupled with congestion control in order to maintain the network stability.

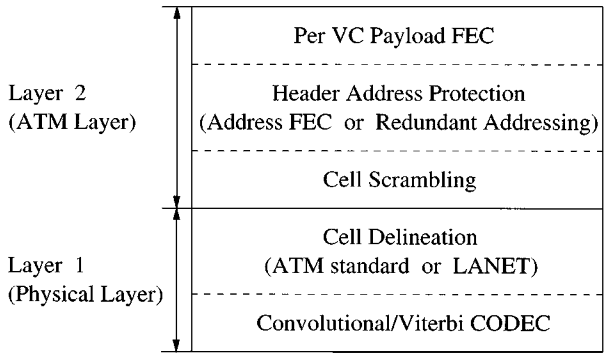


Fig. 1. Structure of the AFEC scheme.

Our new adaptive FEC (AFEC) scheme on the wireless links utilizes a multi-layered structure as shown in Fig. 1. The key requirement is "per virtual circuit FEC." For example, the voice traffic may use a low-performance and short-interleave scheme, since they are error-tolerant and sensitive to delays. On the other hand, the data cells may use high-performance FEC schemes which may have high overhead due to long interleave chains for maximum random and burst-error tolerance. In order to maximize the efficiency, the FEC rate in each channel (virtual channel) can dynamically adapt to the noise level in the wireless channel. In the upper part of the physical layer is the so-called *LANET* [17], which will be explained in the Section V in detail. The *cell scrambling* and *header address protection* are both active in the lower part of the ATM layer. The *payload FEC*, the highest layer in the AFEC scheme, operates in the upper part of the ATM layer.

Our AFEC scheme uses RS codes to protect nibbles in the header and bytes in the payload. Interleaving is also used to enhance the burst-error tolerance without impacting the performance of random error corrections. Our new scheme works well with the convolutional coding at the lowest layer, as shown in Fig. 1. Thus, it is recommended to use radio modems with Viterbi CODECs built in and the coding rates 7/8 or 3/4.

The paper is organized as follows. In Section II, we present a detailed coding procedure of the FEC payload, followed by a description of two implementations for header address protection in Section III. In Sections IV and V, we explain the cell scrambling and LANET framing, respectively. In Section VI, we

Manuscript received September 17, 1998; revised May 28, 1999; approved by IEEE/ACM Transactions on Networking Editor M. El Zarki.

I. F. Akyildiz is with Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Tech, Atlanta, GA 30332 USA (e-mail: ian@ee.gatech.edu).

I. Joe is with Telcordia Technologies, Morristown, NJ 07960-6438 USA,

H. Driver is with Lucent Technologies, Landover, MD 20785 USA (e-mail: hdriver@lucent.com).

Y. L. Ho is with Linsang Partners, Silver Spring, MD 20910 USA.

Publisher Item Identifier S 1063-6692(01)06847-9.

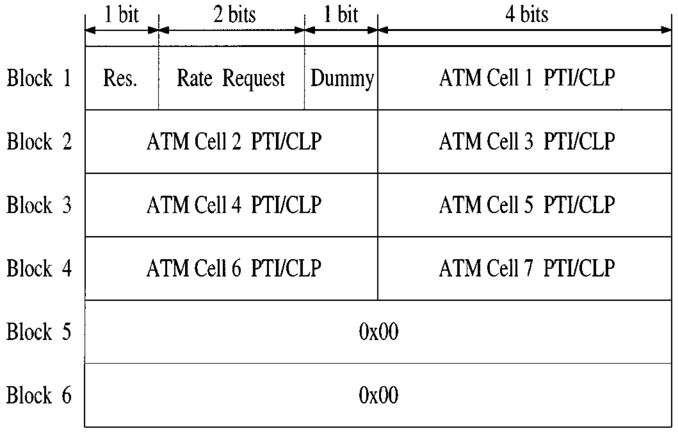


Fig. 2. The header format for six payload blocks.

evaluate the performance of our AFEC scheme by simulation. Finally, we conclude the paper by highlighting our contribution in Section VII.

II. FEC Payload for Data Traffic

The scheme described here is applicable for data traffic. Voice traffic protection, emphasizing compression with FEC and interoperability with the current voice circuit emulation standard, is not yet developed. The FEC payload scheme uses the RS FEC scheme to protect the payload of the ATM cell, as well as the payload type indicator/cell loss priority (PTI/CLP) fields of the ATM cell header in noisy wireless channels.

A. Cell-Encoding Procedure

1) Accumulate ATM cells to form a group considering the current coding rate of the corresponding virtual circuit (VC).
   - For 1/2 coding rate, put one cell in a group.
   - For 3/4 coding rate, put three cells in a group.
   - For 7/8 coding rate, put seven cells in a group.

   Time out if the group is not complete and no cell arrives within 1/4 seconds. If timed-out, pad with *dummy cells* to complete the group. This time-out period works for links at 2400 Bd or faster. Below 2400 Bd, the 1/2 coding rate should be used. Note that the time-out period may depend on the link speed in the future.

   Dummy Cell Format: The last byte of the last cell in the group of dummy cells indicates the number of dummy cells in that group, and the rest of the payloads is set to a specific hexadecimal value $0x6A$.
2) Extract payload and segment each group into six blocks. The block sizes depend on the coding rates as follows:
   - For 1/2 coding rate, there are 8 B/block.
   - For 3/4 coding rate, there are 24 B/block.
   - For 7/8 coding rate, there are 56 B/block.
3) Allocate space for one piggyback byte as the header for each block. The header format for the six payload blocks is shown in Fig. 2. PTI/CLP nibbles for cells 2–3 are applicable at the 3/4 encoding rate. PTI/CLP nibbles for cells 4–7 are applicable at the 7/8 encoding rate.

