Wireless Netw (2009) 15:241–258
DOI 10.1007/s11276-007-0038-5

A hybrid error recovery scheme for scalable video transmission over 3G cellular broadcast networks

Kyungtae Kang · Yongwoo Cho · Heonshik Shin

Published online: 14 April 2007

Abstract The cdma2000 1xEV-DO mobile communication system provides broadcast and multicast services (BCMCS) to meet an increasing demand for multimedia data services. But the servicing of video streams over a BCMCS network faces a challenge from the unreliable and error-prone nature of the radio channel. BCMCS uses Reed-Solomon coding integrated with the MAC protocol for error recovery. We show that this is not effective for mobiles moving at the edge of service area, where the channel condition is bad, resulting in significantly lower video quality. To improve the playback quality of an MPEG-4 FGS (fine granularity scalability) video stream, we propose a hybrid error recovery scheme incorporating a packet scheduler, which uses slots saved by reducing the Reed-Solomon coding overhead. Packets to be retransmitted are prioritized by a utility function which reduces the packet error-rate in the application layer within a fixed retransmission budget by considering of the map of the error control block at each mobile node. Our error recovery scheme also uses the characteristics of MPEG-4 FGS to improve the video quality even for a slow-moving mobile which is experiencing a high error-rate in the physical channel because of error bursts.

Keywords cdma2000 1xEV-DO · Reed-Solomon · Hybrid error recovery · BCMCS · MPEG-4 FGS

K. Kang (✉) · Y. Cho · H. Shin
Dept. of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-744, Korea
e-mail: ktkang@cslab.snu.ac.kr

Y. Cho
e-mail: xtg05@cslab.snu.ac.kr

H. Shin
e-mail: shinhs@cslab.snu.ac.kr

1 Introduction

Work has recently begun, in both the Third Generation Partnership Project (3GPP) and the 3GPP2, on enhancing 3G networks to support multimedia broadcast and multicast services (BCMCS) [1, 2]. The 3GPP2 group has already baselined the specification for a cdma2000 high-rate broadcast packet-data air interface [3, 4]. The goal is to design a system that can deliver multimedia broadcast and multicast traffic with minimum resource usage by both the radio access and core networks.

A mobile user of a wireless channel can experience great variations in multipath fading, path loss from distance attenuation, shadowing by obstacles, and interference from other users. Thus, a wireless radio channel has a much higher error-rate than a wired link. The unreliable and error-prone nature of the radio channel is the major challenge in serving video streams over cdma2000 broadcast networks. In BCMCS, the MAC protocol uses Reed-Solomon (RS) coding as the method of forward error correction. In this paper, we analyze the performance of RS error recovery under different channel conditions, while varying the RS parameters. Extensive simulation results demonstrate that the resulting analytical model can accurately predict the packet error-rate in the application, and subsequent analysis shows that the performance of RS degrades significantly for mobile nodes moving slowly in bad channel conditions, such as those that exist at the edge of the service area, where the length of error bursts increases to the extent that RS cannot recover sufficient data. This reduces the proportion of the coverage area in which a high-speed broadcast service is available.

In order to serve the full coverage area with a high data-rate broadcast service, efficient error control is required. For this purpose, we adopt a hybrid error recovery scheme that incorporates a packet scheduler to maintain the quality of

MPEG-4 FGS video streams [5–7] in poor conditions by improving the performance of existing RS error recovery. Instead of trying to achieve this by adding bulky parity information, we use an RS code with a low parity overhead, to save time-slot resource. This reserved bandwidth can then be used to retransmit corrupted packets, especially those packets which are most important in decoding MPEG-4 FGS video streams. The number of retransmissions is necessarily limited, and packets are selected for participation by considering the arrangement of MAC packets in the RS error control block (ECB), with the aim of improving error recovery capacity, and hence the resulting video playback quality. This is done in a proportionally fair way, so as to increase throughput, using a utility function. By means of a realistic simulation, we demonstrate the effectiveness of our proposed hybrid error recovery scheme in improving both error recovery capacity and video playback quality.

Our approach should enable broadcast streaming and multimedia communication services of higher quality and reliability, and allow BCMCS to provide video to all mobile subscribers with fewer interruptions.

2 Background

2.1 The cdma2000 1xEV-DO broadcast service

The cdma2000 1xEV-DO wireless standard supports the delivery of broadcast and multicast services (BCMCS), which transmit multimedia content from a single source to multiple users simultaneously, complementing unicast services which send video content to subscribers individually (e.g., video-on-demand). Figure 1 is an architectural overview and framework description of BCMCS over a cdma2000 network.

BCMCS content originates from the content provider and goes through the content server, which may be located within the cdma2000 serving network or on an IP network such as the Internet, which is connected to the cdma2000 access network through a packet data serving node (PDSN) that handles the BCMCS content stream. The PDSN communicates with the packet control function of the base station controller (BSC/PCF) to add or remove multicast IP flows. The BSC/PCF is responsible for signaling, establishing and terminating bearer channels between the PDSN and the mobile nodes. A detailed description of broadcast service procedure can be found elsewhere [1, 2, 8, 9].

2.2 Scalability of MPEG-4 FGS video streams

A major topic in traditional multimedia research has been the pursuit of efficient ways to compress data at a given bit-rate. Recently, several scalable video coding schemes have been developed for controlling stream bandwidth flexibly to meet the growing need for streaming video over packet-based networks, such as the Internet, where the network channel capacity varies widely with the type of connection, the level of network traffic and a lot of other external factors. To cope with this growing requirement for flexible stream transport, an MPEG working group has introduced a scalable video coding scheme called fine granularity scalability (FGS).

The FGS video coding scheme in MPEG-4 not only provides an effective method of video compression, but also adapts its bit-rate flexibly to changing network conditions. The FGS scheme encodes video frames into two layers, distinguished by the priority of information that they contain, using bit-plane coding of DCT (discrete cosine transform) coefficients [10]. The layer containing the information that is essential for the decodability of the whole video stream is called the base layer. It requires a modest bit-rate and produces a lower bound on picture quality. The second layer is called the enhancement layer and includes more detailed information for enhancing the quality of the decoded video. The enhancement layer is obtained by bit-plane DCT coefficient coding of the differences between the original picture and the picture degraded by image transformations such as DCT and quantization. Because bit-plane coding considers each quantized DCT coefficient as a binary rather than

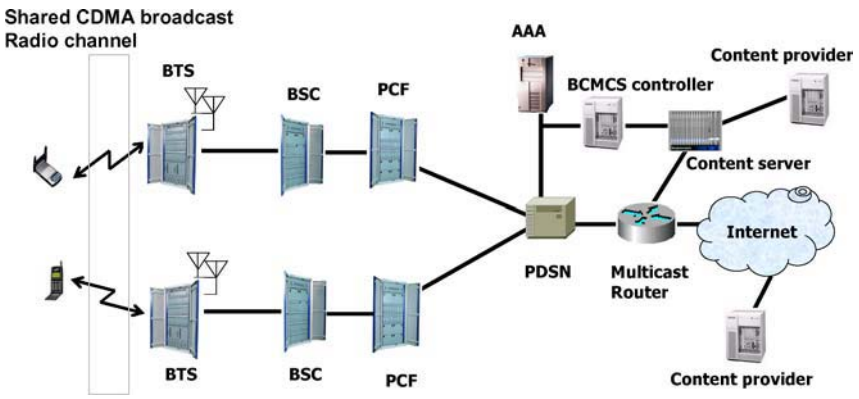


Fig. 1 The cdma2000 BCMCS network architecture

a decimal number, each bit-plane of the enhancement data stream has its own level of significance, from the MSB (most significant bit) to the LSB (least significant bit), and the enhancement layer can be truncated to any reduced number of bits. After receiving the whole of the base layer, the quality of the video increases progressively as more of the enhancement layer is received. Thus FGS provides a very efficient method of scalable video transmission, since the overhead of layer separation accounts for at most 2 dB in the peak signal-to-noise ratio [11].

However, the base layer of an FGS stream is very sensitive to channel errors. If the decoder finds any errors in the base layer, the enhancement layer of the current frame is discarded, whether it is correct or not. If they go undetected, errors in the base layer will propagate to the start of the next group of pictures (GOP) and cause serious drifting problems in the enhancement layers of the following frames. While accuracy in the base layer is essential for decoding video streams, the enhancement layer is more tolerant, and errors in the enhancement layer cannot degrade the video quality below the lower bound provided by the base layer.

To utilize this layered approach effectively, the more important base layer should be provided with more protection against errors than the less important enhancement layer. However the error recovery scheme in the current BCMCS standard makes no allowance for the different value of the two layers, and each is equally likely to be damaged during transmission.

2.3 MAC-layer error recovery in current BCMCS

In contrast to the unicast cdma2000 1xEV-DO standard, where a subscriber's forward-link data rate depends on its RF conditions, BCMCS require service providers to use a common speed to send video to all subscribers in the area covered by their cells. BCMCS can nevertheless deliver a consistent high-quality video stream over a large area by using an error correction scheme based on Reed-Solomon coding [3, 4], which is applied to the layers above the existing turbo code.

Figure 2 shows the protocols that make up the broadcast protocol suite, and Fig. 3 shows the structure of the MAC-layer error recovery mechanism. The broadcast framing protocol segments higher-layer packets at the access network; the broadcast security protocol provides encryption of framing packets; and the broadcast MAC protocol defines the procedures used to transmit over the broadcast channel, and also specifies an outer code which, in conjunction with the physical-layer turbo code, forms the product code. As already mentioned, RS has been chosen as the outer code for cdma2000 BCMCS, and the broadcast MAC-layer packets have a fixed size of 125 octets. The protocol is completed by the broadcast physical layer and an ECB, which is a buffer

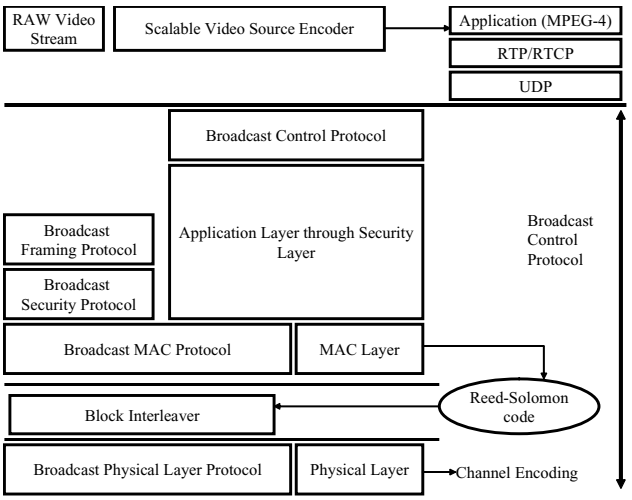


Fig. 2 The BCMCS protocol stack

required to perform RS decoding, and which is transported as payload on one or more subchannels of this layer. Data from multiple ECBs is multiplexed on to the broadcast physical channel, as shown in Fig. 3.

