Error recovery in video transmission over CDMA2000 broadcast networks

K. Kang, Y. Cho, J. Cho and H. Shin

Abstract: Multimedia services over a CDMA2000 broadcast network face a challenge from the unreliable and error-prone nature of the radio channel. Reed–Solomon (RS) coding, integrated with the MAC protocol, is used to cope with this problem. However, performance analysis of RS coding under varying channel conditions shows that it is not always effective, especially for slow-moving nodes which experience relatively long error bursts. Therefore a more efficient scheme is proposed that uses a RS code with reduced parity overhead, and freeing bandwidth can be used flexibly for retransmission. The packets to be retransmitted are prioritised by a utility function derived from the map of the error control block at each mobile node and the number of mobile nodes that require the lost packet. Simulation results show that the gain of retransmission exceeds the loss incurred by reducing the parity, leading to an improvement in the playback quality of MPEG-4 video streams. As a result, service area for high-quality multimedia can be expanded.

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, called multimedia broadcast multicast service (MBMS) [1] and broadcast and multicast services (BCMCS) [2, 3], respectively. The 3GPP2 group has baselined the specification for a CDMA2000 high-rate broadcast packet-data air interface [4, 5]. Their 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. But mobile users also expect low latency when joining or leaving a network, and multimedia streams must be delivered continuously as users move around. A hierarchical design with localised multicasting and local servers is necessary to provide a scalable system, and an efficient air-link must also be designed to ensure that the total throughput of broadcast and multicast services is maximised. However, there has been no research on these topics in the context of CDMA2000 1xEV-DO broadcast and multicast 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.

doi:10.1049/iet-com:20060163

Paper first received 29th March 2006 and in revised form 27th March 2007

K. Kang, Y. Cho and H. Shin are with the Department of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-744, South Korea

J. Cho is with the Department of Computer Engineering, Kyung Hee University, Youngin 449-701, South Korea

E-mail: ktkang@cslab.snu.ac.kr

We analyse the performance of RS error recovery under different channel conditions, while varying the RS parameters. Extensive simulation results will 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, with high data-rate broadcast services, degrades significantly when mobile nodes enter into a zone where the packet error-rate of a channel is high, especially in slow-moving conditions. It is already apparent from previous simulation results [4] that the coverage drops significantly, assuming a dual-receiver access terminal in a time-varying shadowing environment, as the data-rate increases from 409.6 to 1228.8 kb/s, even though the RS code which achieves maximum performance is being used.

In order to improve the performance of RS in video applications, we adopt a hybrid error recovery scheme which includes a packet scheduler to reduce the loss of quality in MPEG-4 fine granular scalability (FGS) video streams [6, 7] by increasing the performance of error recovery. Instead of trying to reduce the error-rate by adding bulky parity information, we use an RS code with a low parity overhead, which can save time-slot resource. This reserved bandwidth can then be used to retransmit corrupted packets using automatic repeat-request (ARQ). The number of retransmissions is necessarily limited, and packets are scheduled for participation by considering the arrangement of MAC packets in the RS error control block (ECB), using a utility function which we will propose, with the aim of improving error recovery capacity and hence also the resulting video playback quality. By means of a realistic simulation, we will show that our scheme is effective in improving video quality and thus increases the coverage area in which a high-quality broadcast service can be received.

The remainder of this paper is organised as follows: In Section 2, we introduce MAC-layer error recovery in BCMCS. Section 3 describes the proposed error recovery scheme and the utility function that it uses. We go on to propose an analytical model of the performance of RS in the current BCMCS environment in Section 4, and our

simulation results are discussed in Section 5. Finally, we conclude the paper in Section 6.

2 MAC-layer error recovery in current BCMCS

In contrast to unicast services operated in accordance with the CDMA2000 1xEV-DO standard, in which a subscriber's forward-link data-rate depends on its RF conditions, BCMCS enable service providers to use a common speed to send video to all subscribers in the area covered by their cells. BCMCS can deliver a consistent high-quality video stream over a large area by using RS coding [4, 5]. In BCMCS, RS coding is applied to the layers above the existing turbo code and is particularly effective in correcting large-scale error bursts.

Fig. 1 shows the structure of the MAC-layer error recovery mechanism at a mobile node. 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 was chosen as the outer code for CDMA2000 BCMCS, and the broadcast MAC layer packets have a fixed size of 125 bytes. The protocol is completed by the broadcast physical layer, and an ECB is transported as payload on one or more subchannels of this layer. Data from multiple ECBs is multiplexed on to the broadcast physical channel.

Each logical channel uses ECBs encoded with the same RS parameters (N, K, $N-K$), 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 an RS codeword, whereas $N-K$ is the number of parity octets: an RS decoder can recover up to $N-K$ octet erasures in each codeword. RS coding is applied to the columns of the ECB, and then the data are transferred row by row to the physical slot, where it forms one or more physical-layer packets. 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.

M in Fig. 1 is the number of MAC packets in an ECB. As the value of M increases, the time diversity also increases

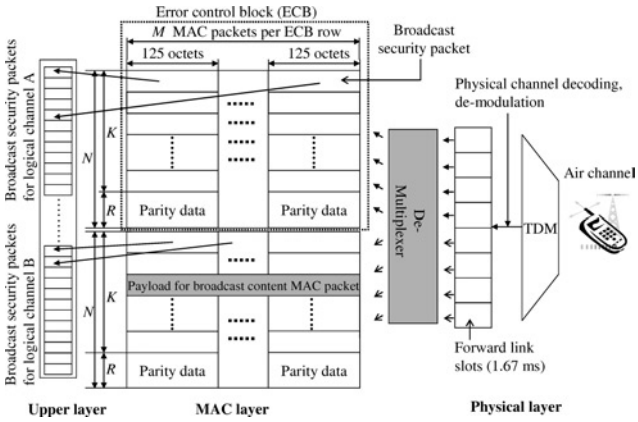

Fig. 1 *Error recovery structure at a receiving node*

and thus a mobile node which is in a time-varying shadow environment is able to recover a substantial amount of corrupted data. According to the BCMCS specification, the value of M for a given ECB has to be less than or equal to 16.