   The "rate request" field indicates the desired encoding rate to the other side, based on the errors detected during decoding. Note that it does not reflect the encoding rate used for this side's output. The rate indications are 0 $\rightarrow$ No Change, 1 $\rightarrow$ 7/8 rate, 2 $\rightarrow$ 3/4 rate, and 3 $\rightarrow$ 1/2 rate. The dummy cell bit, if set to 1, indicates that dummy cells are present in the group.
4) Apply RS encoding on each block above where their size increases according to the encoding rates as follows.
   - For 1/2 rate, encode the block of 9 $(8+1)$ to 15 B.
   - For 3/4 rate, encode the block of 25 $(24+1)$ to 31 B.
   - For 7/8 rate, encode the block of 57 $(56+1)$ to 63 B.

   Then use RS($n = 255$, $k = 249$) code with the 3-B correcting capability. However, k must be shortened to $k = 9$, 25, or 57 depending on the currently used code rate. The code is shortened by implicit zero filling. The zeros are assumed to occupy the front positions in the message stream. The output of the encoder is in reverse order, i.e., the parity bytes are output first followed by the last message byte. Finally, the piggyback bytes are the last to output.
5) Prepare the RS-encoded cell group for output. Use the same header for all cells within a group.
   - For 1/2 coding rate, generate two cells.
   - For 3/4 coding rate, generate four cells.
   - For 7/8 coding rate generate eight cells.
6) Load the delineation byte in the first byte location of each output cell within a group. The period of the delineation byte pattern is used by the receiving side to determine the code rate.
   - For 1/2 rate, use $0x5A$, $0xA5$.
   - For 3/4 rate, use $0x5A$, $0xA5$, $0x0F$, $0x0F$.
   - For 7/8 rate, use $0x5A$, $0xA5$, $0x0F$, $0x0F$, $0x0F$, $0x0F$, $0x0F$, $0x0F$.
7) Load payload.
   - Optionally, the PTI and CLP bits can be overwritten with user-defined messages.
   - Load the remaining payload part of each cell in the group. Interleave data from the six blocks: block 1 byte 1, block 2 byte 1, block 3 byte 1, block 4 byte 1, block 5 byte 1, block 6 byte 1, block 1 byte 2, etc. Since the RS decoder returns data in reverse order, the loaded first bytes are the parity bytes. On the other hand, the last-loaded bytes are the piggyback bytes. For the 7/8 rate case, there is no room for the last two piggyback bytes (blocks 5 and 6). They were defined as $0x00$ and must be discarded.
   - Disregard unused bytes (4 B at 1/2 rate, 2 B at 3/4 rate, none at 7/8 rate).
8) Swap the least-significant nibble between delineation byte and byte 25 in the payload (the delineation byte is byte 1).
9) Output the cell group starting with cell 1 (the one with delineation byte $0x5A$).

B. Cell-Decoding Procedure

1) Cell group delineation.
   - For each cell, swap the least-significant nibble between the delineation byte and byte 25 in the payload (the delineation byte is byte 1).
   - Wait for the next cell's arrival.

- Overwrite bits 7, 5, 3, and 1 of the delineation byte with corresponding bits from the next cell's delineation byte. Thus, if the first delineation byte is $0x5A$ and the next one is $0xA5$, the first delineation byte will become $0xF0$. Assuming the link is error free, the delineation byte sequence becomes:

— 1/2 rate: $0xF0$, $0x0F$, $0xF0$, $0x0F$, $0xF0$, $0x0F$, … (repeats every two cells).

— 3/4 rate: $0xF0$, $0x0F$, $0x0F$, $0x0F$, $0xF0$, $0x0F$, … (repeats every four cells).

— 7/8 rate: $0xF0$, $0x0F$, $0x0F$, $0x0F$, $0x0F$, $0x0F$, $0x0F$, $0x0F$ (eight-cell period).

This method of interleaving doubles the burst error tolerance of the delineation bytes.

- Using table lookup (256-B table), find "start of a group," "which is signaled by the delineation byte as $0xF0$ and all other bytes equal or less than 3 b different from $0xF0$. There are eight possible bytes that are 1-b different, 28 B that are 2-b off, and 56 B that differ by 3 b. In other words, a total of 93 entries in the lookup table should indicate a match for the "start of a group" byte.
- If the cell arrives at the fixed-rate port of the switch, the completion of a group is indicated as follows: If the delineation byte indicates $0xF0$, terminate the current group and save the new cell as the first cell of the next group; if the current group size is two cells based on 1/2 rate, four cells based on 3/4 rate, and eight cells based on 7/8 rate, then terminate the current group with the new cell (the next group has no cells). Otherwise, save the new cell as part of the current group.

  Terminate the group if no cell arrives within 1 s. Discard the group whose length does not match the configuration.
- If the cell arrives at the dynamic rate port of the switch, the completion of a group is indicated by finding a delineation byte indicating $0xF0$ or if no cell arrives within 1/4 s. Discard the group if the length is not two, four, or eight cells. The group length is used to set the decoding rate. Note that the time-out period may depend on the link speed in the future.

2) Extract payload.
   - Extract the six data blocks by de-interleaving. For the 7/8 rate, add a byte of $0x00$ at the end of blocks 5 and 6 to complete the six data blocks.
   - Optionally, pull out the inserted messages from the piggyback byte locations and re-insert the PTI and CLP bits.
3) Decode by using the RS scheme.
4) Rate adjustment for dynamic links.

