CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:1171–1181
Published online 3 April 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1160

Augmenting Reed–Solomon coding with retransmission for error recovery in 3G video broadcasts

Kyungtae Kang*,†, Yongwoo Cho and Heonshik Shin

School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151–744, Korea

SUMMARY

The error-prone nature of the radio channel is the major challenge in servicing video streams over cdma2000 broadcast networks. The MAC protocol for broadcast and multicast services (BCMCS) in cdma2000 specifies forward error correction using Reed–Solomon coding, which is effective in recovering from bursts of errors. However, its performance degrades significantly when channel conditions are bad, a frequent occurrence at the edge of the coverage area, reducing the availability of high-speed broadcasts. We propose an error recovery scheme and a scheduling algorithm based on the use of slots saved by changing to a Reed–Solomon code with a lower parity overhead. Within the fixed transmission budget thus created, corrupted packets are retransmitted in a priority order determined by a utility function that is derived from the map of the error control block at each mobile, and which also considers the number of mobiles that did not receive each lost packet. Simulation results show the effectiveness of the proposed scheme in improving the quality of high-data-rate MPEG-4 video streams over a range of channel conditions. Copyright © 2007 John Wiley & Sons, Ltd.

Received 29 October 2006; Accepted 21 November 2006

KEY WORDS: broadcast and multicast services (BCMCS); Reed–Solomon; Automatic Repeat Request (ARQ); MPEG-4 fine granular scalable (FGS)

1. INTRODUCTION

3GPP2 BCMCS (broadcast and multicast services) comprise a novel multicast-for-mobile solution for cdma2000 networks that includes a flexible common radio channel suitable for point-to-multipoint and broadcast traffic [1]. Aiming to standardize associated services, 3GPP2 has recently baselined the

*Correspondence to: K. Kang, School of Electrical Engineering and Computer Science, Seoul National University, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151–744, Korea.
†E-mail: ktkang@cslab.snu.ac.kr

specification for the cdma2000 high-rate broadcast packet-data interface [2,3]. In BCMCS, the MAC protocol uses Reed–Solomon (RS) coding as the method of forward error correction.

With a dual-receiver access terminal, a channel data-rate of at most 614.4 kb s^{-1} can be supported by a (16,14,2) RS code in no more than 90% of the coverage area, assuming that the rate of packet loss in the upper layer is not to exceed 0.01 (see [2]). Therefore, a high-data-rate broadcast service cannot rely entirely on RS without losing some coverage.

In this paper, we propose a performance model of RS error recovery and analyze it under different channel conditions and at different mobile speeds, while varying the RS encoding parameters. Extensive simulation results will show that the resulting analytic model can accurately predict the error recovery performance of RS, and that RS is not effective at the edge of the coverage area, especially when a mobile is moving slowly in bad channel conditions, and thus experiences relatively long error bursts.

Our objective is to suggest a more efficient error recovery scheme, combined with a packet scheduler, to increase the proportion of the coverage area in which a high-speed video broadcast is available. Instead of adding bulky parity information to improve the performance of RS coding, we use a RS code with a lower parity overhead to save time-slot resource. This reserved bandwidth is then employed to retransmit corrupted packets selectively. The number of retransmissions is necessarily limited, and packets are scheduled for participation using a utility function which takes account of the arrangement of MAC packets in the RS error control block (ECB).

Our aim is to improve error recovery capacity selectively, in a way that maximizes the resulting video playback quality and increases the effective coverage of a video broadcast. We use a realistic simulation to show that our scheme is indeed effective in improving video quality, even at the edge of the coverage area where the channel condition can be extremely bad.

2. MAC-LAYER ERROR RECOVERY IN CURRENT BCMCS

Each logical channel uses ECBs encoded with the same RS parameters (N, K, R), and has M MAC packets per ECB row. K is the number of security-layer octets, R is the number of parity octets, and $N = K + R$. For a small value of K, RS provides improved error correction, but at the cost of a lower effective data-rate for the broadcast service. As M increases, the time-diversity also increases, and thus a mobile in a time-varying shadow environment can still recover a substantial amount of corrupted data. In BCMCS the value of M for a given ECB has to be less than or equal to 16.