3 Proposed scheme for error correction in the MAC layer

3.1 Description

In current BCMCS, three RS codes can be used to construct the ECB: (16, 12, 4), (16, 13, 3) and (16, 14, 2). In order to transmit 450 slots worth of information, the three RS codes that we are considering 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. These saved slots can be used for the retransmission of packets which were corrupted during the original broadcast. Instead of increasing the RS parity overhead to improve error recovery, we reduce it and compensate by employing the ARQ scheme which is already used in CDMA2000 1xEV-DO unicast services [8] as an effective packet scheduler for the retransmission of packets.

Each mobile node selects target packets for retransmission so as to increase the error recovery capacity and video playback quality, while minimising the number of

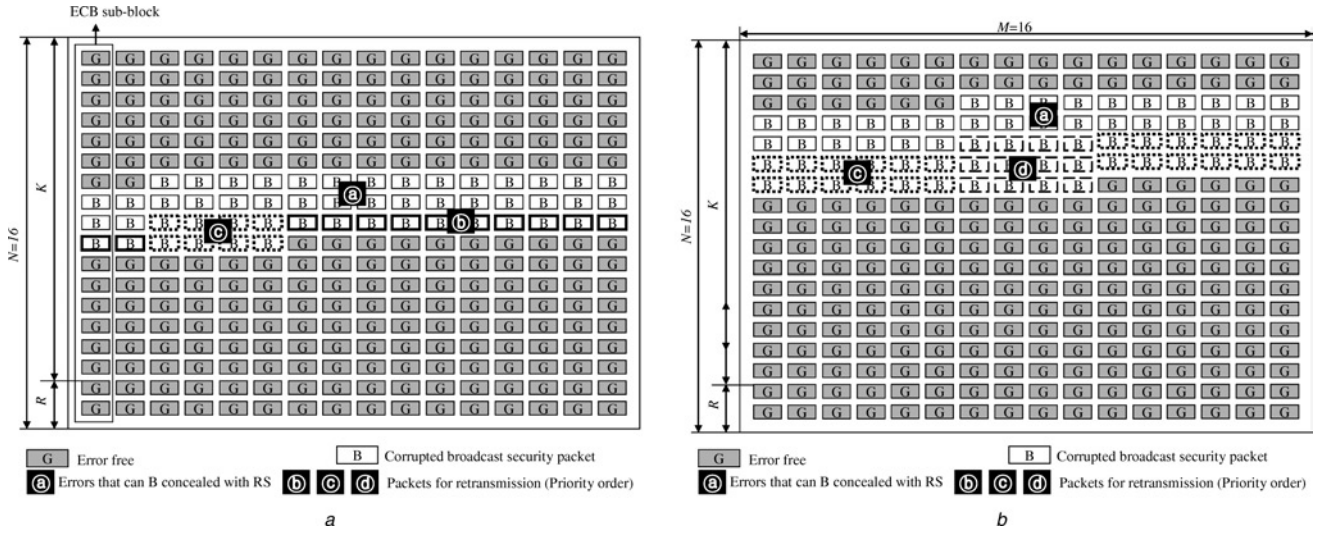

Fig. 2 *Example ECB maps of two mobile nodes*

a ECB of node A
b ECB of node B

packets to be retransmitted. Fig. 2 depicts example ECBs for two mobiles, which we will call node A and node B. 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 examples in Fig. 2, the RS code is (16, 14, 2) and the value of M is 16.

During error recovery, all corrupted packets can be recovered if the errors are restricted to the region marked (a). However, if the error burst is longer than this, a (16, 14, 2) code can no longer correct all the errors. The packets that cannot be corrected are scheduled for retransmission using the ARQ scheme. The number of packets to be retransmitted determines their priority: the fewer there are in a sub-block that needs retransmission, the higher the priority they are allocated.

We will now consider the ECB of node A in Fig. 2. The sequence of corrupted sub-blocks marked (b) can be recovered if just one packet in each sub-block (one of those outlined with bold rectangles) 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 (outlined with dotted-bold rectangles) are successfully retransmitted by ARQ. Similarly, all the packets marked by dotted-bold rectangles (d) in node B are targets for retransmission using ARQ. In this example, the saved slots 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). In this way, a small number of packet retransmissions can use the limited number of reserved slots to make a major improvement to the video playback quality.

In order to prioritise the retransmission of packets with a numerical value, the scheduler uses a function to reflect the utility of each retransmitted packet (τ_i) requested by nodes $\{\mu_0, \mu_1, \ldots, \mu_{\eta-1}\}$, which is defined as follows

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

where N_{node} is the total number of mobile nodes, and $N_{\text{retransmission}}(\tau_i, \mu_\iota)$ 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 each ECB sub-block have the same value of $N_{\text{retransmission}}$ at each mobile node. The reason why we use $[\eta/N_{\text{node}}]$ is to give more priority to the packets which are requested by more mobile nodes, and its effect can be controlled by the weight (ω).