   For rate adjustment in dynamic wireless links, we introduce the so-called *Bucket* concept based on channel-error statistics.

$$[Bucket = Bucket + deterrs - k * Bucket/scalefac]$$

where

k	information symbol size depending on the code rate with $k = 9$ for 1/2 rate, $k = 25$ for 3/4 rate, and $k = 57$ for 7/8 rate;
deterrs	number of detected errors as an output parameter of the RS decoder;
scalefac	scaling factor.

Bucket accumulates the number of detected errors *deterrs* and uses the scaled information symbol size k to reduce it, especially when the link quality becomes better.

When the scaling factor *scalefac* is set to 84 000, the relationship between the *Bucket* size and the channel BER is identified at a given code rate from the real field tests.

- Currently at 1/2 rate: If $Bucket < 100$, then the channel BER becomes 10^{-4}. Then change the coding rate to 3/4 rate.
- Currently at 3/4 rate: If $Bucket > 200$, then the channel BER becomes 10^{-3}. Then change the coding rate to 1/2 rate. If $Bucket < 10$, then the channel BER becomes 10^{-5}. Then change the coding rate to 7/8 rate.
- Currently at 7/8 rate: If $Bucket > 20$, then the channel BER becomes 10^{-4}. Then change the coding rate to 3/4 rate.

5) Regenerate original cells.
   - Fill payload.
   - Pull PTI/CLP nibbles from the piggyback bytes.
   - If the "dummy cell present bit" is set, then read the last byte of the last cell in the group to determine the number of dummy cells to be discarded.
   - Read rate request bits to determine the encoding rate for traffic going to the opposite direction.

III. Header Address Protection

In this section we discuss two implementations for header address protection: address FEC and redundant addressing.

A. Address FEC Encoding

Each port shall support four types of address FEC encoding. The 1-nibble correcting RS(15,13), and 2-nibble correcting RS(15,11) codes are used. Each code is shortened to the necessary length by implicit zero filling. The four types of address FEC encoding schemes are described as follows.

1) Hi-Noise UNI [Fig. 3(a)].

   This method encodes 2 nibbles of VCI to 6, by adding 4 nibbles of parity. There will be 2-nibble correcting and 255 usable VCs.
2) Hi-Noise NNI [Fig. 3(b)].

   This method encodes 3 nibbles of VPI and VCI to 7. The GFC field is used to carry 4 b of VPI.
3) Lo-Noise UNI [Fig. 3(c)].

   This method encodes 4 nibbles of VPI and VCI to 6, adding 2 nibbles of parity. It is 1-nibble correcting and leaves 64K usable VCs.
4) Lo-Noise NNI [Fig. 3(d)].

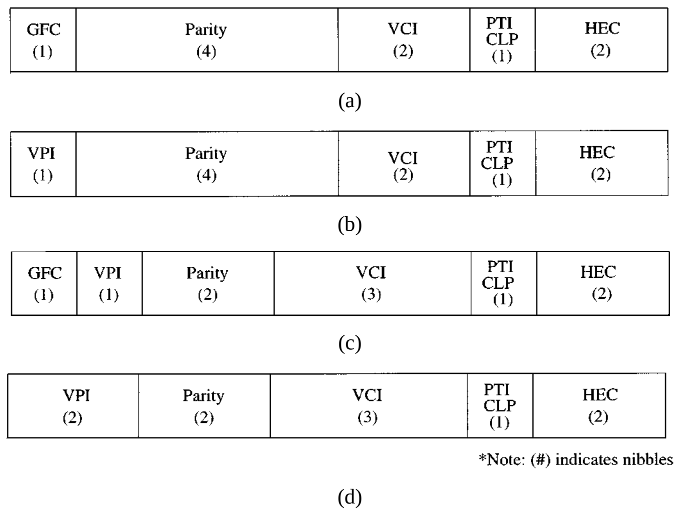


Fig. 3. Address FEC types. (a) Hi-Noise UNI. (b) Hi-Noise NNI. (c) Lo-Noise UNI. (d) Lo-Noise NNI.

This method encodes 5 nibbles of VPI and VCI to 7. It leaves 1 million usable VCs.

B. Low-Cost Redundant Addressing Option

By coding only the address part of the ATM cell header, the AFEC scheme provides a low cost implementation for the header protection scheme. This method does not decode and correct errors in the address; instead, it tolerates errors. This is accomplished by setting up multiple addresses to the same destination during address translation in the switch. One of the addresses is the principal one, corresponding to the original uncorrupted address. The others are variations from the principal address; the differences being the errors in the address the switch can tolerate. For this to work, all principal addresses should be coded address words (parity + VC) from Section III-A above. This ensures adequate Hamming distance between principal address words.

The major advantage for this low-cost *redundant addressing* option is that no special hardware is needed for implementation. Most switches already have hardware to perform address translation. Thus, all that is required is for the controlling software to load the proper addresses in the input lookup table. The disadvantage is that the number of VCs supported is reduced significantly. To tolerate two random bit errors or 4-b burst errors for a Hi-Noise UNI link requires 385 redundant addresses for each principal one. Thus, the actual number of VCs supported per link becomes the size of the input address translation table divided by 385, making this option attractive primarily for the smaller end-user sites.

IV. Cell Scrambling

The cell scrambling is used to strengthen the burst-error tolerance of the header address. It consists of three byte swaps followed by three nibble swaps. Note that the bytes 0–4 are the ATM cell-header bytes.

