Improving MAC-layer Error Recovery for 3G Cellular Broadcasts

Yongwoo Cho, Kyungtae Kang, Yongjin Cho and Heonshik Shin
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea 151–744
{xtg05, ktkang, yongjin, shinhs}@cslab.snu.ac.kr

Abstract

The error-prone nature of the radio channel is the major challenge in servicing video streams over cdma2000 1xEV-DO broadcast networks. Forward error correction is commonly used when broadcasting a video stream over a lossy network: in BCMCS, the MAC protocol uses Reed-Solomon coding. We analyze the performance of Reed-Solomon error recovery with various parameter settings, and prove the correctness of our analysis through experiment. We also propose a Hybrid-ARQ scheme to improve error recovery capacity in the BCMCS environment. Simulation results show a significant reduction in packet loss while quantitatively validating our approach.

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. However, there has been no research on this topics in the context of cdma2000 1xEV-DO broadcast and multicast networks.

A mobile user of a wireless radio 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 shows that the resulting analytical model can accurately predict the packet error rate in the application.

In addition, we suggest an efficient error recovery scheme called on Hybrid-ARQ (Hybrid automatic retransmission request) to increase the performance of error recovery. Instead of adding bulky parity information to improve error recovery, we use an RS code with a low parity overhead, which can save time-slot resource. This reserved bandwidth is used to retransmit corrupted packets. The number of retransmissions is necessarily limited, and packets are selected to reduce the packet error rate as far as possible, by considering the arrangement of MAC packets in the RS error control block (ECB). Using a realistic simulation, we demonstrate the effectiveness of our proposed Hybrid-ARQ scheme in improving error recovery capacity in the application layer, and hence also the resulting video playback quality.

2 Hybrid-ARQ for error recovery

In contrast to the unicast cdma2000 1xEV-DO standard, error control in BCMCS is provided by forward error correction using RS encoding [3][4]. 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: a RS decoder can recover up to R octet erasures in each code word. 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 components of the broadcast MAC protocol are shown in Fig. 1. In current BCMCS, three RS codes are used to construct the ECB: (16, 12, 4), (16, 13, 3) and (16, 14, 2). The (16, 12, 4) code generates 16 code symbols for each block of 12 information symbols input to the encoder. The first 12 of the code symbols are information symbols and the remaining 4 symbols are parity data. Similarly, (16, 13, 3) and (16, 14, 2) codes generates 16 code symbols for each block of 13 or 14 information symbols input to the encoder, while parity data is accommodated in 3 symbols and

Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)
1550-445X/06 $20.00 © 2006 IEEE

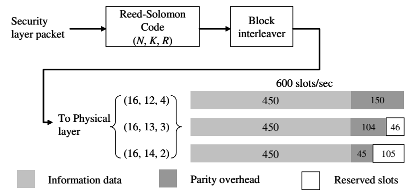


Figure 1. Broadcast MAC protocol components and RS overhead.

2 symbols respectively.

Fig. 1 shows the amount of data corresponding to information and parity during 1 second of transmission. The number of slots per second is 600 and the duration of each is therefore 1.67ms. 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. These leaves 46 slots unallocated using the (16, 13, 3) code, and 105 unallocated by the (16, 12, 4) code, as shown in Fig. 1. These saved slots can be used for retransmission of corrupted broadcast packets. 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 [5][6]. The ECB map shows how each mobile node selects target packets for retransmission so as to increase error recovery capacity while minimizing the number of packets to be retransmitted. Fig. 2 depicts the ECB map for two mobile nodes. In the middle of each of these example ECBs is an error cluster. When a mobile node enters an area with bad channel condition, packets are sequentially corrupted for a certain period. In the example in Fig. 2, the RS code is (16, 14, 2) and the value of M is 16; G stands for a security packet received successfully while B stands for a corrupted packet.

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 conceal 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 their priority.

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 (bold rectangles) in each sub-block is successfully retransmitted using the slots saved by a $(16, 14, 2)$ code. And the all 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

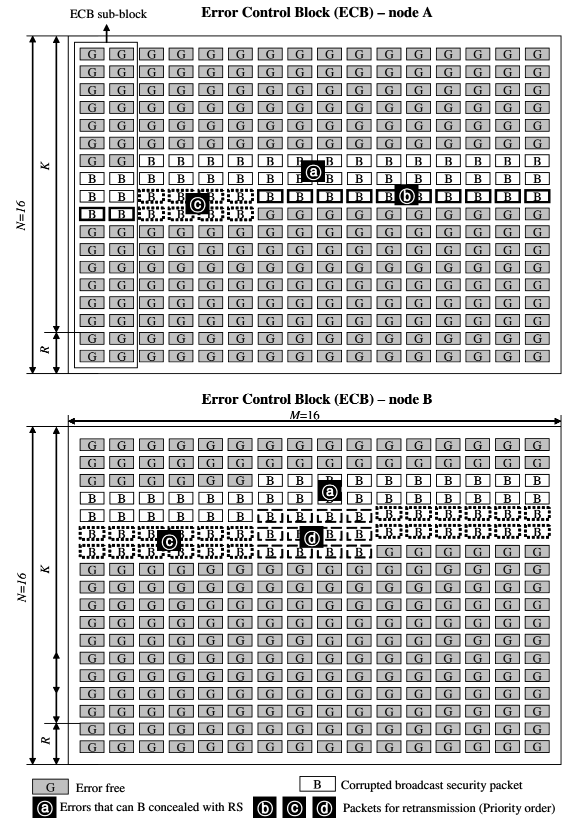


Figure 2. Proposed Hybrid-ARQ scheme.