In summary, the chance of a packet being retransmitted is inversely proportional to the number of corrupted packets in the sub-block at each mobile node, and thus naturally reflects the channel condition of that node during retransmission. Also, the chance of retransmitting a packet increases when the packet is required by more mobile nodes. Thus we expect the proposed scheme to increase throughput in a proportionally fair way, which is not true of the legacy hybrid-ARQ scheme.

3.2 Implementation

Each mobile node calculates the value of $N_{\text{retransmission}}$ for all corrupted packets, and reports this information via the reverse ACK channel. The base station schedules the retransmission of dropped packets based on the value of their utility functions: the higher the value, the higher their priority. The scheduled packets are now retransmitted using the forward traffic channel of BCMCS.

The complexity of our scheme can be analysed separately at the base station and at the mobiles. Because all corrupted packets within a single ECB sub-block have the same value of $N_{\text{retransmission}}$, counting the number of corrupted packets in each ECB sub-block is the only task that needs to be performed at every 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, with a complexity of $O(\eta)$. Both these are modest, which suggests that our scheme will run quickly, provided that there is enough buffering both at the base station and at the mobiles. We will 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.

The low complexity of our algorithm at the base station and mobile suggests the possibility of a bound on the delay during scheduling, so long as the end-to-end delay and jitter between the content server and the mobile can be limited by using adequate buffering.

4 Performance analysis of RS coding

4.1 Channel model

Fading in the air channel is assumed to have a Rayleigh distribution. We can model the fate of each data packet with an error generation scheme, which uses the simple threshold model suggested by Zorzi [9, 10] to simulate the error sequences generated by data transmission over a correlated Rayleigh fading channel. A first-order two-state Markov process can simulate these error sequences which occur in clusters or bursts with relatively long error-free intervals between them. We can model different degrees of correlation in the fading process by choosing different values for the physical-layer packet error-rate and for $f_D N_{BL} T$ (which is the Doppler frequency normalised 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). The value of $f_D N_{BL} T$ determines the correlation properties, which are related to the mobile speed for a given carrier frequency. When $f_D N_{BL} T$ 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 $f_D N_{BL} T$ (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 is successful, given that the $(i-1)$th block was unsuccessful. The steady-state error-rate ε is then obtained as follows

$$\varepsilon = \frac{\alpha}{\alpha + \beta} \quad (2)$$

If the Rayleigh fading margin is F, the average physical-layer packet error-rate can be expressed as

$$\varepsilon = 1 - e^{-1/F} \quad (3)$$

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

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

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

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

is the Marcum-Q function. Thus, the relationship between physical-layer packet error-rate and the Markov parameter can be represented as

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

where

$$\theta = \sqrt{\frac{-2\log(1-\varepsilon)}{1 - J_0^2(2\pi f_D N_{BL} T)}}$$

4.2 Effect of ECB size

The BCMCS system varies the size of the ECB to scatter error clusters into a sparse pattern so as to maximise the error recovery performance of the RS decoder. We suggest that, if sufficient interleaving space is provided, the Rayleigh distribution of an error cluster is converted into a random distribution. We will now present a model to support this contention.

The length of time for which the channel stays in good or bad state determines the length of the error bursts and the normal intervals between them. The pattern of fluctuation in the channel condition is made up of a repetition of these sequences of normal and error packets, and we will call the average length of one sequence of normal packets and error packets a template length ($L_{template}$) where

$$L_{template} = \frac{1}{\alpha} + \frac{1}{\beta} (\text{physical-layer packets}) \tag{7}$$

Fig. 3 shows the relation between the size of the ECB (L_{ECB}) and the sampled average value (S_{avg}). We now consider the relative lengths of L_{ECB} and $L_{template}$. If L_{ECB} is smaller than $L_{template}$, then the composition of the samples is not homogenous, and the value of S_{avg} will vary dramatically. Most errors will be localised into a few samples, and S_{avg} will be exaggerated for those samples. The other samples will contain relatively few errors, and for these S_{avg} will naturally be underestimated.

As L_{ECB} converges to $L_{template}$, the fluctuation of S_{avg} will stabilise. After this point, the distribution of errors in the samples will approach more and more closely to the steady-state error-rate as L_{ECB} increases further, and finally S_{avg} saturates to ε as L_{ECB} grows towards ∞. In BCMCS, the errors in each channel are interleaved by making the ECB larger (which also increases the value of M). The size of the ECB (L_{ECB}) is defined as follows

$$L_{ECB}(M) = M \times N(\text{MAC packets}) \tag{8}$$

In our analysis, we set $f_D N_{BL} T$ variously to 0.001, 0.002, 0.003 to simulate very slow-moving pedestrian nodes, and to 0.01, 0.02 and 0.03 to simulate mobile nodes moving pedestrian-speed with physical-layer packet error-rates ($\varepsilon_{physical}$) between 0.01 and 0.07. The block size N_{BL} is the physical-layer packet length. We used quadrature phase-shift keying (QPSK) modulation with a 1228.8 kb/s data-rate forward channel, and the value of N_{BL} was 256 bytes. Also we set the value of M to 16 so as to maximise performance.