1) Byte swap 1: bytes 0 and 20.
2) Byte swap 2: bytes 1 and 36.
3) Byte swap 3: bytes 2 and 52.

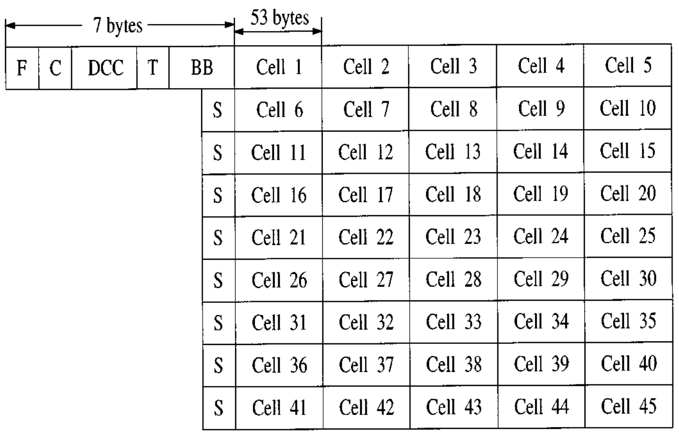


Fig. 4. LANET frame structure.

4) Nibble swap 1: least significant nibble (LSN) between bytes 12 and 36.
5) Nibble swap 2: LSN between bytes 20 and 44.
6) Nibble swap 3: LSN between bytes 28 and 52.

For the Lo-Noise UNI and NNI header-encoding cases [Fig. 3(c) and (d)], the cell scrambling extends the block-error tolerance of the addresses to 57 b. For the Hi-Noise UNI and NNI schemes [Fig. 3(a) and (b)], the block-error tolerance of the address increases to 121 b. To remain compatible with traditional cell delineation methods, the ATM header HEC hould be computed after scrambling.

V. LANET Framing

While the AFEC scheme can function with standard HEC delineation specified by the ATM Forum [3], LANET delineation offers significant performance gain in noisy wireless environments. Indeed, LANET cell delineation functions at BERs as high as 10^{-2}. However, the standard delineation is still required for interoperability when FEC is not needed. The AFEC, thus, specifies that both delineation methods be provided and be selectable.

LANET's design emphasis is on noise tolerance and the possibility of low cost firmware implementation. Yet, its structure is similar to SONET. The ATM cells are packed in a framing structure with frame headers and sub-frame headers, as shown in Fig. 4. By inserting predictable header patterns, the extra information helps to delineate the cells in a noisy environment. A LANET frame has the following features.

- It consists of a byte-oriented serial data stream. The frame size is 2400 B.
- The 2400 B are subdivided into 45 ATM cells (2385 B) with a 15-B overhead. The LANET overhead is, therefore, about 0.63% of the bandwidth.
- The 2400-B LANET frame is subdivided into nine subframes.

As for the overhead bytes, a LANET frame starts with a 1-B frame header of $0x96$ (F). The C field is a checksum byte computed over the previous frame except F. The 2-B data communication channel (DCC) is used to provide for the operation, administration, and maintenance of the communication link. After this field, there is 1 B reserved for the transport-layer control channel (T) whose usage is yet to be defined. In addition, there are two types of special header patterns: $0xF628$ (BB) for byte stuffing control and $0xE8$ (S) for the subframe header.

The advantages of this scheme will become more apparent after examining the recommended implementation discussed below.

A. Recommended Implementation

The recommended implementation is firmware based. This low-cost approach is possible because the LANET headers can be simply detected, and thus, it can provide initial byte alignment. The single-byte headers are also less likely to be corrupted by noise, and thus, additional confidence checks can be achieved for synchronization verification.

Three basic units, the *cell extractor*, the *framer*, and the *history buffer*, constitute the LANET engine. The *cell extractor* extracts cells based on the initial alignment information supplied by the *framer*. It also declares loss of synch (*LOS*) if large numbers of frame headers and ATM headers are corrupted. Current design calls for LOS declaration upon encountering two consecutive frame/sub-frame header errors plus five consecutive ATM HEC errors. At this setting, the mean time between noise-induced LOS is increased by two orders of magnitude over standard HEC-based delineation schemes without loss of ability to detect real synchronization loss. Finally, the *cell extractor* also locates frames (identify $0x96$) if the *framer* only found a sub-frame alignment.

The *framer* locates frame or sub-frame alignment in the event of LOS. Note that it makes use of the *history buffer*, which greatly reduces the synchronization time. Typical synchronization time is about five cells for cold starts (i.e., *history buffer* is empty), and about 2.5 cells for warm starts (i.e., *history buffer* is full).

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of AFEC for TCP/IP data traffic in wireless ATM networks with stationary and mobile wireless terminals by using simulations. We compare the simulation results between AFEC and the existing concatenated FEC scheme [7], [16] and demonstrate the improvements achieved by AFEC.

In order to evaluate the AFEC scheme with and without LANET framing structure in wireless ATM networks, we developed two simulation models, as shown in Fig. 5.

- *Configuration 1*: AFEC scheme without LANET framing in wireless ATM networks.
- *Configuration 2*: AFEC scheme with LANET framing in wireless ATM networks.

The wireless channel model for the physical layer is shown in Fig. 6. For modulation we use π/4-shifted differential quadrature phase shift keying (DQPSK). This modulation scheme is particularly useful for fast fading environments due to differential encoding. Moreover, differential detectors are simpler to implement than the coherent detectors.