3. PROPOSED ERROR RECOVERY SCHEME

3.1. Description

The components of the BCMCS MAC protocol are shown in Figure 1. Three RS codes may be used to construct the ECB: (16,12,4), (16,13,3), or (16,14,2). Figure 1 shows the amount of data corresponding to information and parity that is broadcast during 1 second of transmission. There are 600 slots per second and the duration of each slot is therefore 1.67 ms. In transmitting 450 slots worth of information, parity data takes up 150, 104, and 45 slots, respectively, for the three RS codes.

DOI: 10.1002/cpe

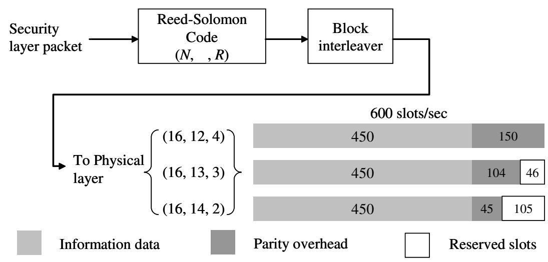


Figure 1. Broadcast MAC protocol components and RS overhead.

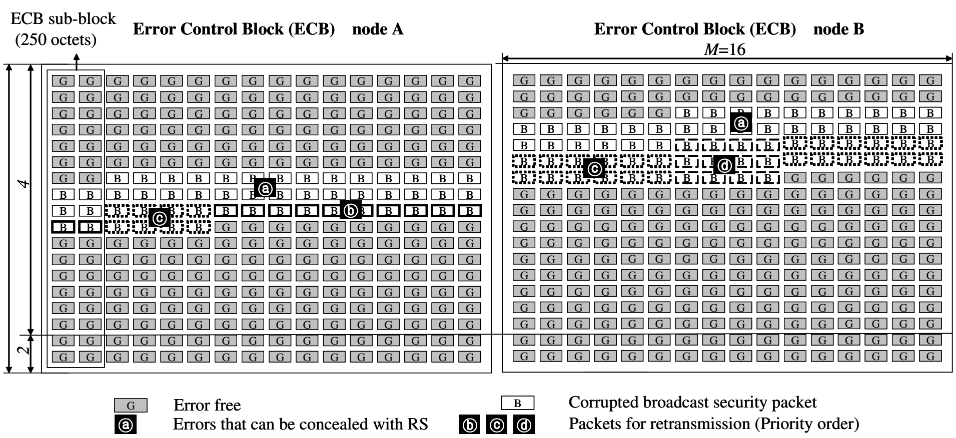


Figure 2. Example ECB maps at two mobiles; the RS code is (16,14,2).

This leaves 46 slots unallocated using the (16,13,3) code, and 105 unallocated by the (16,14,2) code, as shown in Figure 1. These saved slots can be used to retransmit corrupted broadcast packets. We compensate for the reduction in parity by employing ARQ, which is already used in cdma2000 1xEV-DO unicast services [4].

An ECB map shows how each mobile targets packets for retransmission, to maximize the error recovery capacity while minimizing the number of packets to be retransmitted. Figure 2 depicts the ECB map for two mobiles. In the middle of each of these example ECBs is an error cluster. When a mobile enters an area where the channel condition is poor, packets are sequentially corrupted for a certain period. In the example in Figure 2, the RS code is (16,14,2) and M is 16; G stands for a security packet received successfully while B is a corrupted packet.

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

We now consider the ECB of mobile A in Figure 2. The sequence of corrupted sub-blocks marked (b) can be recovered if just one packet in each sub-block (those outlined with bold rectangles) is successfully retransmitted using the slots saved by a (16,14,2) code. All of the corrupted sub-blocks marked (c) can be recovered if more than two packets (outlined with dotted-bold rectangles) are successfully retransmitted. Similarly, all of the packets marked by dotted-bold rectangles (d) in mobile B are candidates for retransmission. In this example, the slots saved by a (16,14,2) code are sufficient to retransmit all of the corrupted packets. However if there is a shortage of slots, the packets belonging to the region marked (b) will be sent 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). This ensures that the small number of packet retransmissions that are available make the largest possible improvement in the video playback quality.