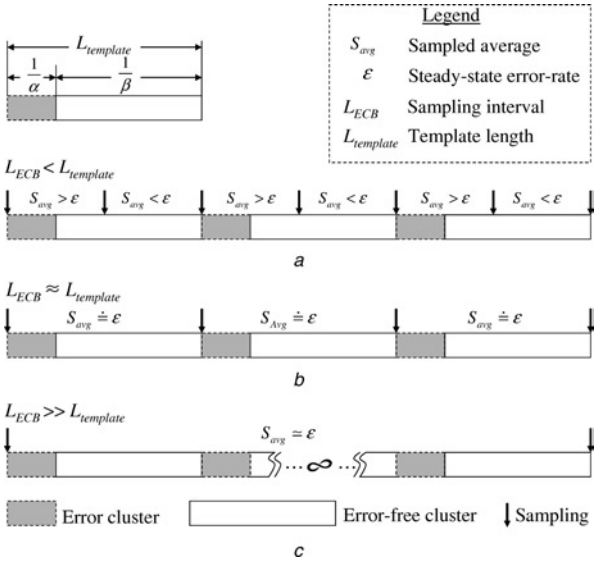

Fig. 3 *Effect of ECB size*

The resulting values of L_{ECB} is 128 physical-layer packets, because each physical-layer packet contains two MAC packets with the modulation just described. Under the channel conditions given above, the minimum average length of a template ($L_{template}$) is 159.5, when $f_D N_{BL} T = 0.01$ and $\varepsilon_{physical} = 0.07$, which is even larger than L_{ECB}. Thus, $L_{template} \gg L_{ECB}$ under these channel conditions, which corresponds to the last case in Fig. 3. The packet error-rate in this situation is analysed in the following section.

4.3 Analysis of the packet error-rate in RS coding

We will now analyse the performance of the current BCMCS scheme in its use of RS coding for error recovery. Our experiments explore the effects of changing the physical-layer packet error-rate (denoted by $\varepsilon_{physical}$), which represents the probability that the packet contained in a certain slot is lost. We will use ε_{upper} to denote the upper-layer packet error-rate of data carrying (not parity carrying) after the corrupted packets have been partly recovered, either by the current RS coding or by the proposed scheme.

When a physical-layer 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. When a node moves very slowly, we expect errors in clusters or bursts. The probability that the length of error cluster is κ can be expressed as

$$P_{error}(\kappa) = (1-\beta)^{\kappa-1}\beta \tag{9}$$

Similarly, the probability that the length of normal cluster is κ' can be expressed as

$$P_{normal}(\kappa') = (1-\alpha)^{\kappa'-1}\alpha \tag{10}$$

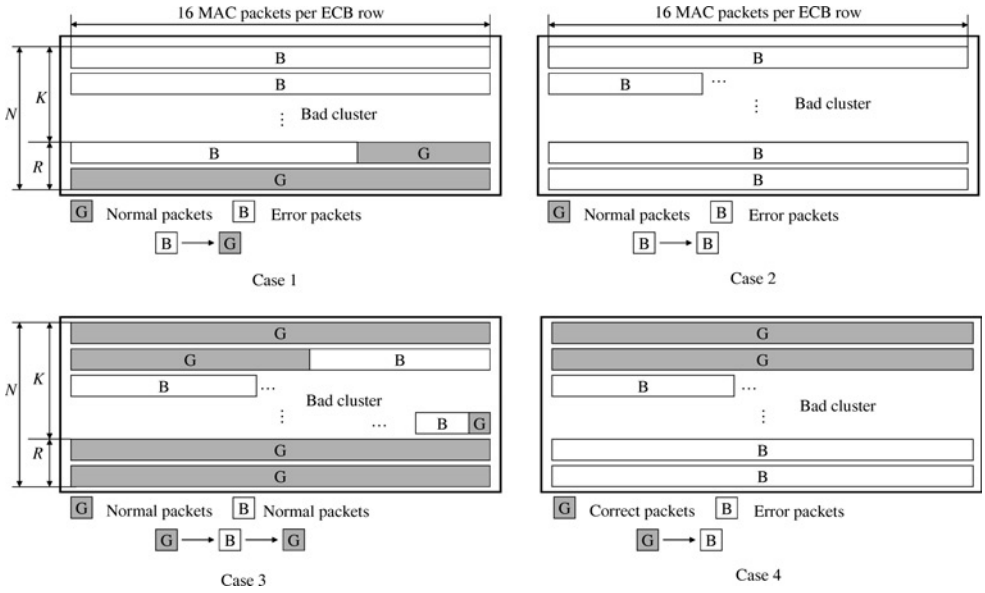

Fig. 4 *Four cases in which an (N, K, N − K) RS code cannot recover the lost packets in an ECB*

If we consider the length of packet errors and assume that they occur with long error-free intervals between them, then the probability that RS cannot recover the lost packets in an ECB can be formulated in terms of the four cases shown in Fig. 4

$$P_{\mathrm{RS}}(\text{failure}) = P_{\text{case1}} + P_{\text{case2}} + P_{\text{case3}} + P_{\text{case4}} \tag{11}$$

In the first case, transmission of an initial sequence of packets in the current ECB fails because of a burst of errors, but the channel subsequently returns to a good state. On the basis that intervals between error bursts are long, we will assume that the state never reverts to bad during delivery of the current ECB. In the second case, the first broadcast security-layer packet of the current control block is corrupted by a burst of errors which continues to the end of the ECB. In the remaining two cases, the initial packets are transmitted successfully. In the third case, the channel state recovers whereas the current ECB is still being transmitted. In the final case, packet delivery fails continuously until the end of the current ECB. In every case, if the error burst is too long, the RS decoder cannot recover the lost 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

$$P_{\text{case1}} = \varepsilon \times \sum_{\kappa=RM+1}^{NM-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-1} \tag{12}$$

$$P_{\text{case2}} = \varepsilon \times (1-\beta)^{NM-1} \tag{13}$$

$$P_{\text{case3}} = (1-\varepsilon) \times \sum_{\lambda=1}^{(N-R)M-2} P_{\text{normal}}(\lambda) \times \left(\sum_{\kappa=RM+1}^{NM-\lambda-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-\lambda-1} \right) \tag{14}$$

$$P_{\text{case4}} = (1-\varepsilon) \times \sum_{\kappa=RM+1}^{NM-1} P_{\text{normal}}(NM-\kappa) \times (1-\beta)^{\kappa-1} \tag{15}$$