The wireless channel suffers from various impairments, such as multipath fading, co-channel interference, and delay spread. The most important of these is the multipath fading due to

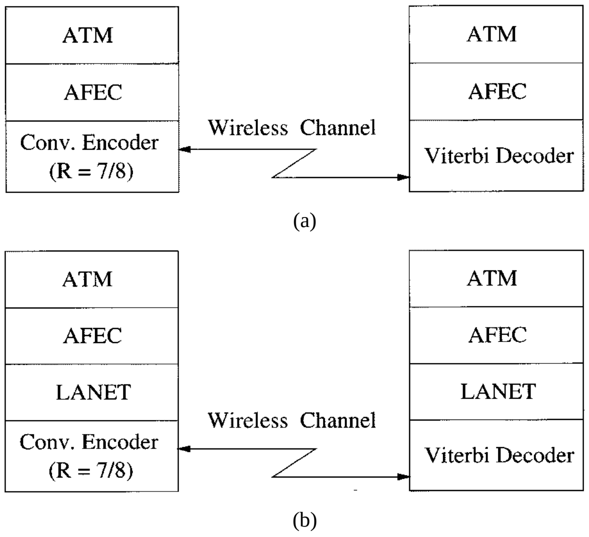


Fig. 5. Simulation models. (a) Configuration 1. (b) Configuration 2.

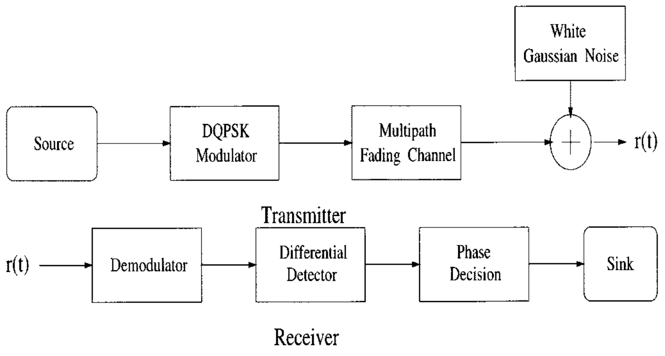


Fig. 6. Physical wireless channel model.

the reception of multiple paths which is caused by reflections or scattering of the transmitted signal from various objects. Each path can be modeled with random amplitudes and phases. Furthermore, the motion of wireless workstations introduces a Doppler shift for each path. The combined effect of these events is that the magnitude of the received signal becomes Rayleigh distributed. Such channels are closely approximated by Jake's model [11], [].

For simulations, we assumed the carrier frequency as a 2.4-GHz ISM band, the data rate as 128 kb/s, i.e., low-speed wireless links, and wireless terminals as stationary or mobile moving at 5 and 55 mph, respectively. Each wireless terminal has a wireless link with an ATM switch, which is used as a base station. At the 2.4-GHz band and normal mobile speeds, the Doppler shift is limited up to 200 Hz. For example, the Doppler shift is about 18 Hz for pedestrians at 5 mph and 197 Hz for the mobile terminal speed of 55 mph. Since two source bits are mapped into one channel symbol at a time in π/4-DQPSK, the symbol interval is about 15.6 μs at 128 kb/s.

From simulations, we obtained the following performance measures as a function of the channel BER for stationary and mobile wireless terminals by modifying the SNR values and Doppler shift.

- *The Cell Sync Error Rate (%)*: The ratio of the number of ATM cells in sync error based on the cell delineation

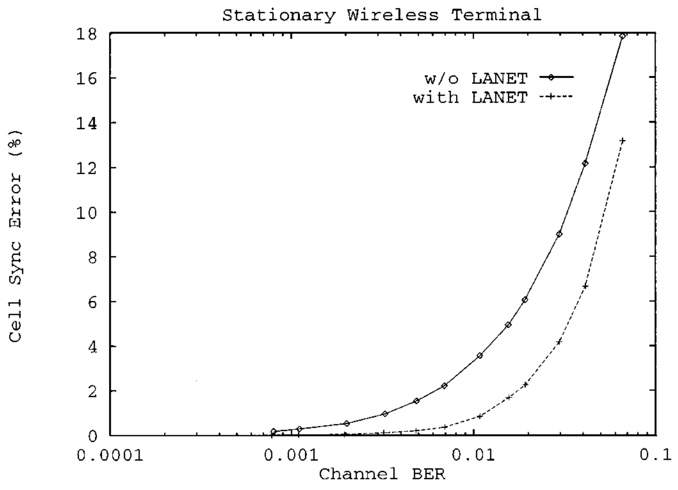


Fig. 7. Sync performance of LANET for stationary wireless terminal.

byte in the ATM payload to the total number of ATM cells transmitted.

- *The BER Performance after FEC Correction*: The final BER values after FEC correction with LANET framing.

Fig. 7 presents the performance of LANET in terms of cell sync error rate as a function of channel BER for the stationary wireless terminal. The simulation results show that LANET increases the sync performance significantly, especially when the channel BER becomes higher. For mobile wireless terminals, we also notice the substantial performance improvement with LANET as shown in Fig. 8(a) and (b) for the mobile speeds of 5 and 55 mph, respectively.

In Fig. 9, we show the performance of AFEC with LANET framing for a stationary wireless terminal. The final BER values after FEC correction are presented as a function of channel BER values. In particular, the final BER after FEC correction is about 10^{-6}, even in the case of high-channel BER of 10^{-2}, which is acceptable for TCP/IP traffic. In Fig. 10, we show the performance of AFEC for a mobile wireless terminal at 5 and 55 mph, respectively.