Each logical channel uses ECBs encoded with the same RS parameters (N, K, R), and has M MAC packets per ECB row. The variables N and K represent the total number of octets and the number of security-layer octets in a RS code word, while R is the number of parity octets: an RS decoder can correct up to R corrupted octets if their positions are known (in which case RS is acting as an erasure code), or detect and correct up to $R/2$ octets if the positions of the errors are unknown (when RS is an error-correcting code). According to the specification for the cdma2000 high-rate broadcast packet-data air interface, RS is used as an erasure code and not as an error-correcting code, because the cyclic redundancy check (CRC) provided by the physical layer is used to detect and erase damaged physical-layer packets.

RS coding is applied to the columns of the ECB, and then the data is transferred row by row to the physical slot, where it forms one or more physical-layer packets. The error control block is designed to provide a structure such that, in the event of a physical-layer packet erasure, octets in the same position are lost from all affected codewords. To decode an RS codeword correctly, the broadcast MAC protocol needs to receive at least K of the N octets in that codeword; but if all K data octets are received without errors, decoding is not needed. The data octets which are successfully received are forwarded to the upper layer of the BCMCS protocol suite.

One of the most significant environmental factors affecting channel condition is the fading effect. This is correlated with the burstiness of errors. In bad, slow-moving conditions, the error occurrence pattern tends to be more bursty than in fast-moving conditions. An RS code (N, K, R) cannot recover any lost data if the corrupted portion is larger

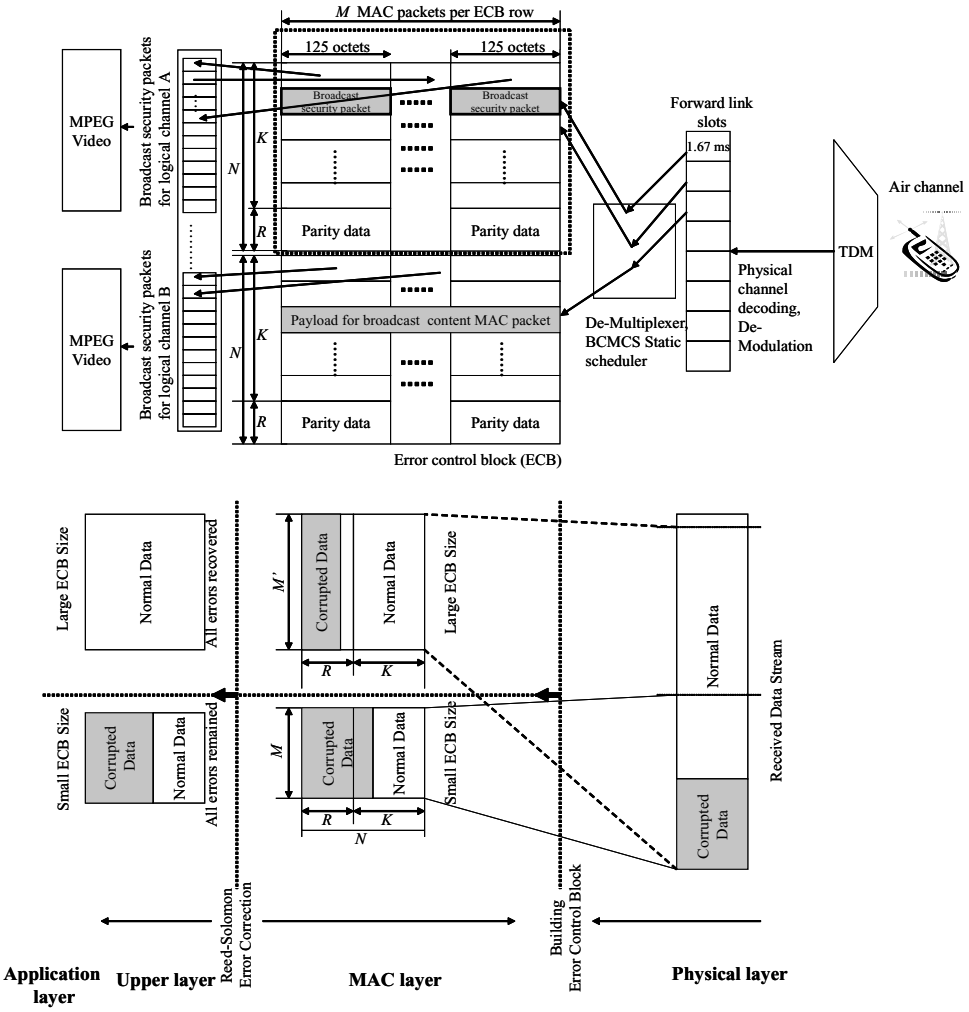


Fig. 3 Error recovery structure for current BCMCS

than R. For this reason, the performance of error correction will drop if the bursts of errors becomes so long that the ECB cannot interleave the errors sufficiently. This situation is shown schematically in Fig. 3. Thus, the burstiness of errors caused by bad channel conditions is an important factor in selecting an appropriate data interleaving interval, determined by the width of the ECB, which is $M \times 125$ octets. The value of M for a given ECB has to be less than or equal to 16. As the value of M increases, the time diversity also increases and thus a mobile node which is in a time-varying shadow environment is able to recover a substantial amount of corrupted data. However, this requires additional storage at the node. Therefore, choosing an appropriate value of M is important in achieving the best error recovery capacity for the available system resource.

3 Related work

A hybrid ARQ scheme uses an error control code in conjunction with a retransmission scheme. It first tries to decode each codeword that it receives, and only requests a retransmission if the uncertainty of the decoding decision is considered too high, i.e. if the reliability of detection is below a certain threshold. A retransmission can be requested by sending a negative acknowledgment signal or implied by the absence of an acknowledgement signal. Properly designed hybrid ARQ is more reliable than an error control code alone and has a higher throughput than a pure ARQ scheme [12].

There are two main types of hybrid ARQ scheme, which are usually called Type-I [13] and Type-II [14, 15]. In a Type-I scheme, a fixed code-rate is used and, when a data packet is declared to be corrupted, the receiver rejects it and asks for retransmission. This scheme does not adapt to changing channel conditions and is best suited to channels with a fairly constant level of noise or interference. In a Type-II scheme, on the other hand, an information packet is first transmitted with enough parity bits for error detection, but too few for error correction. Additional parity bits are transmitted in response to a retransmission request, and the receiver combines these with the original packet to form a more powerful error-correcting code which is usually able to recover all

the information. Type-II hybrid ARQ is usually more efficient than Type-I because it makes use of corrupted packets instead of discarding them. However, in non-stationary time-varying fading channels, errors tend to come in bursts, and a packet might be severely corrupted or completely lost during a long deep fade. Such a packet can play no part in transferring information, no matter whether it originally contained data or parity bits. One solution to this problem is selective combination [16], in which additional parity bits are provided by retransmission and the receiver attempts to recover the information by combining packets selectively, avoiding any severely damaged ones. This hybrid scheme combines the advantages of both Type-I and Type-II ARQ schemes and significantly improves system performance over bursty fading channels.

A hybrid ARQ error control scheme can also be based on the concatenation of an RS code and a rate-compatible punctured convolutional (RCPC) code [17], for applications such as low bit-rate video transmission over wireless channels. This scheme also combines the advantages of Type-I and Type-II hybrid ARQ schemes. A certain degree of error correction capability is incorporated in each (re)transmitted packet, and the information in a corrupted packet can be recovered from that transmission or retransmission alone if the extent of the errors is within the capability (in this respect this scheme is similar to Type-I hybrid ARQ). But a retransmitted packet also contains redundant bits which, when combined with the previous packet, yields a more powerful RS or convolutional concatenated code which often permits recovery of the information after the error correction applicable to individual transmissions has failed (this functionality is similar to Type-II hybrid ARQ).

As an additional measure, adaptive error control protocols can make transmission over time-varying wireless channels more efficient. The throughput achievable with a hybrid ARQ protocol can be maximized by dynamic adaptation of the channel coding parameters. In many adaptive ARQ-based protocols [18–20], the code-rate is selected in response to the actual channel state, which can be estimated by counting the number of erroneous frames; but an accurate estimation of channel state requires the analysis of many frames, so this is slow. A somewhat different adaptive hybrid ARQ model, due to Yang and Hossain [21], is based on the assumption that the channel state (in terms of symbol error rate) can be estimated on a frame-by-frame basis from the signal-to-noise ratio (SNR) fed back by the receiver. This is intended to lead to faster code-rate adaptation when the channel condition changes. Yang and Hossain [21] model and analyze an adaptive error control protocol at the radio link level, using RS codes for wireless networks. They show that their dynamic rate adaptation strategy provides much better throughput than conventional Type-I or Type-II hybrid ARQ protocols.

The use of RS codes with perfect interleaving in hybrid-ARQ schemes has been analyzed for fading channels [22], and a method of analyzing the use of non-interleaved RS codes in Rayleigh fading channels has also been proposed [23]. By considering the equivalence between a block-interleaved system and a non-interleaved system, this analysis can be extended [24] from non-interleaved RS codes to the more general case of the imperfectly interleaved RS codes used in hybrid ARQ.

Most of the existing work relating to hybrid ARQ has focused on increasing the reliability of transmission and on increasing throughput in the physical layer within a unicast environment, for which it is natural to use a feedback channel. Our hybrid scheme is aimed at a broadcast network environment, in particular 3G cellular communications, which provides a feedback channel for ACK/NACK that can be used by our scheme. We have realized our scheme as an expansion of the hybrid ARQ protocol used in the MAC layer of 3G cellular broadcast services. Because this is a broadcast environment, adapting the code-rate to the channel state is not possible, which differentiates our approach from existing unicast solutions.

Our scheme incorporates a scheduler that determines the priorities of corrupted packets using a utility function which will be described in detail later. Our scheme is also content-aware, which means that it considers the classification of the content of an MPEG-4 FGS video stream: specifically, whether a packet corresponds to the base layer or not. Our scheme also considers the topologies of corrupted packets within an ECB sub-block at the mobiles. This allows retransmission of error packets to be requested more selectively, and is therefore exploits Type-II hybrid ARQ more efficiently.