Thus, the expected number of lost packets in an ECB, reflecting burst error patterns in the physical-slot layer, can be obtained by considering four cases, corresponding to the probabilities expressed in (12)–(15), as follows

$$E_{\text{case1}} = \varepsilon \times \left(\sum_{\kappa=RM+1}^{(R+1)M-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-1} \Theta + \sum_{\kappa=(R+1)M}^{NM-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-1} \kappa \right) \tag{16}$$

$$E_{\text{case2}} = \varepsilon \times (1-\beta)^{NM-1} \times NM \tag{17}$$

$$E_{\text{case3}} = (1-\varepsilon) \times \left(\sum_{\lambda=1}^{(N-R-1)M-2} P_{\text{normal}}(\lambda) \times \Phi_{\text{case3}} \right) + (1-\varepsilon) \times \left(\sum_{\lambda=(N-R-1)M-1}^{(N-R)M-2} P_{\text{normal}}(\lambda) \times \Phi'_{\text{case3}} \right),$$

where

$$\Phi_{\text{case3}} = \sum_{\kappa=RM+1}^{(R+1)M-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-\lambda-1} \Theta + \sum_{\kappa=(R+1)M}^{NM-\lambda-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-\lambda-1} \kappa, \text{ and}$$

$$\Phi'_{\text{case3}} = \sum_{\kappa=RM+1}^{NM-\lambda-1} P_{\text{error}}(\kappa) \times (1-\alpha)^{NM-\kappa-\lambda-1} \Theta \tag{18}$$

$$E_{\text{case4}} = (1-\varepsilon) \times \left(\sum_{\kappa=RM+1}^{(R+1)M-1} P_{\text{normal}}(NM-\kappa) \times (1-\beta)^{\kappa-1} \Theta + \sum_{\kappa=(R+1)M}^{NM-1} P_{\text{normal}}(NM-\kappa) \times (1-\beta)^{\kappa-1} \kappa \right) \tag{19}$$

The variable Θ is $(R+1) \times (\kappa \text{ modulo } M)$. The expected total number of lost packets can be obtained by summing

these results

$$E_{RS}(\text{lost packet}) = E_{case1} + E_{case2} + E_{case3} + E_{case4} \quad (20)$$

Finally, as the physical-layer error-rate ($\varepsilon_{physical}$) changes, the error-rate in transmitting packets using RS coding can be expressed as

$$\varepsilon_{upper} = \frac{E_{RS}(\text{lost packet})}{NM} \quad (21)$$

The packet error-rate in the upper layer is a measure of the performance of RS coding. The results provided by this analysis are discussed in the next section.

5 Performance evaluation

In this section, we explain the experimental environment which we will use in Section 5.1 to derive the performance of the RS coding scheme and of our proposed scheme. The analytic and experimental results subsequently provided in Section 5.2 will show that our scheme can deal with condition in which mobile nodes move very slowly or have a high packet error-rate, as well as other situations, better than the current method of employing RS coding.

5.1 Experimental environment

Fig. 5 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 error model, and the other measures video quality.

We used the 220 kb/s Foreman testbench video sequences streamed at 30 frames per second, with a total of 10 000 frames. The number of subscribers to each video sequence is uniformly distributed, and results are presented for physical-layer packet error-rates between 1% and 7%, averaged across all mobile nodes. Each video stream is handled with our reference MPEG-4 FGS codec, which is derived from the framework of the European ACTS Project Mobile Multimedia Systems (MoMuSys) [11]. All video streams are channel coded and packetised before being transmitted through a CDMA2000 physical slot.

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 ($M = 16$). The weight (ω) in the utility function is set to 0.4. To evaluate the two error recovery schemes, errors are injected into the original transport stream of a target video sequence and the peak signal-to-noise ratio (PSNR) of the resulting video stream is calculated to estimate the difference in quality between a reconstructed image and an original image.

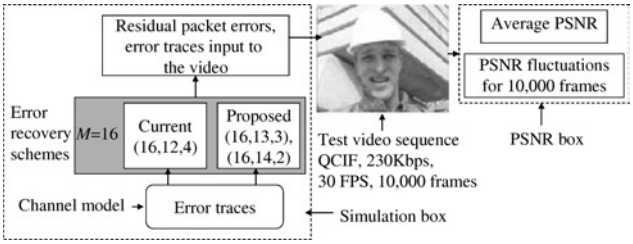

Fig. 5 *Simulation structure*

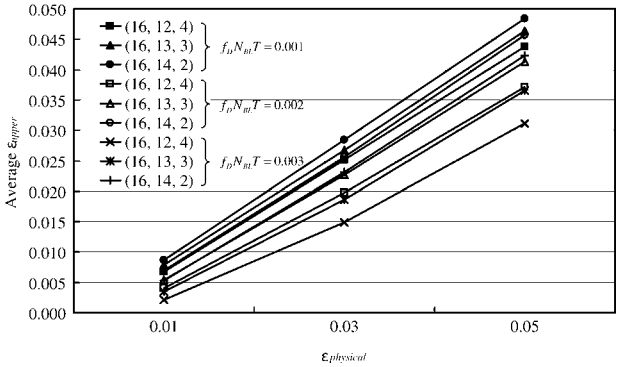

Fig. 6 *Packet error-rate in the upper layer under varying channel conditions: analytic results*

5.2 Experimental results

Fig. 6 shows the performance of the current BCMCS error recovery scheme with a varying average channel condition, for ten mobile nodes. We can see that the error capacity of RS declines suddenly for slow-moving nodes (low values of $f_D N_{BL} T$), and thus the packet error process is strongly auro-correlated. In slow-moving conditions, the errors are so bursty that the ECB does not provide sufficient interleaving, even with a (16, 12, 4) code. These results are also compatible with previous simulation results [4], in which the coverage drops suddenly to below 10% with a dual-receiver access terminal, as the data-rate increases from 409.7 to 1228.8 kb/s, even with a (16, 12, 4) code.