As the mobile speed increases, the error rates become higher and the error patterns tend to become burstier [4]. Since the concatenated FEC and AFEC schemes both use the RS coding with its inherent burst-error correcting capability, they exhibit better performance in case of high mobility under the same channel BER conditions. The final BER after FEC correction is about 10^{-6} at the channel BER of 10^{-2} for both mobile speeds, as it was in the stationary case before.

Since the AWGN portion is dominant in case of the stationary wireless terminal, the final BER values after FEC correction agree with the results from the benchmark experiments where random bit errors are injected to generate error situations [17]. Even for mobile wireless terminal cases, the performance of AFEC does not degrade at all, because the interleaving scheme in payload FEC (Section II) works effectively to take care of the burst errors due to the Doppler shift. That is, the final BER after FEC correction is still about 10^{-6} at the channel BER of 10^{-2}, regardless of the mobile speed.

The existing concatenated FEC scheme [7], [16] is also implemented to compare with our AFEC scheme. We use a common convolutional code with the code rate of 1/2 and constraint length of 7 as the inner code in this concatenated FEC scheme [7], [16]. We use an RS code with the symbol size of 8 and the error-correcting capability of 8 as the outer code. The block size of the RS code is chosen as one ATM cell. As shown in Fig. 9, the AFEC scheme provides a slightly better performance over the entire range than the existing

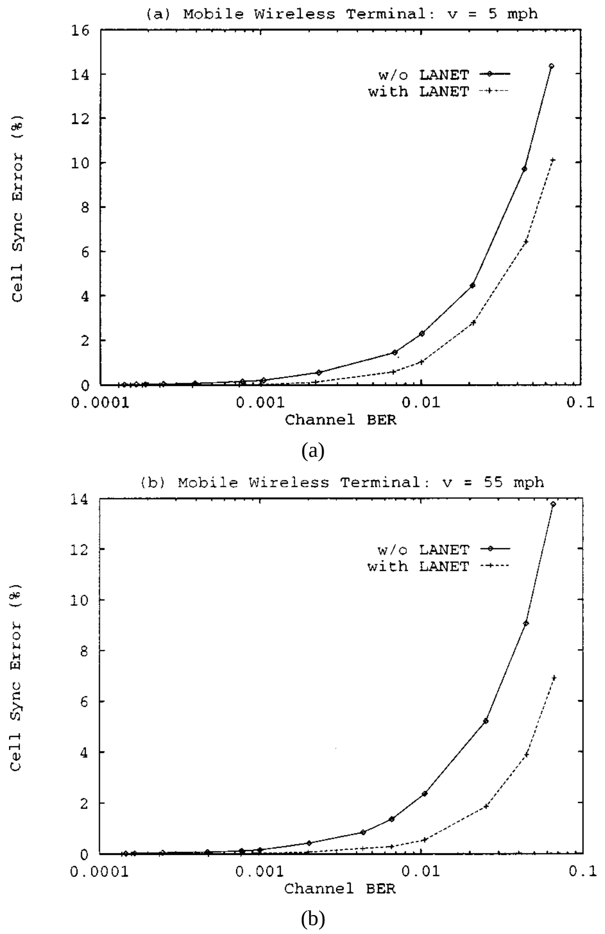


Fig. 8. Synch performance of LANET for mobile wireless terminal. (a) Mobile wireless terminal: v = 5 mph. (b) Mobile wireless terminal: v = 55 mph.

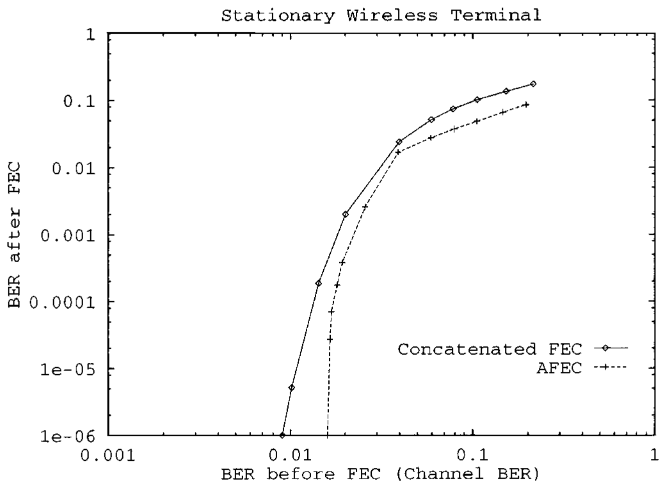
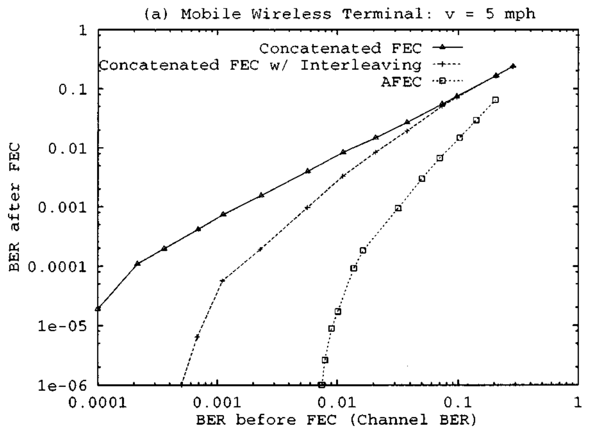
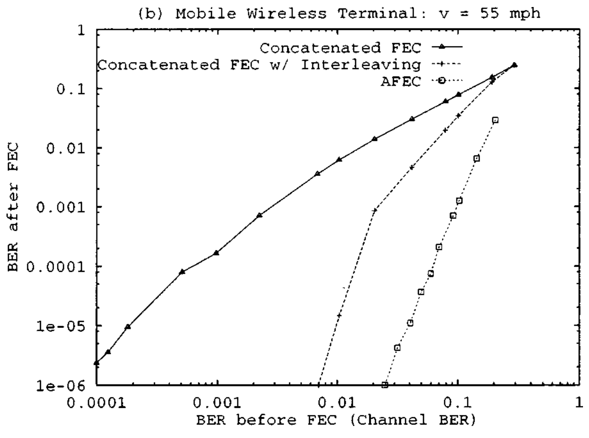


Fig. 9. FEC performance for stationary wireless terminal.