4 Hybrid error recovery in the MAC layer

4.1 Slack slot stealing

The components of the broadcast MAC protocol are shown in Fig. 4. As mentioned above, error control is provided using forward error correction, and the size of the RS code

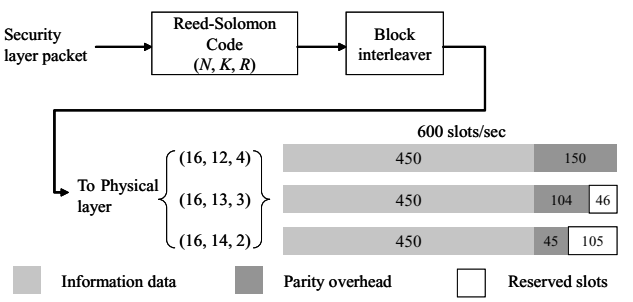


Fig. 4 Broadcast MAC protocol components and RS overhead

block is specified by values of the tuple (N, K, R). In current BCMCS, three RS codes may be used (16,12,4), (16,13,3) and (16,14,2). The (16,12,4) code contains 16 code symbols for each block of 12 information symbols input to the encoder. The first 12 code symbols contain information and the remaining 4 contain parity data. Similarly, the (16,13,3) and (16,14,2) codes contain 16 symbols for each block of 13 or 14 information symbols input to the encoder, while parity data is accommodated in 3 and 2 symbols respectively. When a small value of K is used, the RS error recovery scheme provides improved error correction, but at the cost of a lower effective data-rate for the broadcast service. Conversely, a large value of K reduces the coding overhead at the expense of the error correction capability, thus preventing a high-speed broadcast service from operating in bad channel conditions.

Figure 4 shows the amount of data corresponding to information and parity during 1 second of transmission. There are 600 slots per second and the duration of each slot is therefore 1.67 ms. In order to transmit 450 slots worth of information, the three RS codes require 150 slots, 104 slots and 45 slots for parity data, respectively. This leaves 46 slots unallocated by the (16,13,3) code, and 105 unallocated by the (16,14,2) code, as shown in Fig. 4. These saved slots can be used for retransmission of corrupted broadcast packets. Thus, instead of increasing the RS parity overhead to improve error recovery, we propose to reduce it and to compensate by employing the ARQ error recovery scheme which is already used in cdma2000 1xEV-DO unicast services [25, 26]. Multimedia broadcast and multicast services (MBMS) in GSM/WCDMA also use RLC/MAC with ARQ to enhance performance [27].

4.2 Description of the proposed error recovery scheme

The ECB map shows how each mobile node selects target packets for retransmission, so as to increase error recovery capacity and video playback quality while minimizing the number of packets to be retransmitted. Figure 5 depicts the ECB map for two mobile nodes, node A and node B, which have the same ratio of base-layer data to enhancement-layer data. In the middle of each of these example ECBs is an error cluster. When a mobile node enters an area with a bad channel condition, packets are sequentially corrupted for a certain period. In the example in Fig. 5, the RS code is (16,14,2) and the value of M is 16; packets successfully received are marked G, while corrupted packets are labeled B. Each packet corresponds either to the base layer or to the enhancement layer. Assuming that a physical-layer packet consists of two MAC-layer logical security packets (QPSK modulation with a 1228.8 kb/s data-rate, as given in the cdma2000 1xEV-DO specification) and that the bit-rates of the base and enhancement layers are the same, then base-layer packets and enhancement-layer packets will be arranged in an

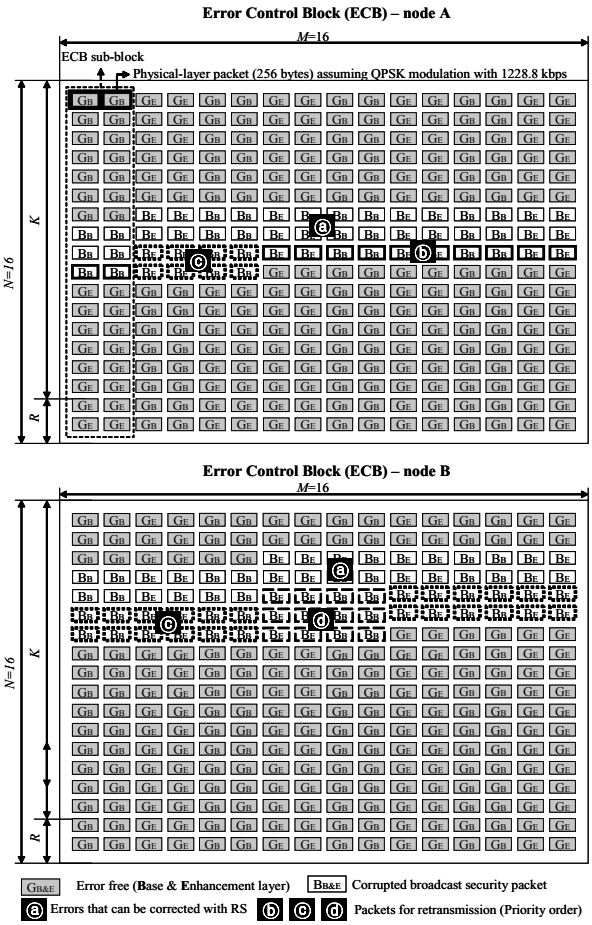


Fig. 5 ECB map of node A and node B

ECB as shown in Fig. 5. Consequently, each ECB sub-block contains either 16 pairs of base-layer security packets or 16 pairs of enhancement-layer security packets. Each pair forms one physical-layer packet.

During error recovery, all corrupted packets can be recovered if the errors are restricted to the region marked (a) in Fig. 5. However, if the error burst is longer than this, a (16,14,2) code can no longer conceal all the errors. The packets that cannot be corrected are scheduled for retransmission using the ARQ scheme. The class to which a packet corresponds and the number of other corrupted packets in the same sub-block determine its priority: corrupted packets that correspond to sub-blocks containing base-layer packets are also given a higher priority than those containing enhancement-layer packets, because the safe receipt of base-layer packets prevents abrupt playback quality degradation, as mentioned in Section 2.2; and when this requirement has been met, packets are assigned a higher priority if relatively few retransmissions are needed to complete a sub-block. The overall effect of this process is to give priority to mobile nodes which are experiencing better channel conditions, in a proportionally fair way, and thus improve throughput.

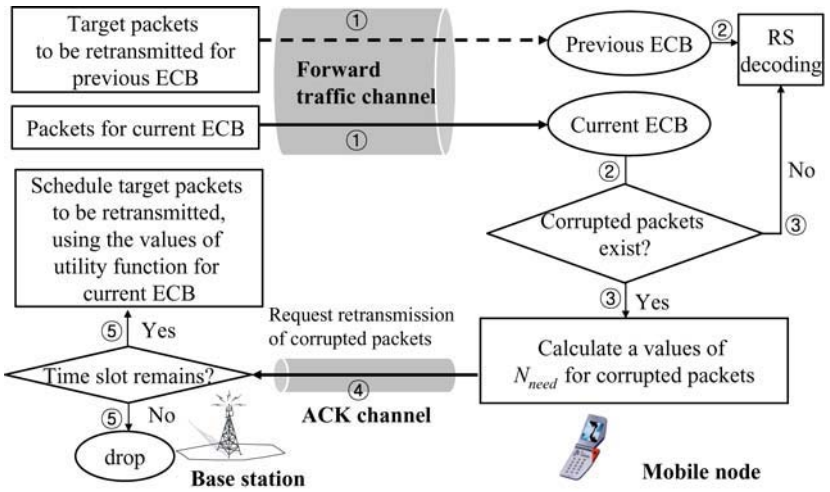


Fig. 6 Overall error recovery structure

We will now consider the ECB at node A in Fig. 5. The sequence of corrupted sub-blocks marked (b) can be recovered if just one packet (bold rectangles) in each sub-block is successfully retransmitted using the slots saved by a (16, 14, 2) code. And all the corrupted sub-blocks marked (c) can be recovered if more than two packets (dotted-bold rectangles) are successfully retransmitted by ARQ. Similarly, all the packets represented by dotted-bold rectangles (d) in node B are targets for retransmission using ARQ. In this example, the slots saved by a (16,14,2) code are sufficient to retransmit all the corrupted packets. But if there is a shortage of slots saved for retransmission, the packets belonging to the region marked (b) will be transmitted first, because they have a higher priority than those in regions (c) and (d). Similarly, the packets in (c) have a higher priority than those in (d).

If slots reserved for ARQ remain after recovering all corrupted base-layer packets, the recovery of enhancement-layer packets is attempted in the same way. Thus a small number of packet retransmissions, utilizing the limited number of reserved slots, greatly improves the video quality. Even in the case of a (16,12,4) code, the packet error-rate in the upper layer can be significantly reduced by reserving slots for retransmission using ARQ, despite very poor channel conditons.

Figure 6 shows a schematic diagram of the proposed hybrid error recovery scheme. The selection of target packets for retransmission is performed as explained in the previous section. In order to prioritize the retransmission of packets with a numerical value, the scheduler uses a function which reflects the utility of each retransmitted packet (τ_i) requested by nodes $\{\mu_0, \mu_1 \ldots \mu_{\eta-1}\}$, defined as follows:

$$f_{utility}(\tau_i) = f_{contents}(\tau_i) + \omega \left[\frac{\eta}{N_{node}}\right] + (1-\omega)\left[1 - \frac{1}{\eta}\sum_{\iota=0}^{\iota=\eta-1} N_{need}(\tau_i, \mu_\iota)\right], \quad (1)$$

where $f_{contents}(\tau_i)$ has the following meaning:

$$f_{contents}(\tau_i) = \begin{cases} 1 & \text{if the corrupted packet } \tau_i \text{ corresponds to the base layer;} \\ 0 & \text{if the corrupted packet } \tau_i \text{ corresponds to the enhancement layer.} \end{cases}$$

N_{node} is the total number of mobile nodes, and $N_{need}(\tau_i, \mu_\iota)$ is the number of corrupted packets that need to be retransmitted in the ECB sub-block to which τ_i corresponds, in order to recover τ_i successfully in mobile node μ_ι. All the corrupted packets in a single ECB sub-block have the same value of N_{need}. The term $[\frac{\eta}{N_{node}}]$ ensures that packets which are requested by more mobile nodes have an improved chance of retransmission, and its effect can be controlled by the weight (ω).

4.2.1 Implementation

There is no priority management in the physical-layer modulator. The priority of each packet is only determined in the MAC layer, using the information received from mobiles. We are assuming that each base-layer and enhancement-layer stream is delivered separately by the content server to the base station, as shown in Fig. 7, which also shows how the BCMCS video scheduler in the base station constructs an ECB and assigns the time-slots to deliver it. Security-layer octets are entered in the ECB row by row, and RS coding is applied along the columns. Each row of the ECB then forms the payload for one or more physical-layer packets.