The three graphs in the middle of Fig. 7 show experimental and analytic packet error-rates in the upper layer for a (16, 12, 4) code, as $\varepsilon_{physical}$ varies between 0.01 and 0.07, and $f_D N_{BL} T$ takes values of 0.001, 0.002 and 0.003. In all three of these graphs, the simulation results are clustered closely around the curve that corresponds to our analytical

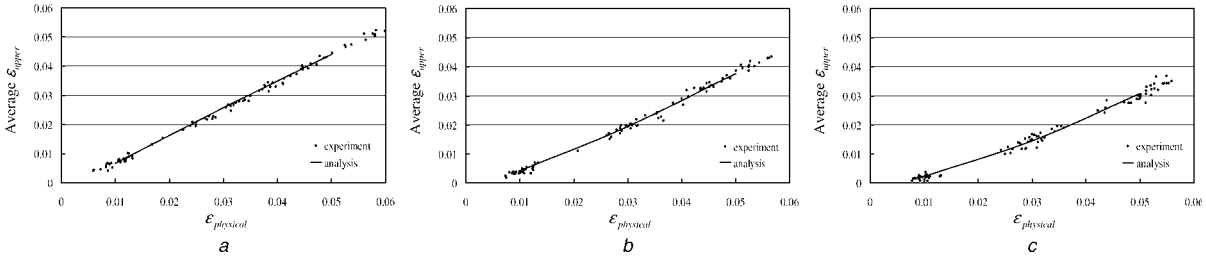

Fig. 7 *Packet error-rate in the upper layer under varying channel conditions: experimental and analytic results for a (16, 12, 4) code*

a $f_D N_{BL} T = 0.001$
b $f_D N_{BL} T = 0.002$
c $f_D N_{BL} T = 0.003$

model of error recovery performance. The average discrepancy between the packet error-rate predicted by our model ($\varepsilon_{\text{upper}}$(model)) and the rate from the simulation ($\varepsilon_{\text{upper}}$(simulation)) can be expressed as follows

$$\text{Average discrepancy rate (\%)} = 100 \times \text{Avg}\left[\frac{|\varepsilon_{\text{upper}}(\text{simulation}) - \varepsilon_{\text{upper}}(\text{model})|}{\varepsilon_{\text{upper}}(\text{simulation})}\right]_{\varepsilon_{\text{run}}} \quad (22)$$

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.

The average discrepancy between the results obtained from the analytic model and the simulation are summarised in Table 1, which confirms the accuracy of the model. The average discrepancy is greater for faster nodes. This is because the error bursts become shorter as a mobile node moves faster, and this in turn increases the probability that an ECB will be received containing more than two error clusters, contradicting the assumption made in Section 4, that there are only four cases. In fact, the results show that the error-rate rarely rises above 5%.

Next, we compared the error recovery capacity of our proposed scheme (using (16, 13, 3) and (16, 14, 2) codes) with the original scheme (using (16, 12, 4)). The results shown in Figs. 8*a* and *b* were obtained by averaging the values of $\varepsilon_{\text{upper}}$ for ten mobile nodes. The results for a mobile node moving very slowly are depicted in Fig. 8*a*: we see that the average value of $\varepsilon_{\text{upper}}$ is reduced from 0.063 to 0.044, using our scheme and a (16, 13, 3) code, and to 0.03, using our scheme and a (16, 14, 2) code; in both cases, $\varepsilon_{\text{physical}} = 0.07$ and $f_{\text{D}}N_{\text{BL}}T = 0.001$. When $\varepsilon_{\text{physical}} = 0.07$ and $f_{\text{D}}N_{\text{BL}}T = 0.003$, the average value of $\varepsilon_{\text{upper}}$ is reduced from 0.048 to 0.015, using a (16, 14, 2) code with our scheme. These results show that, if we reserve slots by reducing the overhead incurred by RS coding, and use those slots flexibly for retransmission, the overall error recovery

Table 1: Average discrepancy of the proposed model

$f_{\text{D}}N_{\text{BL}}T$	RS code (16, 12, 4), %	(16, 13, 3), %	(16, 14, 2), %
0.001	1.09	1.46	2.10
0.002	1.83	1.62	2.15
0.003	1.36	2.13	1.52
0.01	3.78	3.93	4.12
0.02	4.22	4.31	4.82
0.03	4.59	4.72	4.90