(a)



(b)

Fig. 10. FEC performance for mobile wireless terminal. (a) Mobile wireless terminal: v = 5 mph. (b) Mobile wireless terminal: v = 55 mph.

concatenated FEC scheme in case of the stationary wireless terminal [7], [16].

Furthermore, as shown in Fig. 10, the AFEC scheme provides a significant improvement in terms of error correcting performance compared to the existing concatenated FEC scheme [7], [16] for mobile wireless terminals because the AFEC scheme captures burst errors very effectively.

On the other hand, the AFEC scheme is designed to change its RS code rate adaptively according to the channel error statistics. It provides three code rates of RS coding (1/2, 3/4, and 7/8). We achieved the best performance using the code rate of 1/2.

We have conducted many experiments to determine the code rate threshold between RS code rates 1/2, 3/4, and 7/8 for the AFEC scheme. In Fig. 11 we show the AFEC performance for different code rates. The code rate threshold is determined from the maximum tolerable BER based on different applications. When the maximum tolerable BER is given for a particular application, then the code rate threshold can be obtained from Fig. 11. For example, if the maximum tolerable BER is 10^{-6} for TCP/IP traffic, then the code rate threshold is as follows.

- Channel BER 2×10^{-4} between code rates 7/8 and 3/4.
- Channel BER 2×10^{-3} between code rates 3/4 and 1/2.

For the channel-BER range below the point of 2×10^{-4}, the code rate of 7/8 can be used for TCP/IP traffic requiring a maximum tolerable BER of 10^{-6}. However, the performance

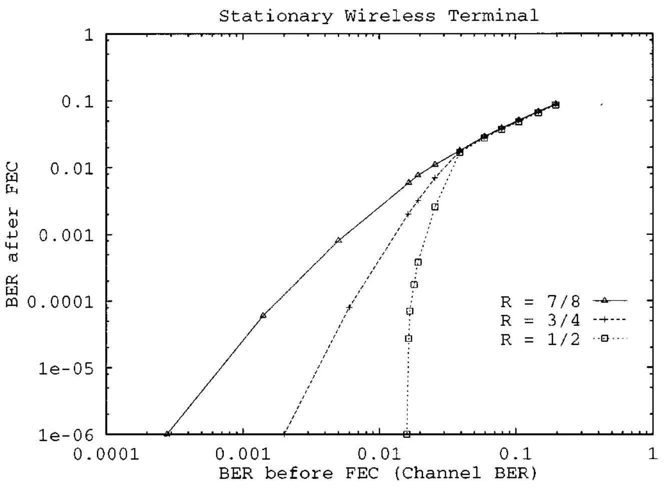


Fig. 11. FEC performance for different code rates in wireless channel.

of AFEC with the code rate 1/2 is not sufficient for the channel-BER range beyond the point of 2×10^{-2}. From the simulation results we determined the code rate thresholds as 1/2 for 10^{-3} BER and as 3/4 for 10^{-4} BER, and as 7/8 for 10^{-5} BER for the rate adjustment in dynamic wireless links to support TCP/IP traffic.

VII. Conclusion

In this paper, we presented the design and performance of a novel AFEC scheme, for TCP/IP data traffic in wireless ATM networks. The AFEC scheme is based on RS coding to protect the ATM cell payload and PTI/CLP fields in the ATM header. In order to enhance the error tolerance in terms of framing and correct delivery, the AFEC scheme functions within our existing LANET framing and addresses protection protocols. We have also presented the performance evaluation results using simulation models. For stationary wireless terminals the performance of AFEC with LANET framing agreed with the benchmark experiments in which random bit errors are injected [17]. At channel BER of 10^{-2}, the final BER after FEC correction is about 10^{-6}, which is acceptable for TCP/IP traffic. Even for mobile wireless terminal cases, the performance of AFEC does not degrade at all, because the interleaving scheme in payload FEC (Section II) works effectively to take care of the burst errors due to the Doppler shift. Finally, we compared the performance of our AFEC scheme with the existing concatenated FEC scheme [7], [16] we demonstrated that our scheme provides significant improvement in terms of error performance.

References

[1] I. F. Akyildiz *et al.*, “A new adaptive FEC scheme for wireless ATM networks,” in *Proc. IEEE MILCOM’98*, Oct. 1998, pp. 277–281.

[2] J. F. Arrigo *et al.*, “Adaptive FEC on a reconfigurable processor for wireless multimedia communications,” in *Proc. IEEE Int. Symp. Circuits and Systems*, 1998, pp. 417–420.

[3] ATM Forum Technical Committee, “ATM UNI Specification,” ATM Forum Report, 1993.

[4] E. Ayanoglu *et al.*, “AIRMAIL: A link-layer protocol for wireless networks,” *ACM-Baltzer Wireless Networks (WINET)*, pp. 47–60, Feb. 1995.