3.2. Packet scheduling based on a utility function

In order to obtain the numerical values on which the retransmission of packets is based, the scheduler uses a function that reflects the utility of each retransmitted packet (τ_i) requested by mobiles $\{\mu_0, \mu_1, \ldots, \mu_{\eta-1}\}$, and which is defined as follows:

$$f(\tau_i) = \omega\left[\frac{\eta}{N_{\text{node}}}\right] + (1-\omega)\left[1 - \frac{1}{\eta}\sum_{\iota=0}^{\iota=\eta-1}\frac{N_{\text{need}}(\tau_i, \mu_\iota)}{N}\right] \tag{1}$$

where N_{node} is the total number of mobiles, and $N_{\text{need}}(\tau_i, \mu_\iota)$ is the number of corrupted packets that need to be retransmitted in the ECB sub-block corresponding to τ_i, in order to recover τ_i successfully in mobile μ_ι. All of the corrupted packets in a single ECB sub-block have the same value of N_{need}. The term $[\eta/N_{\text{node}}]$ ensures that the more popular packets receive a higher priority, and its effect is controlled by the weight ω.

4. PERFORMANCE ANALYSIS OF MAC-LAYER ERROR RECOVERY

4.1. Channel model

Fading in the air channel is assumed to have a Rayleigh distribution. A first-order two-state Markov process can simulate the error sequences encountered during transmission on a correlated Rayleigh fading channel, and we model the fate of each data packet using the simple threshold model suggested by Zorzi *et al*. [5]. We can also model different degrees of correlation in the fading process by choosing different values for the packet loss rate and $f_d N_{BL} T$ (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, which we call Γ).

The value of Γ determines the correlation properties, which are related to the mobile speed for a given carrier frequency. 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 arrives intact, given that the $(i-1)$th block was corrupted. These Markov parameters can be obtained elsewhere [6].

In our study, we set Γ variously to 0.001, 0.002, 0.003, 0.01, 0.02, and 0.03, which correspond to mobile speeds of about 1 km h^{-1} (s_1), 2 km h^{-1} (s_2), 3 km h^{-1} (s_3), 10 km h^{-1} (s_4), 20 km h^{-1} (s_5), and 30 km h^{-1} (s_6), with a reference channel data-rate of 1228.8 kb s^{-1} and a 900-MHz carrier frequency. The first three speeds correspond to pedestrian mobiles, and the last three to vehicles moving in urban traffic. We assumed the use of Quadrature Phase Shift Keying (QPSK) modulation with a 1228.8 kb s^{-1} physical layer data-rate forward channel. According to the forward-link variable-rate parameters in cdma2000 1xEV-DO, this implies a 2048-bit packet length, and so $N_{\mathrm{BL}} = 2048$ bits [4].

4.2. Modeling RS error recovery performance

We use $\varepsilon_{\mathrm{residual}}$ to denote the rate of upper-layer packet loss 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 scheme.

We have derived an analytic model of the error recovery performance of RS to reflect imperfect interleaving conditions, when error bursts are too long to interleave perfectly, which is exactly what we expect to when a mobile moves slowly and channel conditions are poor.

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

$$P_{\mathrm{burst}}(\kappa \mid \varepsilon, \Gamma) = (1-\beta)^{\kappa-1}\beta \tag{2}$$

The probability that RS decoding cannot recover the lost packets in an ECB can be formulated in terms of four cases:

$$P_{\mathrm{RS}}(\mathit{failure} \mid \varepsilon, \Gamma) = P(\mathit{case}1 \mid \varepsilon, \Gamma) + P(\mathit{case}2 \mid \varepsilon, \Gamma) + P(\mathit{case}3 \mid \varepsilon, \Gamma) + P(\mathit{case}4 \mid \varepsilon, \Gamma) \tag{3}$$