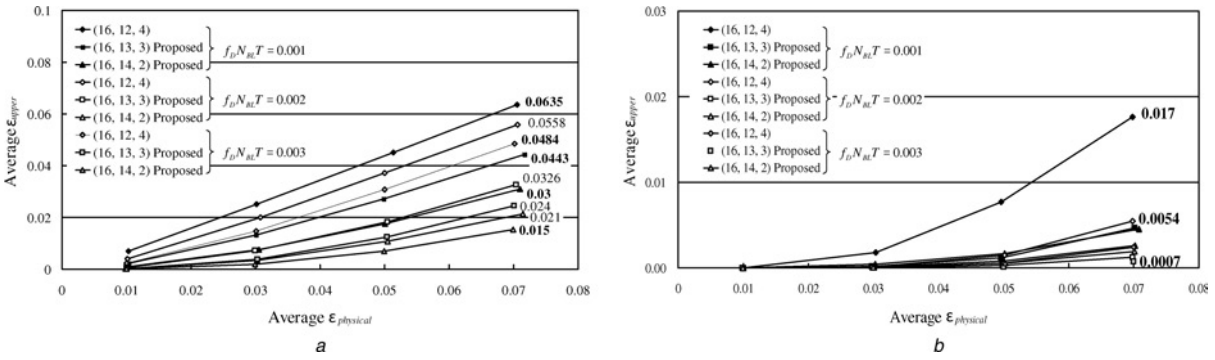

Fig. 8 *Average ε_{upper} for varying values of $\varepsilon_{physical}$*
a $f_{\text{D}}N_{\text{BL}}T = 0.001$, 0.002 and 0.003
b $f_{\text{D}}N_{\text{BL}}T = 0.01$, 002 and 0.03

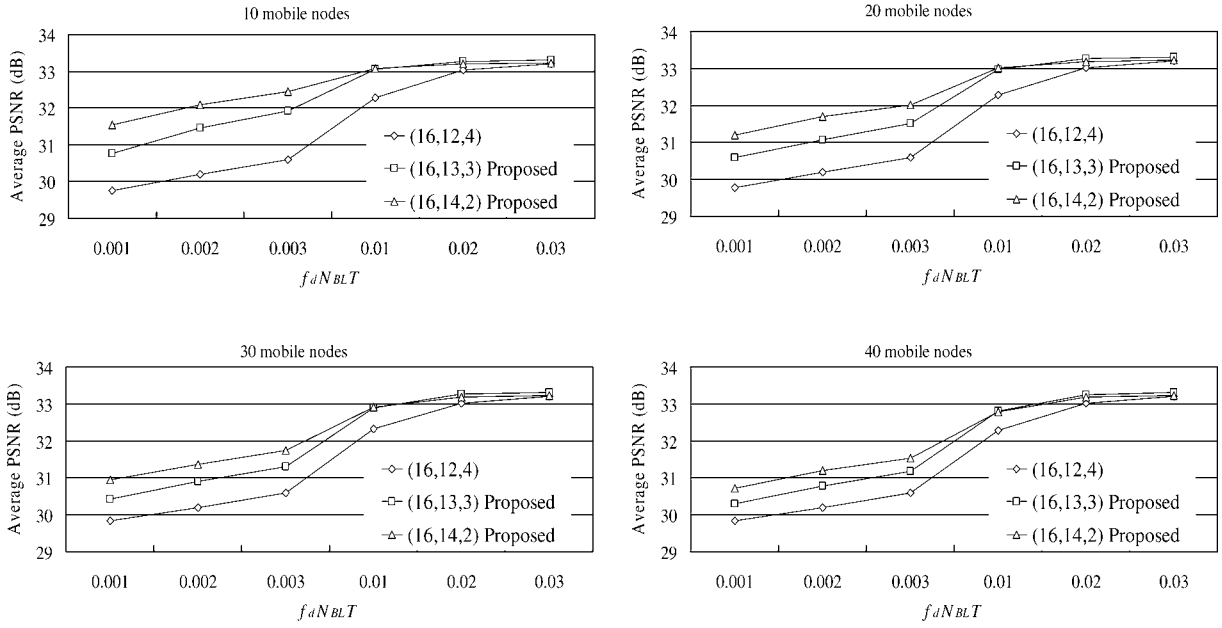

Fig. 9 *Average PSNR when $\varepsilon_{physical} = 0.07$*

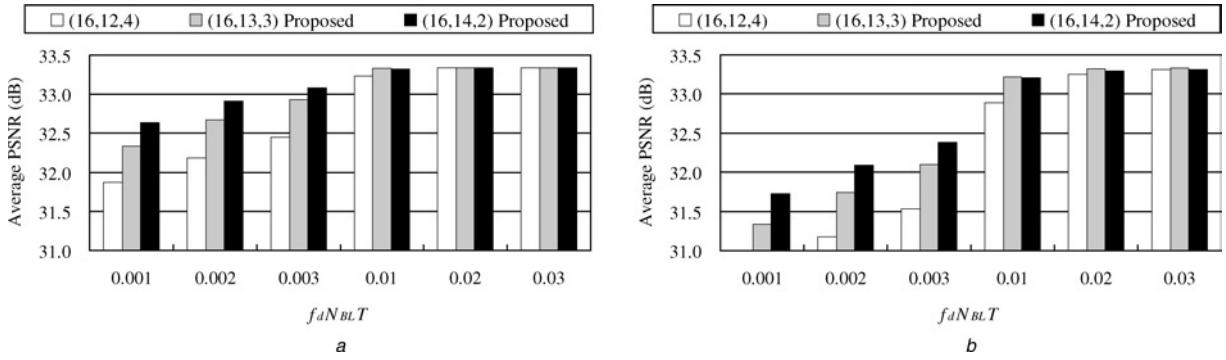

Fig. 10 *Average PSNR of 40 mobile nodes when $\varepsilon_{physical} = 0.03$ or 0.05*

a $\varepsilon_{\text{physical}} = 0.03$
b $\varepsilon_{\text{physical}} = 0.05$

capacity can be improved dramatically as the packet error-rate of the channel increases. Similar tendencies are shown when mobile nodes move around with moderate speed, as depicted in Fig. 8*b*. When $\varepsilon_{\text{physical}} = 0.07$ and $f_D N_{BL} T = 0.01$, $\varepsilon_{\text{upper}}$ is reduced from 0.017 to 0.0054 by using a (16, 14, 2) code with the proposed scheme. Similarly, $\varepsilon_{\text{upper}}$ drops by as much as 0.0007 with the proposed scheme when $\varepsilon_{\text{physical}} = 0.07$ and $f_D N_{BL} T = 0.03$. The relative error recovery performance of our scheme increases as the channel condition deteriorates.