[5] E. Ayanoglu, P. Pancha, A. R. Reibman, and S. Talwar, "Forward error control for MPEG-2 video transport in a wireless ATM LAN," *ACM-Baltzer Mobile Networks and Applications (MONET)*, pp. 245–257, 1996.

[6] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for Internet telephony," in *Proc. IEEE INFOCOM'99*, Mar. 1999, pp. 1453–1460.

[7] J. B. Cain and D. N. McGregor, "A recommended error control architecture for ATM networks with wireless links," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 16–28, Jan. 1997.

[8] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ," in *Proc. ICC'99*, June 1999, pp. 1212–1216.

[9] P. R. Denz and A. A. Nilsson, "Performance of error control coding techniques for wireless ATM," in *Proc. ICC'98*, June 1998, pp. 1099–1103.

[10] P. L. Hiew, M. Zukerman, and M. Gitlits, "WATM operation optimization based on effect of FEC code rate and ARQ retransmission," in *Proc. IEEE Vehicular Technology Conf. VTC'98*, 1998, pp. 2542–2546.

[11] W. C. Jakes, *Microwave Mobile Communications*: Wiley, 1974.

[12] Y. Nakayama and S. Aikawa, "Cell discard and TDMA synchronization using FEC in wireless ATM systems," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 29–34, Jan. 1997.

[13] K. Park and W. Wang, "AFEC: An adaptive forward error correction protocol for end-to-end transport of real-time traffic," in *Proc. IC3N*, 1998, pp. 196–205.

[14] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network Mag.*, pp. 40–48, Sept./Oct. 1998.

[15] D. Raychaudhuri and N. D. Wilson, "ATM-based transport architecture for multiservices wireless personal communications networks," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1401–1414, Oct. 1994.

[16] Y. A. Tesfai and S. G. Wilson, "FEC schemes for ATM traffic over wireless links," in *Proc. IEEE Int. Conf. Commun. ICC'96*, Aug. 1996, pp. 948–953.

[17] Lucent Technologies, "LANET with adaptive payload FEC," Lucent Technologies, Landover, MD, Tech. Rep., 1997.

[18] M. Zorzi and R. R. Rao, "On the impact of burst errors on wireless ATM," *IEEE Personal Commun. Mag.*, pp. 65–76, Aug. 1999.

Ian F. Akyildiz (M'86–SM'89–F'96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen–Nuernberg, Germany, in 1978, 1981, and 1984, respectively.

Currently, he is Director of the Broadband and Wireless Networking Laboratory and Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His current research interests include wireless networks, satellite networks, and the Internet. He is the Editor-in-Chief of *Computer Networks* and Editor of the *Journal for Multimedia Systems*, *Journal of ACM Wireless Networks*, and *Journal of Cluster Computing*

Dr. Akyildiz is a past Editor IEEE/ACM TRANSACTIONS ON NETWORKING (1996–2001) and a past Editor for IEEE TRANSACTIONS ON COMPUTERS (1992–1996). He was the Program Chair of the "9th IEEE Computer Communications" Workshop (1994), the ACM/IEEE MOBICOM'96 Conference, and the IEEE INFOCOM'98. He will be the General Chair for the ACM/IEEE MOBICOM 2002 Conference and the Technical Program Chair of IEEE ECC'2003. He served as a National Lecturer for ACM from 1989 to 1998 and received the ACM Outstanding Distinguished Lecturer Award in 1994. He also received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria, Chile, and the 1997 IEEE Leonard G. Abraham Prize Award for his paper "Multimedia Group Synchronization Protocols for Integrated Services Architecture," published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in January 1996. He is a Fellow of the Association of Computing Machinery (ACM).

Inwhee Joe received the B.S. and M.S. degrees in electronics engineering from Hanyang University, Seoul, Korea, the M.S. degree from the University of Arizona at Tucson in 1994, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, GA, in 1998, both in electrical and computer engineering.

From 1998 to 2000, he was a member of the Network Research Group, Oak Ridge National Laboratory, Oakridge, TN, where his research focused on MAC and routing protocols for mobile *ad-hoc* networks. He is currently a research scientist in the Wireless Network Management Research Lab, Telcordia Technologies, Morristown, NJ. His research interests include 3G wireless networks, wireless ATM networks, mobile networking, multimedia networking, and performance evaluation.

Henry H. Driver received the B.S. degree in electrical engineering from the University of La Habana, Cuba, in 1964, the M.S. degree in electrical engineering in 1973, and the Ph.D. degree in computer science in 1977, both from the University of Delaware.

Currently, he is a Senior Network Consultant with Lucent Technologies, Landover, MD. His professional interests include ATM networks, MPLS, G-MPLS, IP over DWDM, and automatic switched optical networks.

Yung-Lung Ho received the B.E.S. and M.S.E. degrees in electrical engineering from the Johns Hopkins University, Baltimore, MD, in 1981 and 1982, respectively, and the Ph.D. degree in physics from the University of Wisconsin at Madison in 1988.

From 1988 to 1995, he was a Research Scientist at Science Applications International Corporation, SAIC, where he studied MHD plasma behavior, RF heating of fusion plasmas, and 3-D atmospheric models. In 1995, he joined Yurie Systems Inc., Landover, MD, as Director of Advanced Technology, involved in developing encryption, queuing, and wireless-ATM technologies. He is currently a member of the Linsang Partners, LLC, Investment Group, Silver Spring, MD, developing the infrastructure to support universal access of customized information.