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 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. In the remaining two cases, the initial packets are transmitted successfully. In the third case, the channel recovers while 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, RS decoding cannot recover the lost packets. The four variables $P(\mathit{case}1)$, $P(\mathit{case}2)$, $P(\mathit{case}3)$, and $P(\mathit{case}4)$ represent the probability of recovery failing in each case, and are expressed as follows:

$$P(case1 \mid \varepsilon, \Gamma) = \Delta_1 \times \left[\sum_{\kappa=RM+1}^{NM-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1} \right]$$
$$P(case2 \mid \varepsilon, \Gamma) = \Delta_1 \times (1-\beta)^{NM-1}$$
$$P(case3 \mid \varepsilon, \Gamma) = \Delta_2 \times \sum_{\lambda=1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \times \left[\sum_{\kappa=RM+1}^{NM-\lambda-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1} \right] \quad (4)$$
$$P(case4 \mid \varepsilon, \Gamma) = \Delta_2 \times \left[\sum_{\kappa=RM+1}^{NM-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1} \right]$$

Δ_1 is the probability that the first packet of an ECB is corrupted if 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 when the last packet of the previous ECB was corrupted or not. These probabilities follow:

$$\Delta_1 = (1-\varepsilon)\alpha + \varepsilon(1-\beta) \quad \text{and} \quad \Delta_2 = (1-\varepsilon)(1-\alpha) + \varepsilon\beta \quad (5)$$

The expected number of lost packets in an ECB, reflecting error burst patterns, can now be obtained by considering four cases, corresponding to the probabilities of (4), as follows:

$$E[case1 \mid \varepsilon, \Gamma] = \Delta_1 \times \Phi_{case1},$$
$$\Phi_{case1} = \sum_{\kappa=RM+1}^{(R+1)M-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1}\Theta + \sum_{\kappa=(R+1)M}^{NM-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-1}\kappa$$
$$E[case2 \mid \varepsilon, \Gamma] = \Delta_1 \times (1-\beta)^{NM-1}NM$$
$$E[case3 \mid \varepsilon, \Gamma] = \Delta_2\left[\sum_{\lambda=1}^{(N-R-1)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi_{case3} \right] + \Delta_2 \times \left[\sum_{\lambda=(N-R-1)M-1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi'_{case3} \right],$$
$$\Phi_{case3} = \sum_{\kappa=RM+1}^{(R+1)M-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\Theta + \sum_{\kappa=(R+1)M}^{NM-\lambda-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\kappa$$
$$\Phi'_{case3} = \sum_{\kappa=RM+1}^{NM-\lambda-1} P_{\text{burst}}(\kappa \mid \varepsilon, \Gamma)(1-\alpha)^{NM-\kappa-\lambda-1}\Theta$$

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

$$\Phi_{case4} = \sum_{\kappa=RM+1}^{(R+1)M-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1}\Theta + \sum_{\kappa=(R+1)M}^{NM-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1}\kappa \tag{6}$$

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

$$E_{RS}[lost\ packet \mid \varepsilon, \Gamma] = E[case1 \mid \varepsilon, \Gamma] + E[case2 \mid \varepsilon, \Gamma] + E[case3 \mid \varepsilon, \Gamma] + E[case4 \mid \varepsilon, \Gamma] \tag{7}$$

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

$$\varepsilon_{\text{residual}} \mid \varepsilon, \Gamma = \frac{E_{RS}[lost\ packet \mid \varepsilon, \Gamma]}{N \times M} \tag{8}$$

The residual packet loss rate in the upper layer is a measure of RS performance.

5. PERFORMANCE EVALUATION

5.1. Experimental environment

We evaluated the performance with two metrics: the rate of residual packet loss and video quality. We used 10 000-frame MPEG-4 FGS (fine granular scalable) [7,8] Foreman QCIF video sequences streamed at 30 frames per second. The sequences are encoded at 230 kb s^{-1}, made up of a base-layer bit-rate of 130 kb s^{-1} and an enhancement-layer bit-rate of 100 kb s^{-1}, and 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 loss rate for all mobiles was varied from 1% to 5%. Each video stream is handled with our reference MPEG-4 FGS codec, which is modified version of the framework of the European ACTS Project Mobile Multimedia Systems (MoMuSys) [9].

We compared our error recovery scheme with the original BCMCS scheme using RS codes of (16,12,4), (16,13,3), and (16,14,2), with 16 MAC packets per ECB row ($M = 16$) for maximum error recovery performance. The weight (ω) in the utility function was set to 0.4. 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 of Section 4.1.