The effect of PSNR on the average playback quality across all the mobile nodes, with $\varepsilon_{\text{physical}} = 7\%$, is shown in Fig. 9, while varying the numbers of subscribers from 10 to 40. The average PSNR of all mobile nodes using the proposed scheme is higher, which means that the playback quality is better than that of the current scheme in all cases. The relative advantage of the proposed scheme is smaller for fast-moving nodes than for slow-moving nodes. Additionally, the gain in average PSNR declines as the number of mobile nodes that receive the same content increases, because more subscribers mean more retransmission of corrupted packets which results in a shortage of slots saved. This reduces the chance of successful retransmission of corrupted packets. However, even those cases, our scheme shows better performance.

The average PSNR of 40 mobile nodes when $\varepsilon_{\text{physical}} = 0.03$ and $\varepsilon_{\text{physical}} = 0.05$ are also presented in Fig. 10. The results for the (16, 12, 4) code were obtained without using the proposed scheme, but in this case more parity information is available to increase performance. The slots saved with the (16, 13, 3) and (16, 14, 2) codes were reassigned by our scheme. These figures show that our scheme improves the overall average quality of the video streams for both values of $\varepsilon_{\text{physical}}$, even though the number of mobile nodes has increased to 40.

Overall, these experiments demonstrate the effectiveness of our proposed error recovery scheme, operating in the MAC layer. They show that it is especially beneficial for mobile nodes in slow-moving conditions, and that its relative advantage increases as the channel conditions deteriorate. It improves the capacity of error recovery by utilising the forward traffic channel more flexibly to improve video quality. As a result, it is possible to expand the coverage area in which a high-quality video broadcast service is possible, bringing watchable video to mobiles far away from the transmitter.

6 Conclusion

We have modelled the performance of RS coding in the current BCMCS environment under a range of channel conditions, and verified its accuracy using a simulation. Subsequent analysis indicates that the performance of RS coding degrades significantly at the edge of the service area where channel conditions are poor, making it impossible to provide multimedia services at a high data-rate. The situation becomes worse when a mobile moves slowly, so that the length of error bursts increases. We have therefore proposed a more efficient hybrid error recovery scheme which combines RS coding with ARQ. The bandwidth for packet retransmission is obtained by using RS codes with less parity information, whereas a utility function prioritises the packets to be retransmitted in a way that maximises throughput. The simulation results show that overall error recovery capacity can be improved dramatically by the proposed scheme, and the relative improvement become greater as the channel conditions deteriorate. The quality of the MPEG-4 video that is received, is also improved, although the gain in average PSNR drops as the number of mobile nodes that receive the same content increases. Overall, the effect is to expand the service area in which a high-quality multimedia service is available.

7 References

1 Boni, A., Launay, E., Mienville, T., and Stuckmann, P.: 'Multimedia broadcast multicast service – technology overview and service aspects'. Proc. IEE Int. Conf. 3G Mobile Communication Technologies, London, UK, October 2004, pp. 634–638
2 C.S0077-0 v1.0: 'Broadcast multicast service for cdma2000 1x systems', April 2006
3 Wang, J., Sinnarajaj, R., Chen, T., Wei, Y., and Tiedemann, E.: 'Broadcast and multicast services in cdma2000', *IEEE Commun. Mag.*, 2004, **42**, (2), pp. 76–82
4 Agashe, P., Rezaiifar, R., and Bender, P.: 'cdma2000 high rate broadcast packet data air interface design', *IEEE Commun. Mag.*, 2004, **42**, (2), pp. 83–89
5 C.S0054-A v1.0: 'cdma2000 high rate broadcast-multicast packet data air interface specification', March 2006
6 Li, W.: 'Overview of fine granularity scalability in MPEG-4 video standard', *IEEE Trans. Circuits Syst. Video Technol.*, 2001, **11**, (3), pp. 301–317
7 ISO/IEC 14496-2: 'Coding of Audio-Visual Objects - Part 2: Visual', May 2004
8 Bender, P., Black, P., Grob, M., Padovani, R., Sindhushayana, N., and Viterbi, A.: 'CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users', *IEEE Commun. Mag.*, 2000, **38**, (7), pp. 70–77
9 Zorzi, M., and Rao, R.R.: 'On the statistics of block errors in bursty channels', *IEEE Trans. Commun.*, 1997, **45**, (6), pp. 660–667
10 Zorzi, M., Rao, R.R., and Milstein, L.B.: 'Error statistics in data transmission over fading channels', *IEEE Trans. Commun.*, 1998, **46**, (11), pp. 1468–1477
11 Pearmain, A., Carvalho, A., Hamosfakidis, A., and Cosmas, J.: 'The MoMuSys MPEG-4 mobile multimedia terminal'. Proc. ACTS Mobile Summit Conf., Rhodes, Greece, June 1998, pp. 224–229