We propose using the reserved area of the physical-layer packet header in our scheme. Each physical-layer packet is 256 bytes in length when using QPSK modulation with a data-rate of 1228.8 kb/s, according to the forward-link variable-rate parameters in the cdma2000 specification. Each packet can contain a payload consisting of two broadcast content MAC packets, which occupy 250 bytes, and the

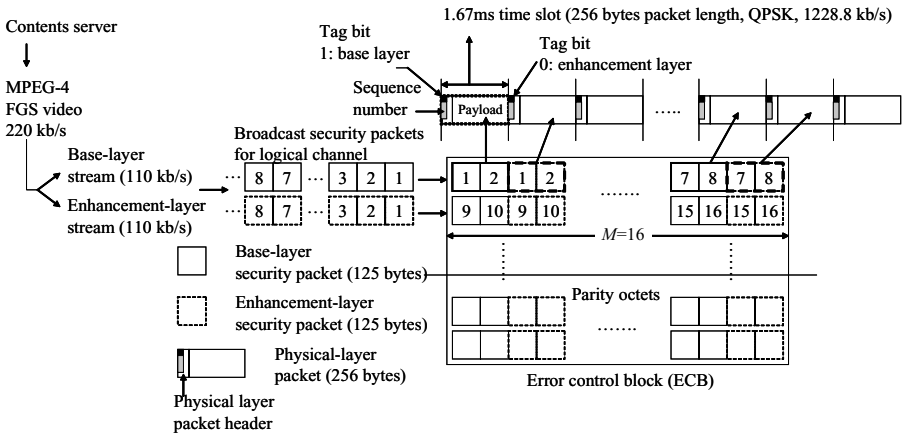


Fig. 7 Packet structure for hybrid error recovery

remaining bytes make up the header. However, only a single tag bit in the header is actually required, to indicate whether the physical-layer packet corresponds to the base layer or enhancement layer of the video (see Fig. 7), although there are also a number of reserved bits that comprise the sequence number required to identify each physical packet. The base station can also determine whether each packet corresponds to the base layer or to the enhancement layer using this sequence number when it calculates the utility function.

Each mobile node calculates the value of N_{need} for all corrupted packets, and reports this information via the reverse ACK channel with the corresponding sequence numbers. This arrangement is already used in unicast services [25, 26], and we propose extending its use to a broadcast environment. These values of N_{need} can be piggybacked on to the negative acknowledgement packets. The base station uses the information that it receives from the mobiles to calculate a value of the utility function for each packet. Theses utilities are then used as priorities to schedule retransmission of the packets: the higher the value, the higher their priority. The scheduled packets are now retransmitted using the forward traffic channel of BCMCS.

The characteristics of our hybrid error recovery scheme may be summarized as follows:

- It is application-aware, which means that it reflects the importance of decoding to the video stream during error recovery.
- The chance of retransmitting a packet is inversely proportional to the number of corrupted packets in the sub-block.
- The throughput is increased because the priority reflects the channel condition of each mobile node during retransmission.

4.2.2 Analysis of delay and jitter

The transport stream system target decoder model of an MPEG system provides a guideline for managing timing constraints and buffers in the MPEG transport stream decoding process. In an MPEG system, the delay should not exceed 100 ms, from arrival to system decoder, and this requirement can be satisfied by using a buffer of adequate size if the end-to-end delay can be bounded. Thus a major challenge in achieving the necessary quality of service over a network is the implementation of a bounded delay service: that is, a communication service with deterministically bounded delays for all packets.

Bounds on the end-to-end delay and jitter between the content server and base station are guaranteed by the jitter-EDD discipline due to Verma et al. [28], who built a model based on packet-switching networks and used it to determine the end-to-end delay bound and the delay jitter. Their channel establishment procedure requires three steps to obtain the necessary information about the bandwidth, processing power and buffer size, and to confirm the delay bound.

We also have to consider the delay and jitter between the base station and a mobile. The delays in feeding back the information required for our error recovery scheme can be controlled by using buffers of adequate size, provided that their performance has a constant-time bound. The scheduler and the round trip itself are other sources of delay. The worst-case round-trip time can be estimated by measuring the time from a base station in the center of the cell to a mobile which is roaming at its edge. And the scheduling delay can be bounded by allowing the base station to restrict the number of NACKs that are issued, depending on the channel conditions of all mobiles in the coverage area. For example, the base station can use the BCMCS overhead messages to specify the maximum number of NACK requests per mobile or to restrict the number of requests to retransmit enhancement-layer packets.

The complexity of our scheme at the base station and at the mobiles can be analyzed separately. Because all corrupted packets within a single ECB sub-block have the same value of N_{need}, counting the number of corrupted packets in each ECB sub-block is the only task that needs to be performed

at each mobile. There are M sub-blocks in each ECB, and so the complexity per ECB is O(M). Somewhat more work has to be done at the base station, which calculates the utility of each corrupted packet using Eq. (1), with a complexity of O(η). These modest complexities suggest that our scheme will run quickly, provided that there is enough buffering both at the base station and the mobiles. We will therefore simplify the rest of our analysis by assuming that there are no significant delays in feeding back the information required to compute the utility function, by means of ARQs, nor in calculation of the utility function itself.

A complexity analysis of our algorithm at the base-station and mobile shows the possibility of a bound on the delay during scheduling, while the end-to-end delay and jitter requirement between the content server and the mobile can be guaranteed by using adequate buffering.

5 Performance analysis of RS error recovery

5.1 Channel model

5.1.1 Rayleigh fading channel

Fading in the air channel is assumed to have a Rayleigh distribution. A first-order two-state Markov process can be used to simulate the error sequences encountered during transmission on a correlated Rayleigh fading channel, and we can model the fate of each data packet using the simple threshold model suggested by Zorzi et al. [29, 30]. These authors considered the details of the specific coding/modulation scheme and tracked the fading process symbol by symbol. Thus they were able to show that a Markov approximation is a good model of the packet error process over a broad range of parameters. Their observations suggest that the marginal error-rate and the transition probability largely depends on an appropriately normalized version of the Doppler frequency. An adequate approximation of this relationship can be simply computed using the threshold model. This simplifies the performance analysis of upper-layer protocols.

We can model different degrees of correlation in the fading process by choosing different values for $f_d N_{BL} T$ (the Doppler frequency normalized to the data-rate with block size N_{BL}, where f_d is the Doppler frequency, equal to the mobile velocity divided by the carrier wavelength [31], which we will denote as Γ). When Γ is small, the fading process has a strong correlation, which means long bursts of errors (slow fading). Conversely, the occurrence of errors has a weak correlation for large values of Γ (fast fading).

In the equations that are to follow, α is the probability that the ith block of a packet is corrupted, given that the $(i-1)$th block was transmitted successfully, and β is the probability that the ith block of a packet arrives intact, given that the $(i-1)$th block was corrupted. The steady-state packet error-rate ε is then obtained as follows:

$$\varepsilon = \frac{\alpha}{\alpha + \beta}. \tag{2}$$

If the Rayleigh fading margin is F, the average rate of packet errors can be expressed as

$$\varepsilon = 1 - e^{-\frac{1}{F}}. \tag{3}$$

Using Eq. (2) and the equations which follow, we can now derive values for α and β. The average length of a sequence of packet errors is given by $1/\beta$, where

$$\beta = \frac{Q(\theta, \rho\theta) - Q(\rho\theta, \theta)}{e^{\frac{1}{F}} - 1}, \quad \text{and} \quad \theta = \sqrt{\frac{2/F}{1-\rho^2}}. \tag{4}$$

The term ρ is the correlation coefficient of two samples of the complex Gaussian fading process, and is expressed as $\rho = J_0(2\pi\Gamma)$, where $j_0(\cdot)$ is a Bessel function of the first kind and of zeroth order. Additionally,

$$Q(x, y) = \int_y^\infty e^{-\frac{x^2+w^2}{2}} I_0(xw) w \, dw \tag{5}$$

is the Marcum-Q function. Thus, the relationship between the average rate of packet errors and the Markov parameter can be represented as

$$\beta = \frac{1-\varepsilon}{\varepsilon}[Q(\theta, \rho\theta) - Q(\rho\theta, \theta)], \tag{6}$$

where

$$\theta = \sqrt{\frac{-2\log(1-\varepsilon)}{1 - J_0^2(2\pi\Gamma)}}. \tag{7}$$

5.1.2 Fading parameters

In our study, we set Γ variously to 0.001, 0.002, 0.003, 0.01, 0.02 and 0.03. The block size N_{BL} is the packet length. We assumed the use of QPSK modulation with a 1228.8 kb/s physical layer data-rate forward channel. According to the forward-link variable-rate parameters specified in cdma2000 1xEV-DO, a 2048-bit packet length should be used with this modulation, and so $N_{BL} = 2048$ bits [26].

The maximum Doppler frequency of the system is given by $f_c \times v/c$, where v is the mobile speed, c is the speed of the electromagnetic wave, and f_c is the carrier frequency. Therefore, values of Γ of 0.001, 0.002, 0.003, 0.01, 0.02 and 0.03 correspond to mobile speeds of about 1 km/h (s_1), 2 km/h (s_2), 3 km/h (s_3), 10 km/h (s_4), 20 km/h (s_5) and 30 km/h

(s_6), with a reference channel data-rate of 1228.8 kb/s and a 900 MHz carrier frequency. The first three speeds correspond to pedestrian mobiles, and the last three to mobiles in vehicles moving at urban traffic speeds.

5.2 Performance model of RS decoding

First, we analyze the performance of RS error recovery in current BCMCS. Our analysis explains the effects of changing the steady-state packet error-rate (ε) and the speed of the mobiles. We are using $\varepsilon_{residual}$ to denote the rate of upper-layer packet errors in the data (i.e. excluding parity-carrying packets) after as many corrupted packets as possible have been recovered, either by RS coding alone or by the new error recovery scheme.

When a packet is transmitted through a channel, the success or failure of the corresponding sequence of transport packet blocks can be approximated by a two-state Markov chain, as explained in the previous section. Using this model, the probability of success (P_m^s) or failure (P_m^f) of the mth transport packet block is given as follows:

$$P_m^s = (1-\alpha)P_{m-1}^s + \beta P_{m-1}^f, \tag{8}$$

$$P_m^f = \alpha P_{m-1}^s + (1-\beta)P_{m-1}^f, \tag{9}$$

$$P_m^s + P_m^f = 1. \tag{10}$$

The expectation of lost broadcast security-layer packets in a certain ECB, under perfect interleaving conditions (i.e. random errors), can be calculated as follows:

$$E_{RS}[corrupted\ packet] = \sum_{\nu=R+1}^{N} \binom{N}{\nu} \varepsilon^{\nu}(1-\varepsilon)^{N-\nu}\nu. \tag{11}$$

However, this equation cannot be applied when error bursts are too long to interleave them perfectly, which is exactly what we expect to happen when a mobile moves slowly and channel conditions are poor. We have derived an analytic model of the error recovery performance of RS that uses a modified version of Eq. (11) to reflect the imperfect interleaving which occurs under these conditions.