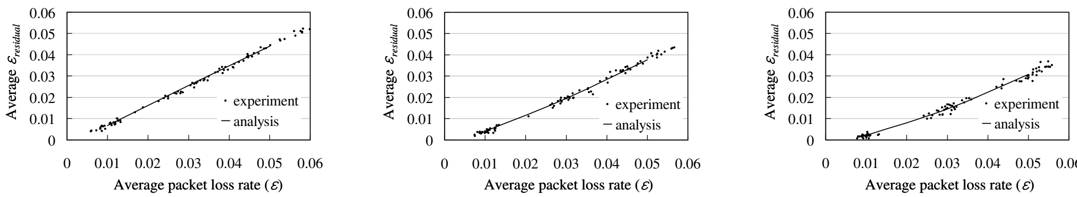
5.2. Performance of the existing scheme

5.2.1. Verification of the analytic model

In Figure 3 we have plotted the values of $\varepsilon_{\text{residual}}$ obtained by simulation for values of ε between 0.005 and 0.06, at pedestrian speeds (s_1, s_2, and s_3), and compared them with the results derived from our analytic model for a (16,12,4) RS code. In all three graphs the simulation results are clustered closely around the curve corresponding to our analytical model of error recovery performance.

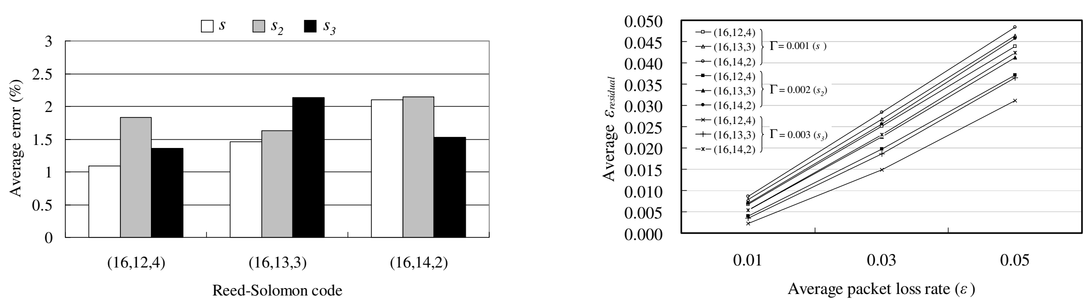
The resulting errors, shown in Figure 4(a), confirm the accuracy of the model. Most of the results are within two standard deviations of the mean, and the average error is never higher than 2.2%.

DOI: 10.1002/cpe



(a) (b) (c)

Figure 3. Verification of the analytic model for a (16,12,4) code: (a) $\Gamma = 0.001$ (s_1); (b) $\Gamma = 0.002$ (s_2); (c) $\Gamma = 0.003$ (s_3).



(a) (b)

Figure 4. (a) Average error between analytic and simulation results and (b) analytically determined values of $\varepsilon_{\text{residual}}$ for varying channel conditions.

5.2.2. *Analysis of error recovery performance by the existing scheme*

Figure 4(b) shows the performance of the current BCMCS error recovery scheme as predicted by our analytic model at pedestrian speeds, 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 most redundant (16,12,4) code. The situation is especially unfavorable for slow-moving mobiles (low values of Γ), indicating that the packet error process is strongly correlated, as shown in Table I. For a mobile speed of s_1, the recovery rate by the RS decoder is only 31.9% when the packet loss rate at the input to the decoder is 0.01. The situation becomes worse as the channel deteriorates, and the recovery rate finally drops to 12.5%, when the rate of packet loss is 0.05. However, the same rate of packet loss 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 inevitably reduces coverage if high data rates are used with the current BCMCS specification.

DOI: 10.1002/cpe

Table I. Error recovery rate (10 mobiles).