The probability that the length of an error burst is κ when the steady-error rate is ε and the degree of correlation is Γ can be written as:

$$P_{burst}(\kappa \mid \varepsilon, \Gamma) = (1-\beta)^{\kappa-1}\beta. \tag{12}$$

The terms α and β are functions of ε and Γ, and were introduced in the previous section. The probability that RS decoding cannot recover the corrupted packets in an ECB can be formulated in terms of the four cases shown in Fig. 8,

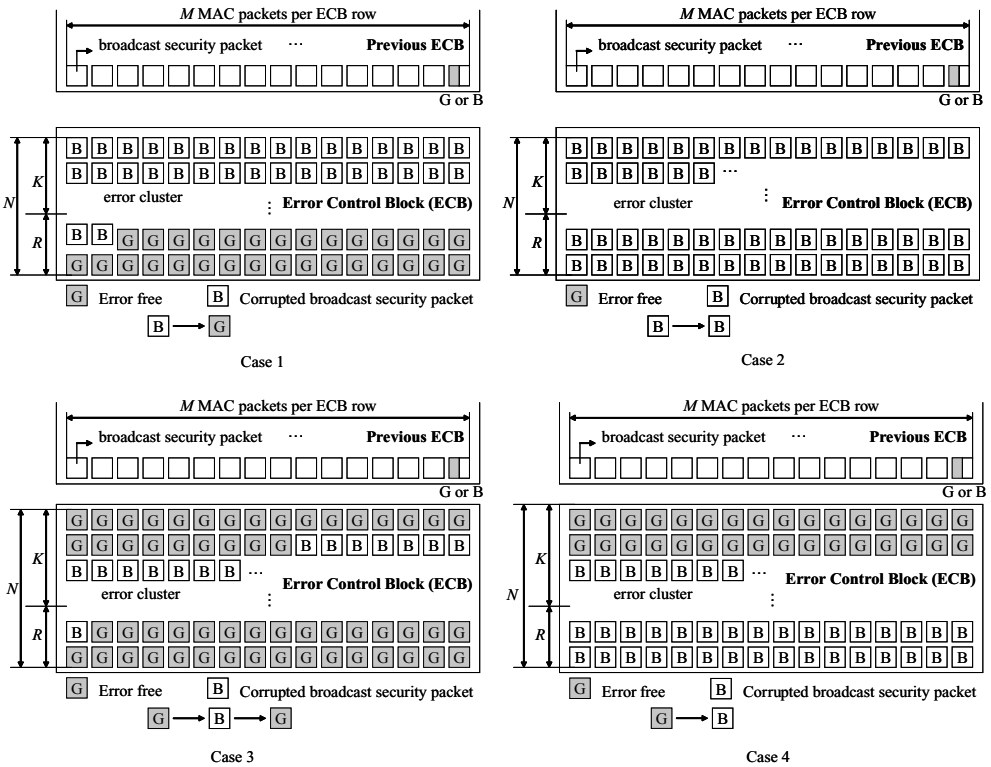


Fig. 8 Four cases in which an (N, K, R) RS code cannot recover the corrupted packets in an ECB

as follows:

$$P_{RS}(failure \mid \varepsilon, \Gamma) = P(case1 \mid \varepsilon, \Gamma) + P(case2 \mid \varepsilon, \Gamma) + P(case3 \mid \varepsilon, \Gamma) + P(case4 \mid \varepsilon, \Gamma). \tag{13}$$

In the first case, transmission of an initial sequence of packets from the current ECB fails due to a burst of errors, but the channel subsequently returns to a good state. On the basis that the intervals between error bursts are long, we will assume that the channel never reverts to a poor state during delivery of the current ECB. In the second case, the first packet of the current control block is corrupted by a burst of errors which continues to the end of the ECB. We now consider the remaining two cases, in which the initial packets are transmitted successfully. In the third case, the channel recovers while the current ECB is still being transmitted. In the fourth and final case, packet delivery fails continuously until the end of the current ECB. In every case, if the error burst is too long, RS decoding cannot recover the corrupted packets. The four variables P_{case1}, P_{case2}, P_{case3} and P_{case4} represent the probability of recovery failing in each case, and are expressed as follows:

$$\begin{aligned}
&P(case1 \mid \varepsilon, \Gamma) \\
&\quad = \Delta_1 \times \left[\sum_{\kappa=RM+1}^{NM-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1} \right], \\
&P(case2 \mid \varepsilon, \Gamma) = \Delta_1 \times (1-\beta)^{NM-1}, \\
&P(case3 \mid \varepsilon, \Gamma) = \Delta_2 \times \sum_{\lambda=1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \\
&\quad \times \left[\sum_{\kappa=RM+1}^{NM-\lambda-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1} \right], \\
&P(case4 \mid \varepsilon, \Gamma) \\
&\quad = \Delta_2 \times \left[\sum_{\kappa=RM+1}^{NM-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1} \right].
\end{aligned} \tag{14}$$

Δ_1 is the probability that the first packet of an ECB is corrupted, whether the last packet of the previous ECB was corrupted or not; and Δ_2 is the probability that the first packet of an ECB is not corrupted, whether the last packet of the previous ECB was corrupted or not. These probabilities are given as follows:

$$\begin{aligned}
\Delta_1 &= ((1-\varepsilon)\alpha + \varepsilon(1-\beta)), \\
\Delta_2 &= ((1-\varepsilon)(1-\alpha) + \varepsilon\beta).
\end{aligned} \tag{15}$$

The expected number of corrupted packets in an ECB, reflecting error burst patterns, can now be obtained by considering four cases, corresponding to the probabilities of Eq. (14), as follows:

$$\begin{aligned}
&E[case1 \mid \varepsilon, \Gamma] = \Delta_1 \times \Phi_{case1}, \quad \text{where} \\
&\Phi_{case1} = \sum_{\kappa=RM+1}^{(R+1)M-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1}\Theta \\
&\qquad + \sum_{\kappa=(R+1)M}^{NM-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1}\kappa.
\end{aligned}$$

$$E[case2 \mid \varepsilon, \Gamma] = \Delta_1 \times (1-\beta)^{NM-1}NM.$$

$$\begin{aligned}
E[case3 \mid \varepsilon, \Gamma] &= \Delta_2 \times \left[\sum_{\lambda=1}^{(N-R-1)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi_{case3} \right] \\
&\quad + \Delta_2 \times \left[\sum_{\lambda=(N-R-1)M-1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi'_{case3} \right],
\end{aligned}$$

where

$$\begin{aligned}
\Phi_{case3} &= \sum_{\kappa=RM+1}^{(R+1)M-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\Theta \\
&\quad + \sum_{\kappa=(R+1)M}^{NM-\lambda-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\kappa,
\end{aligned}$$

and

$$\Phi'_{case3} = \sum_{\kappa=RM+1}^{NM-\lambda-1} P_{burst}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\Theta.$$

$$E[case4 \mid \varepsilon, \Gamma] = \Delta_2 \times \Phi_{case4},$$

where

$$\begin{aligned}
\Phi_{case4} &= \sum_{\kappa=RM+1}^{(R+1)M-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1}\Theta \\
&\quad + \sum_{\kappa=(R+1)M}^{NM-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1}\kappa.
\end{aligned} \tag{16}$$

The variable Θ is $(R+1) \times (\kappa \text{ modulo } M)$. The expected total number of corrupted packets after RS decoding can now be obtained by summing these results:

$$\begin{aligned}
&E_{RS}[corrupted\ packet \mid \varepsilon, \Gamma] \\
&\quad = E[case1 \mid \varepsilon, \Gamma] + E[case2 \mid \varepsilon, \Gamma] \\
&\qquad + E[case3 \mid \varepsilon, \Gamma] + E[case4 \mid \varepsilon, \Gamma].
\end{aligned} \tag{17}$$

Finally, as the packet error-rate (ε) and mobility pattern (Γ) change, the residual error-rate for transmitting packets using the RS scheme can be expressed as

$$\varepsilon_{residual}|_{\varepsilon,\Gamma} = \frac{E_{RS}[corrupted\ packet \mid \varepsilon, \Gamma]}{N \times M}. \quad (18)$$

The residual packet error-rate in the upper layer is a measure of the performance of RS coding. The error recovery rate (ERR) achieved by the RS decoder can be expressed as follows:

$$ERR|_{\varepsilon,\Gamma} = \left[1 - \frac{\varepsilon_{residual}|_{\varepsilon,\Gamma}}{\varepsilon}\right] \times 100. \quad (19)$$

The results provided by this analysis are discussed in the next section.

6 Experimental setup

In this section, we will start by explaining the experimental environment that we used to measure the performance of RS coding and of the proposed hybrid error recovery scheme. The experimental results that we subsequently present and discuss in Section 6.3 show that our scheme outperforms pure RS coding in situations in which mobile nodes move very slowly or have a high packet error-rate, as well as in more favorable conditions.

6.1 Simulation environment

We evaluated performance with two metrics: rate of residual packet errors ($\varepsilon_{residual}$) and video quality (PSNR: peak signal-to-noise ratio). Figure 9 shows the overall experimental structure of our study, which has two main components: one simulates the current and proposed error recovery processes, with reference to the BCMCS specification, and simulates them using the channel model; and the other measures video quality. To evaluate the two error recovery schemes, error traces of the data arriving at the RS decoder in a mobile were obtained by simulating the channel model explained in Section 5.1. Using this model, we can derive error traces, for data transmission over fading channels for a given packet error-rate and also determines the degree of correlation in the

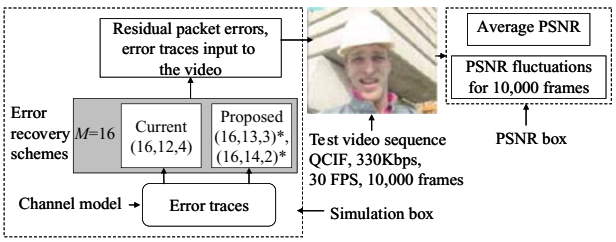


Fig. 9 Simulation structure

fading process. The transition probabilities of the underlying Markov error process are determined by Eqs. (2)–(7).

Once the error traces have been generated, they can be used to simulate the error recovery process of either the current or proposed schemes. The simulator is written in C language and runs in the Linux (Fedora Core 5) environment. The simulation follows the specification of the cdma2000 high rate broadcast-multicast packet data air interface [4].

From our simulation, residual errors and their locations after recovery are generated for each error recovery process, from which we can derive the received video quality. Our simulator automatically injects errors to original video stream using error locations of residual errors. Finally, PSNR [33] values are calculated to estimate the difference in quality between a reconstructed image and an original image using the EvalVid [34] tool.

We used MPEG-4 FGS (fine granular scalable) Foreman QCIF video sequences streamed at 30 frames per second, with a total length of 10,000 frames. These sequences were encoded at 220 kb/s, made up of a base-layer bit-rate of 110 kb/s and an enhancement-layer bit-rate of 110 kb/s, and then forwarded via four logical traffic channels. The total number of subscribers was varied between 10 and 40, distributed uniformly between the logical channels. The average packet error-rate (ε) for all mobiles was varied from 1% to 7%. Each video stream was handled with our reference MPEG-4 FGS codec, which is derived from the framework of the European ACTS Project Mobile Multimedia Systems (MoMuSys) [32], but has been modified for our experiments.

The MPEG-4 FGS decoder was modified so that it can extract error location information from the video stream. A practical way of doing this is to utilize physical and MAC-layer signaling. This allows the decoder to process an entire video stream without any exceptions.

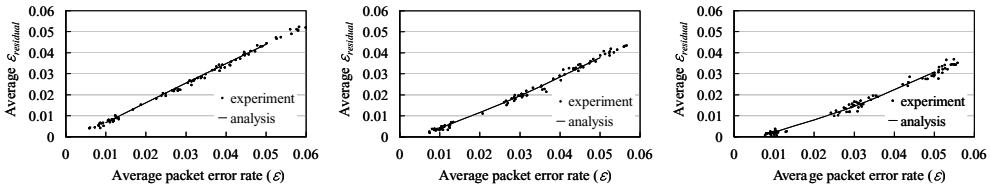
The unmodified decoder includes some error-resilient features, but they cannot always prevent a crisis occurring (this is acknowledged in the readme file of the source code). Using error location information, we can avoid these exceptional situations, and we were therefore able to measure the quality of entire video streams successfully.

We compared our error recovery scheme with the original RS-based scheme employed in BCMCS, using RS codes of (16,12,4), (16,13,3) and (16,14,2), with 16 MAC packets per ECB row ($M = 16$) so as to maximize the error recovery performance. The weight (ω) in the utility function was set to 0.4.

6.2 Performance of the RS scheme

6.2.1 Verification of the analytic model

In Fig. 10, we have plotted the values of $\varepsilon_{residual}$ obtained by simulation for values of ε between 0.005 and 0.06, pedestrian



(a) $\Gamma = 0.001$ (s_1) (b) $\Gamma = 0.002$ (s_2) (c) $\Gamma = 0.003$ (s_3)

Fig. 10 Verification of the analytic model for a (16,12,4) code: analytic and simulation results

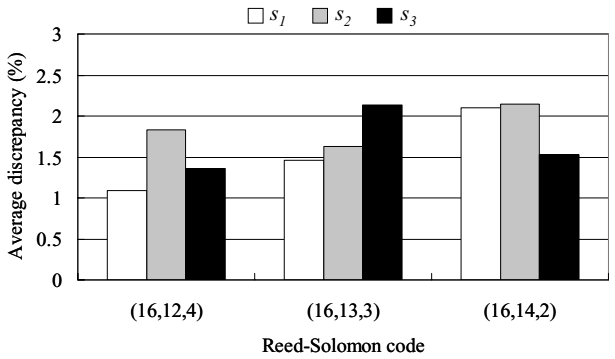


Fig. 11 Average discrepancy between the analytic and simulation results

speeds (s_1, s_2 and s_3), and a (16,12,4) RS code, and compared them to the results derived from our analytic model of error recovery performance. In all three of these graphs, the simulation results are clustered closely around the curve that corresponds to our analytic model. The average discrepancy between the residual rate of packet error predicted by our model ($\varepsilon_{residual}|_{\varepsilon,\Gamma}$) and the rate from the simulation ($\varepsilon_{residual}(\text{sim.})|_{\varepsilon,\Gamma}$) can be expressed as the following percentage:

$$\text{Average discrepancy} = 100 \times \text{Avg}\left[\frac{\varepsilon_{residual}|_{\varepsilon,\Gamma} - \varepsilon_{residual}(\text{sim.})|_{\varepsilon,\Gamma}}{\varepsilon_{residual}(\text{sim.})|_{\varepsilon,\Gamma}}\right]_{\varepsilon_{run}}. \quad (20)$$

A total of 100 values of ε were generated between 0.005 and 0.06, with the majority clustered around 0.01, 0.03 and 0.05.

Values of the average discrepancy between the analytic model and the simulation are shown in Fig. 11 and confirm the accuracy of the model. Most of the results for one run of the simulation are within two standard deviations of the mean, and the average discrepancy is never higher than 2.2%.

6.2.2 Analysis of the error recovery performance of pure RS

Figure 12 shows the performance of the current BCMCS error recovery scheme, as predicted by our analytic model, at pedestrian speeds and under varying channel conditions. We can see that the error recovery capacity of the RS decoder declines significantly for mobiles experiencing poor channel conditions, even when using the (16,12,4) code, which is the best for error recovery. The situation is especially unfavorable for slow-moving mobiles (low values of Γ), indicating that the packet error process is strongly auto-correlated, as shown in Table 1. For a mobile speed of s_1, the recovery rate achieved by the RS decoder is only 31.9% when the packet error-rate at the input to the decoder is 0.01. The situation becomes worse as the channel condition deteriorates, and the recovery rate finally drops to 12.5%, when the rate of packet errors is 0.05. But note that the same error-rate has relatively little effect on a fast-moving mobile, illustrating the decisive effect of the length of error bursts.

The overall picture clearly shows that the RS ECB does not provide sufficient interleaving for the slowest mobiles, even with a (16,12,4) code. This implies an inevitable loss of coverage if high data-rates are used with the current BCMCS specification.

6.3 Comparative performance of our scheme

We will now assess the performance of our scheme against that of the existing pure RS scheme, using the metrics of residual packet loss rate ($\varepsilon_{residual}$) and video playback quality (PSNR).

First, we will compare the error recovery capacity of our our scheme (using (16,13,3) and (16,14,2) codes, which we will mark with an asterisk to indicate that RS coding is supplemented by our hybrid scheme) with the original RS scheme (using a (16,12,4) code for maximum performance). Table 2 shows the results for 10 mobile nodes moving slowly or moderately fast. The relative performance improvement in terms of the residual error-rate of base-layer packets is shown in Fig. 13. As the original RS scheme does not distinguish between the base layer and the enhancement layer when recovering corrupted packets, we present only the average $\varepsilon_{residual}$ for the (16,12,4) code. In fact, the average $\varepsilon_{residual}$ is similar for both layers using this code. But when using other RS codes in the hybrid error recovery scheme, we found some difference between the average value of $\varepsilon_{residual}$ in the base and enhancement layers. We would expect this

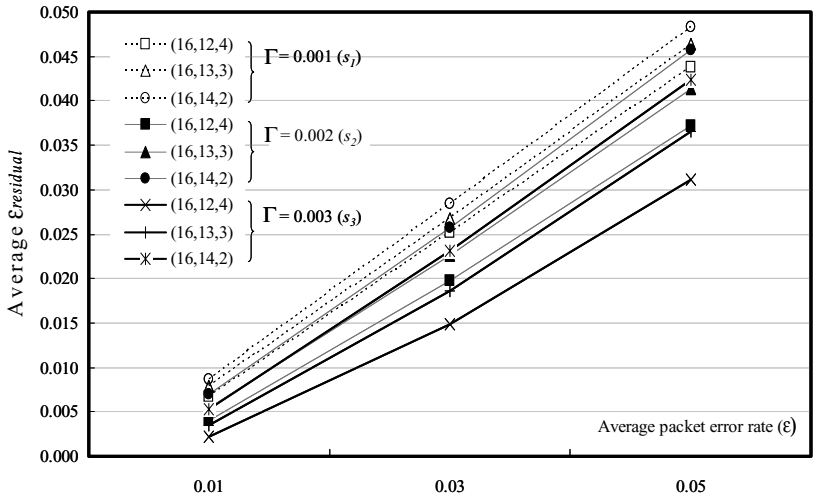


Fig. 12 Analytically determined values of $\varepsilon_{residual}$ for varying channel conditions

Table 1 Error recovery rate ($ERR|_{\varepsilon,\Gamma}$) using a (16,12,4) RS code

		Mobility (Γ)					
RS code	ε	s_1	s_2	s_3	s_4	s_5	s_6
(16,12,4)	0.01	31.9%	60.2%	77.7%	100%	100%	100%
	0.03	17.4%	34.6%	51.5%	94%	99.5%	99.9%
	0.05	12.5%	25.8%	37.8%	84.5%	97.0%	98.9%

Table 2 Simulated values of average $\varepsilon_{residual}$ (the asterisked codes use our hybrid error recovery scheme)

RS code	ε	Γ	Base	Enhance	Average	Γ	Base	Enhance	Average
(16,12,4)	0.01	0.001(s_1)	0.0069060	0.0069102	0.0069081	0.002(s_2)	0.0039759	0.0039784	0.0039772
(16,13,3)*			0.0007651	0.0031148	0.0019399		0.0002164	0.0008167	0.0005165
(16,14,2)*			0.0005354	0.0010462	0.0007908		0.0002376	0.0002534	0.0002455
(16,12,4)	0.03		0.0248288	0.0248336	0.0248312		0.0195700	0.0195740	0.0195720
(16,13,3)*			0.0065305	0.0179135	0.0122220		0.0027848	0.0101110	0.0064479
(16,14,2)*			0.0036974	0.0097753	0.0067364		0.0019043	0.0044474	0.0031759
(16,12,4)	0.05		0.0442695	0.0442813	0.0442754		0.0370077	0.0370171	0.0370124
(16,13,3)*			0.0162439	0.0365028	0.0263734		0.0087885	0.0257869	0.0172877
(16,14,2)*			0.0091687	0.0238684	0.0165185		0.0050330	0.0146044	0.0098187
(16,12,4)	0.07		0.0628899	0.0628929	0.0628914		0.0555034	0.0555086	0.0555060
(16,13,3)*			0.0288008	0.0565876	0.0426942		0.0178768	0.0448228	0.0313498
(16,14,2)*			0.0169262	0.0413146	0.0291204		0.0099320	0.0299658	0.0199489
(16,12,4)	0.01	0.01(s_4)	0.0000172	0.0000169	0.0000170	0.02(s_5)	0.0000004	0.0000003	0.0000003
(16,13,3)*			0.0000012	0.0000013	0.0000012		0.0000001	0.0000000	0.0000000
(16,14,2)*			0.0000103	0.0000100	0.0000101		0.0000020	0.0000011	0.0000015
(16,12,4)	0.03		0.0017320	0.0017305	0.0017312		0.0001435	0.0001407	0.0001421
(16,13,3)*			0.0001812	0.0001786	0.0001799		0.0000304	0.0000285	0.0000294
(16,14,2)*			0.0003950	0.0003519	0.0003734		0.0001340	0.0001239	0.0001290
(16,12,4)	0.05		0.0076877	0.0076857	0.0076867		0.0014655	0.0014710	0.0014682
(16,13,3)*			0.0009916	0.0013416	0.0011666		0.0003006	0.0003014	0.0003010
(16,14,2)*			0.0016261	0.0015454	0.0015857		0.0008181	0.0007779	0.0007980
(16,12,4)	0.07		0.0176752	0.0176715	0.0176734		0.0054380	0.0054280	0.0054330
(16,13,3)*			0.0027482	0.0052102	0.0039792		0.0011662	0.0011482	0.0011572
(16,14,2)*			0.0039041	0.0043217	0.0041129		0.0024547	0.0023428	0.0023987