RS code	ε	Mobility (Γ)					
		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%
(16,13,3)*	0.01	77.4%	94.0%	98.7%	100%	100%	100%
	0.03	56.3%	76.3%	87.2%	99.4%	99.9%	100%
	0.05	45.4%	63.5%	75.2%	97.6%	99.4%	99.7%
(16,14,2)*	0.01	90.9%	97.5%	98.4%	99.9%	100%	100%
	0.03	75.3%	88.9%	93.9%	98.6%	99.7%	99.9%
	0.05	64.8%	78.6%	85.9%	96.6%	98.4%	99.3%

5.3. Comparative performance of our scheme

We now compare the performance of our scheme against pure RS, using the metrics of error recovery rate and video playback quality (PSNR [10]). PSNR is most commonly used to assess reconstruction quality in image compression, making it a measure of error recovery performance as seen at the application layer.

5.3.1. Error recovery rate

The average rate of residual packet loss decreases dramatically with our error recovery scheme. In particular, when a mobile moves at a pedestrian speed, the error recovery rate improves by as much as 46%, using a (16,13,3)* code, and by 59%, using (16,14,2)*, when compared with the original scheme using a (16,12,4) code; in both cases, $\varepsilon = 0.01$ and the average mobile speed is s_1. As the channel condition deteriorates and ε reaches 0.05, the average error recovery rate increases by as much as 33% with a (16,13,3)* code, or by 52% with a (16,14,2)* code, as shown in Table I.

With faster-moving mobiles, our scheme recovers more errors, although it is not such a significant improvement. For a mobile moving at vehicular speed (s_4), the improvement in error recovery rate is only 0.2% with a (16,14,2)* code, when the average packet loss rate is 0.01. The relative advantage of our scheme increases as the channel condition deteriorates. This performance improvement increases to 5% and 12% using (16,13,3)* and (16,14,2)* codes, respectively, under the conditions just described. However, if the mobile moves faster, the performance of the best pure RS code, which is (16,12,4), converges with the (16,13,3)* and (16,14,2)* codes.

5.3.2. Video playback quality (PSNR)

Figure 5 compares the average playback quality achieved by our error recovery scheme with that of the current RS scheme, for between 10 and 40 mobiles, with average values of ε of 0.01 and 0.03,

DOI: 10.1002/cpe

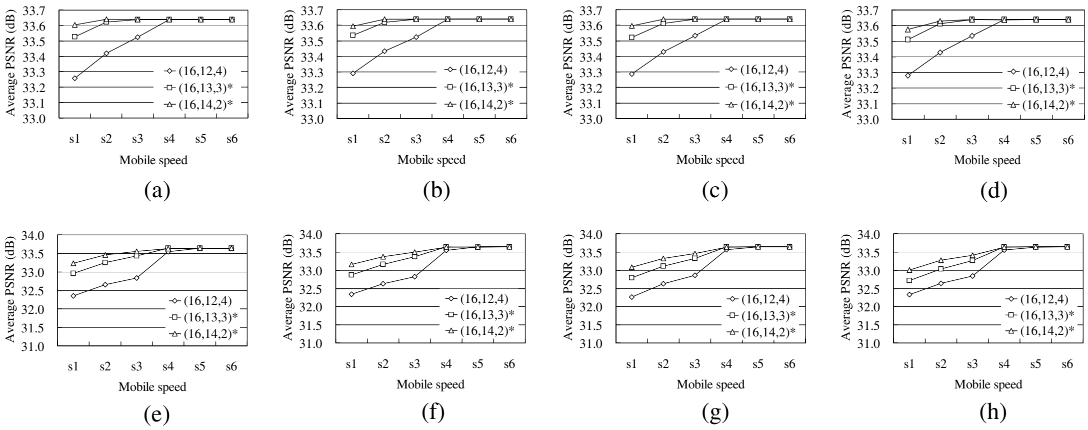


Figure 5. Comparison of playback quality: (a) 10 mobiles ($\varepsilon = 0.01$); (b) 20 mobiles ($\varepsilon = 0.01$); (c) 30 mobiles ($\varepsilon = 0.01$); (d) 40 mobiles ($\varepsilon = 0.01$); (e) 10 mobiles ($\varepsilon = 0.03$); (f) 20 mobiles ($\varepsilon = 0.03$); (g) 30 mobiles ($\varepsilon = 0.03$); (h) 40 mobiles ($\varepsilon = 0.03$).