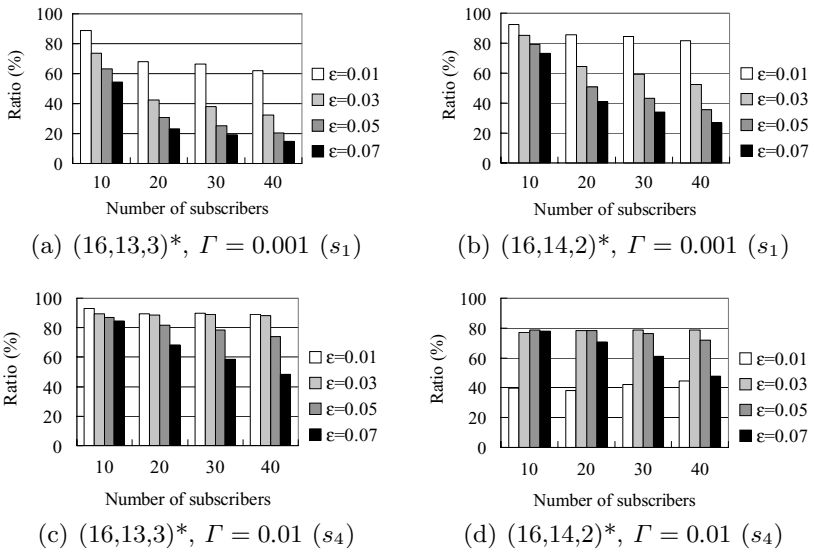


Fig. 13 Relative improvement in the average error-rate of base-layer packets achieved by the hybrid error recovery scheme as the number of subscribers varies

because the hybrid error recovery scheme gives more chance of recovery to base-layer packets to prevent abrupt distortion of the video image, as mentioned in the previous section.

As Table 2 shows, the overall average $\varepsilon_{residual}$ using the hybrid error recovery scheme is lower than that for the original RS (16,12,4) code for all values of ε and Γ. In particular, using the hybrid error recovery scheme, the packet error-rate in the base layer drops significantly compared to the error-rate achieved by the original scheme. The reduction in $\varepsilon_{residual}$ is less pronounced in the base layer than in the enhancement layer for fast-moving nodes. In this situation, error bursts are short, which means that there are enough spare slots to retransmit all corrupted packets. As a result, base-layer packets and enhancement-layer packets have an equal chance of retransmission. However, $\varepsilon_{residual}$ is much lower in the base layer than in the enhancement layer for higher values of ε or larger numbers of subscribers, when there are not enough saved slots to accommodate all retransmission requests. The extent to which the average value of $\varepsilon_{residual}$ in the base layer is reduced by the hybrid error recovery scheme as the number of subscribers increases is shown in Fig. 13. Changes to the average value of $\varepsilon_{residual}$ in the base layer may be taken as a measure of performance, since the base layer is more critical to video quality.

The hybrid scheme improves error recovery performance for all channel conditions and subscriber numbers. As Figs. 13(c) and 13(d) show, our scheme also recovers from more errors in the base layer than the original RS scheme, even at fast-moving nodes. The reduction in the error-rate of base-layer packets reaches almost 90% when the value of $\varepsilon = 0.01$ or 0.03 or 0.05 with a $(16,13,3)^*$ code and $\Gamma = 0.01$. In the case of slow-moving nodes ($\Gamma = 0.001$), the performance ratio between the schemes improves to a maximum of 90% when the value of $\varepsilon = 0.01$ and the number of subscribers is 10. Even in the worst case, when the value of $\varepsilon = 0.07$, the number of subscribers is 40, and we are using a $(16,13,3)^*$ code, there is still a 13% improvement, as shown in Fig 13(a).

The results for the hybrid scheme show that a $(16,14,2)^*$ code outperforms a $(16,13,3)^*$ code, as shown in Fig. 13(b). In general, the improvement is less when there is a large number of subscribers and a high value of ε, because of the lack of physical slots to deliver more retransmission requests. This shows the importance of making good use of the limited number of slots available for retransmission, which is achieved by the utility function already described.

These improvements in error-rate can be shown to affect the playback quality of the video. Figure 14 compares the average playback quality achieved by the hybrid error recovery scheme with that of the current RS scheme, when the value of ε is 1% or 5%. The number of mobile nodes varies from 10 to 40. As Fig. 14 shows, the improvement in PSNR is more prominent for smaller values of Γ, corresponding to slower-moving conditions. The benefit also increases as the packet error-rate of the physical channel deteriorates. Even if the number of subscribers increases to 40, hybrid error recovery gives better video playback quality at slow-moving nodes. Although the gain in quality is more marginal for fast-moving nodes, it is still present. When the channel condition is very good, the current RS scheme and the proposed hybrid error recovery scheme achieve similar values of PSNR, but the relative advantage of the new scheme increases with the value of ε for fast-moving nodes. Overall, our scheme achieves better technical performance under all conditions, and also improves the playback quality at individual mobile nodes. Figure 15 shows the fluctuations of the PSNR for 1000 frames at a mobile node randomly chosen from among 40 nodes when $\varepsilon = 1\%$ and $\Gamma = 0.001$.

From the graphs in Fig. 15, we see that the PSNR fluctuates more significantly when using the original (16,12,4)

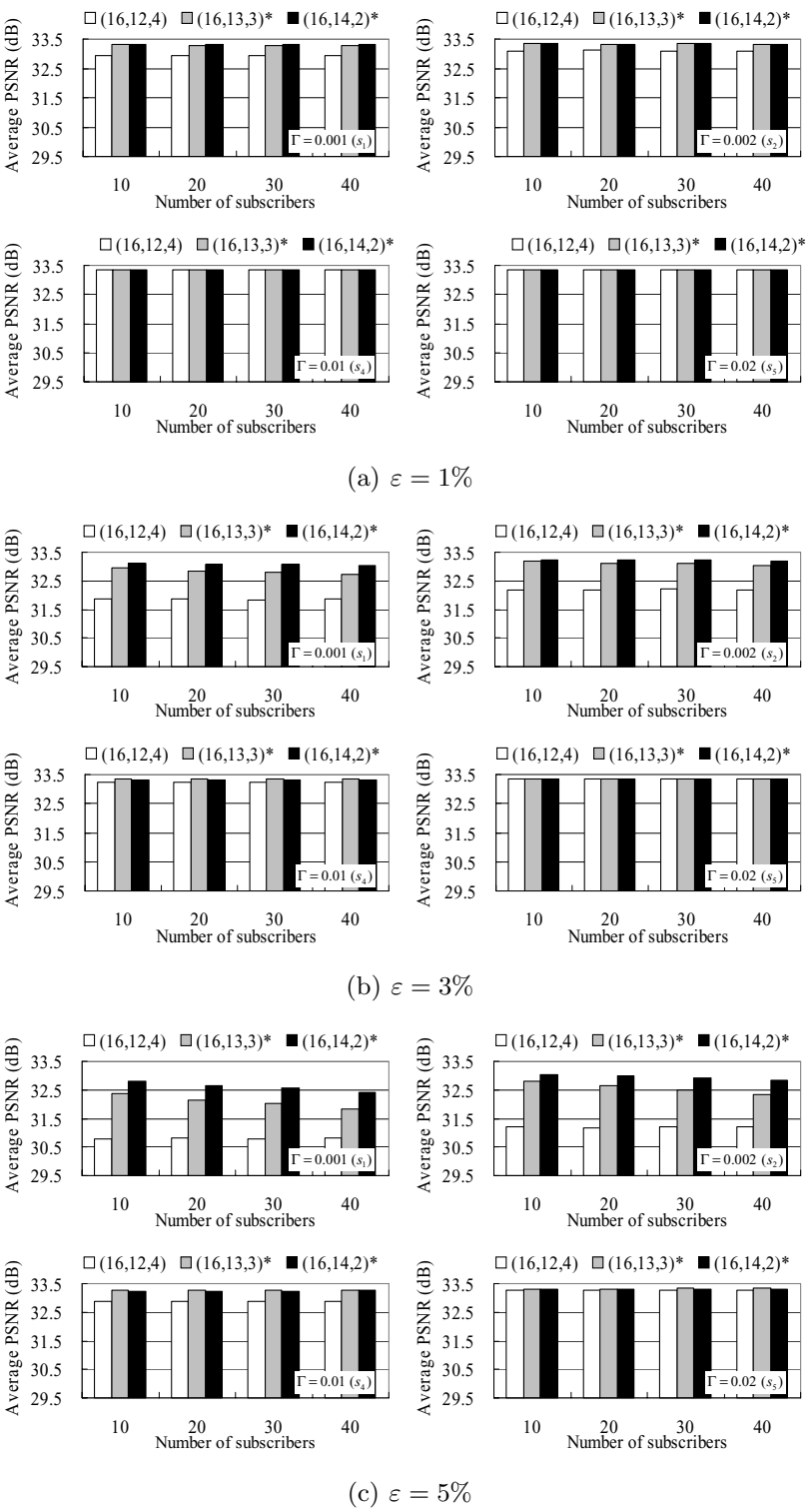
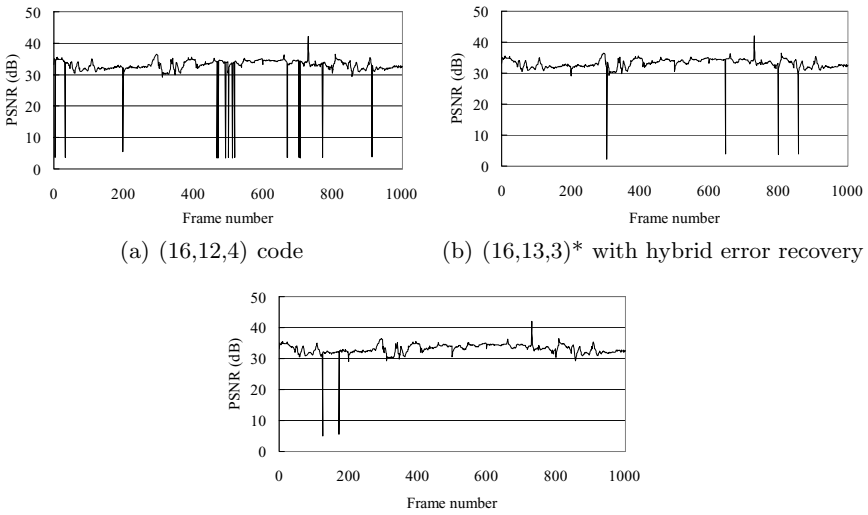


Fig. 14 Comparison of average PSNR at the mobile nodes

RS code than it does with the hybrid scheme. When base-layer packets are corrupted, the PSNR drops below 5 dB, which corresponds to serious distortion of the picture, but the enhancement-layer packets have much less influence on the PSNR. Much better images are obtained with the hybrid error recovery scheme, which tries to restrict errors to the enhancement layer by selective retransmission of base-layer packets. This shows the effectiveness of incorporating awareness of the characteristics of the application. Using our scheme, almost all corrupted base-layer packets are recovered by giving them more chance of retransmission during error recovery, and packet loss is restricted to the enhancement



(a) (16,12,4) code

(b) (16,13,3)* with hybrid error recovery

(c) (16,14,2)* with hybrid error recovery

Fig. 15 The PSNR fluctuations at a mobile node when $\varepsilon = 1\%$ and $\Gamma = 0.001$ (s_1)

layer. This is confirmed by the reduced number of fluctuations shown in Figs. 15(b) and 15(c). Thus the hybrid error recovery scheme delivers a video that can be enjoyed with fewer distractions, and this advantage becomes more noticeable as the value of ε increases.

7 Conclusions

We have investigated the performance of error recovery in the current BCMCS environment for a range of channel conditions, and compared results from our analytic model with experimental data. Additionally, we have proposed a hybrid error recovery scheme, incorporating a packet scheduler, to recover errors and thus improve the playback quality of MPEG-4 FGS encoded video, utilizing the characteristics of a scalable video stream. By using RS and ARQ in concert, we can reduce the packet error-rate in the application layer and prevent the abrupt degradation of playback quality by giving more chance of recovery to more important data (the base layer) with ARQ. This is achieved by cutting the amount of parity information required for RS encoding, and making flexible use of the slots we save. A key element in this approach is to give a high priority to packets that belong to sub-blocks which contain base-layer data and can be recovered with a small number of retransmissions. This is achieved using the proposed utility function. Extensive simulation has shown that the new hybrid error recovery scheme has better performance than the current scheme with respect both to packet error-rate in the application layer and to the average playback quality of the video.

In future, we plan to analyze the performance of other RS codes such as (32,24,8), (32,26,6) and (32,28,4), which are also employed in BCMCS but have somewhat different characteristics.

References

1. J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, E. Tiedemann, and QUALCOMM, Broadcast and multicast services in cdma2000, IEEE Communications Magazine 42(2) (February 2004) 76–82.
2. 3GPP2, X.P0019 v 0.1.3, Broadcast-Multicast Services (BCMCS) Framework Draft Document (August 2003).
3. P. Agashe, R. Rezaiifar, P. Bender, and QUALCOMM, cdma2000 high rate broadcast packet data air interface design, IEEE Communications Magazine 42(2) (February 2004) 83–89.
4. 3GPP2, C.S0054 v1.0, cdma2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification (February 2004).
5. W. Li, Overview of fine granularity scalability in MPEG-4 video standard, IEEE Transactions on Circuits and Systems for Video Technology 11(3) (March 2001) 301–317.
6. F. Wu, S. Li and Y.Q. Zhang, A framework for efficient progressive fine granularity scalable video coding, IEEE Transactions on Circuits and Systems for Video Technology 11(3) (March 2001) 332–344.
7. ISO/IEC 14496-2, Coding of Audio-Visual Objects—Part2 (May 2004).
8. K. Kang, J. Cho and H. Shin, Dynamic packet scheduling for cdma2000 1xEV-DO broadcast and multicast services, in: *Proc. IEEE Wireless Communications and Networking Conference*, vol. 4 (March 2005) pp. 2393–2399.
9. K. Kang, J. Cho, Y. Cho and H. Shin, Dynamic scheduling for scalable media transmission over cdma2000 1xEV-DO broadcast and multicast networks, Lecture Notes in Computer Science 3462 (April 2005) 968–979.
10. W. Li, F. Ling and H. Sun, Bitplane coding of dct coefficients, ISO/IEC JTC1/SC29/WG11, MPEG97/M2691 (October 1997).
11. W. Li and Y. Chen, Experiment result on fine granularity scalability, ISO/IEC JTC1/SC29/WG11, MPEG99/M4792 (March 1999).
12. S.B. Wicker, High-reliability data transfer over land mobile radio channel using interleaved hybrid-ARQ error control, IEEE Transactions on Vehicular Technology 39(1) (February 1990) 48–55.
13. S. Lin and D.J. Costello Jr., *Error Control Coding: Fundamentals and Applications* (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983).
14. S. Lin and P.S. Yu, A hybrid ARQ scheme with parity retransmission for error control of satellite channels, IEEE Transactions on Communications 30(7) (July 1982) 1701–1719.

15. Y.M. Wang and S. Lin, A modified selective repeat type-II hybrid ARQ system and its performance analysis, IEEE Transactions on Communications 31(5) (May 1983) 593–608.
16. Q. Zhang, Hybrid ARQ with selective combining for fading channels, IEEE Journal on Selected Areas in Communications 17(5) (May 1999) 867–880.
17. H. Liu and M.E. Zarki, Performance of H.263 video transmission over wireless channels using hybrid ARQ, IEEE Journal on Selected Areas in Communications 15(9) (December 1997) 1775–1997.
18. B. Vucetic, An adaptive coding scheme for time-varying channels, IEEE Transactions on Communications 39(5) (May 1991) 653–663.
19. A. Shiozaki, K. Okuno, K. Suzuki and T. Segawa, A hybrid ARQ scheme with adaptive forward error correction for satellite communications, IEEE Transactions on Communications 39(5) (April 1991) 482–484.
20. J.A.C. Martins and J.D.C. Alves, ARQ protocols with adaptive block size perform better over a wide range of bit rates, IEEE Transactions on Communications 38(6) (June 1990) 737–739.
21. C.Q. Yang, E. Hossain and V.K. Bhargava, On adaptive hybrid error control in wireless networks using Reed-Solomon codes, IEEE Transactions on Wireless Communications 4(3) (May 2005) 835–840.
22. S.B. Wicker, Reed-Solomon error control coding for Rayleigh fading channels with feedback, IEEE Transactions on Vehicular Technology 41(2) (May 1992) 124–133.
23. J. Lai and N. Mandayam, Packet error rate for burst-error-correcting codes in Rayleigh fading channels, in: *Proc. Vehicular Technology Conference* (May 1998) pp. 1568–1572.
24. J. Lai and N. Mandayam, Performance of Reed-Solomon codes for hybrid-ARQ over Rayleigh fading channels under imperfect interleaving, IEEE Transactions on Communications 48(10) (October 2000) 1650–1659.
25. 3GPP2 C.S0024 v3.0, cdma2000 high rate packet data air interface specification (December 2001).
26. P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi, CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users, IEEE Communications Magazine 38(7) (July 2000) 70–77.
27. M. Bakhuizen and U. Horn, Mobile broadcast/multicast in mobile networks, Ericsson Review (1) (2005).
28. D. Verma, H. Zhang and D. Ferrari, Guaranteeing delay jitter bounds in packet switching networks, in: *Proc. IEEE Conference on Communication Software: Communications for Distributed Applications and Systems* (April 1991) pp. 35–46.
29. M. Zorzi and R.R. Rao, On the statistics of block errors in bursty channels, IEEE Transactions on Communications 45(6) (June 1997) 660–667.
30. M. Zorzi, R.R. Rao and L.B. Milstein, Error statistics in data transmission over fading channels, IEEE Transactions on Communications 46(11) (November 1998) 1468–1477.
31. J. G. Proakis, *Microwave Mobile Communications* (Wiley, 1974).
32. A. Pearmain, A. Carvalho, A. Hamosfakidis and J. Cosmas, The momusys MPEG-4 mobile multimedia terminal, in: *Proc. 3rd ACTS Mobile Summit Conference* (June 1998) pp. 224–229.
33. R.C. Gonzalez and R.E. Woods, *Digital Image Processing* (Addison-Wesley, 1992).
34. J. Klaue, B. Rathke and A. Wolisz, EvalVid—A framework for video transmission and quality evaluation, in: *Proc. 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation* (September 2003) pp. 255–272.

Kyungtae Kang received B.S. (1999) and M.S. (2001) degrees in computer engineering from Seoul National University, Korea. He received Ph.D. degree in Dept. of Electrical Engineering and Computer Science from Seoul National University, Korea in 2007. He is a member of IEEE and IEICE. His research interests include packet scheduling, error control, QoS provision, and energy minimization issues in nextgeneration wireless/mobile networks. In particular, he is researching the performance and energy requirements of 3G cellular broadcast services such as BCMCS and MBMS.

Yongwoo Cho received the Premedical Degree from the College of Medicine, University of Ulsan, in 1997, a B.S. degree in Computer Science from Korea National Open University in 2004, while he was an military service, and an M.S. degree in Electrical Engineering and Computer Science from Seoul National University in 2006. He has worked as a researcher in Dooin Corp. and as a general manager in Bluecord Technology, Inc. His primary interests include multimedia systems, digital broadcasting, next-generation wireless/mobile networks, error control, real-time computing, and low-power design. He is currently a Ph.D. student in the School of Electrical Engineering and Computer Science at Seoul National University.

Heonshik Shin received the B.S. degree in applied physics from Seoul National University, Korea, in 1973. Since he received Ph.D. degree in computer engineering from the University of Texas at Austin in 1985, he has actively involved himself in researches of various topics, ranging from real-time computing and distributed computing to mobile systems and software. He is currently a professor of School of Computer Science and Engineering at Seoul National University.