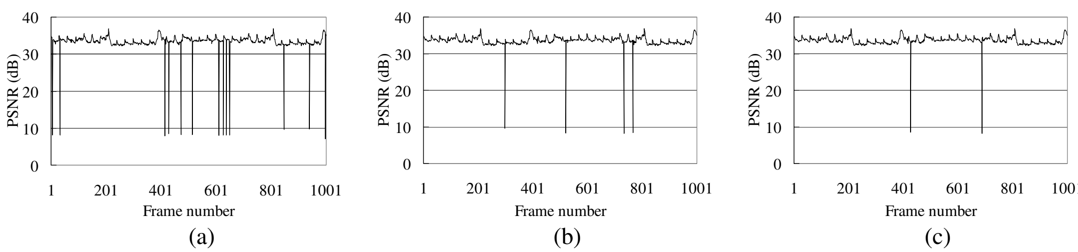


Figure 6. The PSNR fluctuation of a mobile when $\varepsilon = 0.01$ and $\Gamma = 0.001$: (a) (16,12,4); (b) (16,13,3)*; (c) (16,14,2)*.

and shows that the improvement in PSNR is more prominent for slower mobiles, as we would by now expect. The benefit increases as the packet loss rate of the channel deteriorates. Even with 40 subscribers, the new scheme gives better video playback. The gain in quality is more marginal for fast-moving mobiles, but it is still present. When the channel condition becomes very good, at $\varepsilon = 0.01$ for example, the two schemes achieve similar PSNR values, but ours remains ahead. A concrete example of the improvement that a user might expect is provided by Figure 6, which shows the fluctuations of PSNR for 1000 frames at a randomly chosen mobile, when NC = 40, with $\varepsilon = 0.01$ and $\Gamma = 0.001$. These graphs show that the PSNR fluctuates more widely when using the original (16,12,4) RS code than it does with the (16,13,3)* and (16,14,2)* codes.

DOI: 10.1002/cpe

6. CONCLUSIONS

By using RS and ARQ in concert, we can reduce the packet loss rate in the application layer. We cut the amount of parity information required for RS coding, and make flexible use of the slots we save to retransmit corrupted data. The key element in this approach is to give high priority to packets that belong to sub-blocks which can be recovered with a small number of retransmissions. Extensive simulation has shown that our scheme, incorporating a packet scheduler based on a utility function, is more efficient than the current scheme. This enables mobiles to receive high-data-rate broadcast services, and specifically video, under poor channel conditions where the packet loss rate is high.

REFERENCES

1. Wang J, Sinnarajaj R, Chen T, Wei Y, Tiedemann E. QUALCOMM. Broadcast and multicast services in cdma2000. *IEEE Communications Magazine* 2004; **42**(2):76–82.
2. Agashe P, Rezaiifar R, Bender P, QUALCOMM. cdma2000 high rate broadcast packet data air interface design. *IEEE Communications Magazine* 2004; **42**(2):83–89.
3. 3GPP2. *cdma2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification*, C.S0054 v1.0, 2004.
4. Bender P, Black P, Grob M, Padovani R, Sindhushayana N, Viterbi A. CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Communications Magazine* 2000; **38**(7):70–77.
5. Zorzi M, Rao RR, Milstein LB. Error statistics in data transmission over fading channels. *IEEE Transactions on Communications* 1998; **46**(11):1468–1477.
6. Cho Y, Kang K, Cho Y, Shin H. Improving MAC-layer error recovery for 3G cellular broadcasts. *Proceedings of the International Conference on Advanced Information Networking and Applications*. IEEE Computer Society Press: Washington, DC, 2006; 418–422.
7. Li W. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology* 2001; **11**(3):301–317.
8. Wu F, Li S, Zhang YQ. A framework for efficient progressive fine granularity scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology* 2001; **11**(3):332–344.
9. Pearmain A, Carvalho A, Hamosfakidis A, Cosmas J. The MoMuSys MPEG-4 mobile multimedia terminal. *Proceedings of the 3rd ACTS Mobile Summit*, June 1998; 224–229. Available at: http://cordis.europa.eu/infowin/acts/ienm/newsclips/arch1998/980201at.html.
10. Gonzalez RC, Woods RE. *Digital Image Processing*. Addison-Wesley: Reading, MA, 1992.

DOI: 10.1002/cpe