(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).

3 Performance analysis of RS

3.1 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 [7] to simulate the error sequences generated by data transmission on a correlated Rayleigh fading channel. A first-order two-state Markov process can simulate the error sequences generated by data transmission on a correlated Rayleigh fading channel: these errors 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 $f_d N_{BL} T$ (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). 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 our experiments, we set $f_d N_{BL} T$ variously to 0.001, 0.002, 0.003, 0.01, 0.02 and 0.03. The block size N_{BL} is the packet length.

In the equations that are to follow, α is the probability that the i^{th} block of a packet is corrupted, given that the $(i-1)^{\text{th}}$ block is transmitted successfully, and β is the probability that the i^{th} block of a packet is successful, given that the $(i-1)^{\text{th}}$ block was unsuccessful. The calculation of these probabilities is described elsewhere [8].

3.2 Analysis of packet error rate using RS coding

First, we analyze the performance of the current BCMCS scheme, in its use of RS for error recovery. The structure of an error control block (ECB) when the RS code is (N, K, R) is shown in Fig. 3.

Our experiments explain the effects of changing the physical-layer packet error rate (denoted as $\varepsilon_{physical}$), which represents the probability that the packet contained in a certain slot is lost. We are using ε_{upper} to denote the upper-layer packet error-rate of data carrying (not parity carrying) after the corrupted packets are partly recovered, either by the current RS method or by the proposed Hybrid-ARQ error recovery scheme.

When a mobile node moves very slowly, we expect errors in clusters or bursts. The probability that the burst length is κ can be written as:

$$P_{burst}(\kappa) = (1-\beta)^{\kappa-1}\beta. \tag{1}$$

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. 3:

$$P_{RS}(failure) = P_{case1} + P_{case2} + P_{case3} + P_{case4}. \tag{2}$$

In the first case, transmission of an initial sequence of packets in the current ECB fails due to 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

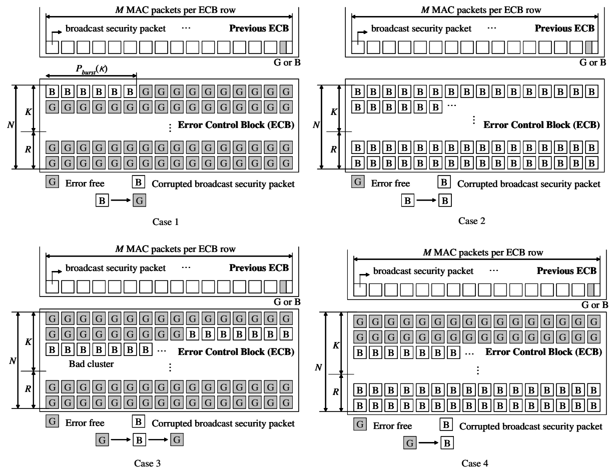


Figure 3. Four cases in which an (N, K, R) RS code cannot recover the lost packets.

assume that the state never reverts to bad 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 state 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 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:

$$\begin{aligned}
P_{case1} &= \Delta_{1,2} \times \alpha + \varepsilon(1-\beta)) \\
&\quad \times \sum_{\kappa=RM+1}^{NM-1} P_{burst}(\kappa)(1-\alpha)^{NM-\kappa-1}, \\
P_{case2} &= \Delta_{1,2} \times \alpha + \varepsilon(1-\beta))(1-\beta)^{NM-1}, \\
P_{case3} &= \Delta_{1,2} \times \sum_{\lambda=1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \\
&\quad \times \sum_{\kappa=RM+1}^{NM-\lambda-1} P_{burst}(\kappa)(1-\alpha)^{NM-\kappa-\lambda-1}, \\
P_{case4} &= \Delta_{1,2} \times \Omega_{temp},
\end{aligned}$$

where

$$\Omega_{temp} = \sum_{\kappa=RM+1}^{NM-1} (1-\alpha)^{NM-\kappa-1}\alpha(1-\beta)^{\kappa-1}. \tag{3}$$

$\Delta_{1,2}$ represents the probability that the first packet of an ECB is corrupted if the last packet of the previous ECB was not corrupted; and $\Delta_{3,4}$ is the probability that the first packet of the new ECB is corrupted when the last packet of the previous ECB was not. These probabilities are given as follows:

$$\Delta_{1,2} = ((1-\varepsilon)\alpha + \varepsilon(1-\beta)),$$
$$\Delta_{3,4} = ((1-\varepsilon)(1-\alpha) + \varepsilon\beta). \quad (4)$$

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 of Eq. 3, as follows:

$$\begin{aligned} E_{case1} &= \Delta_{1,2} \times \Phi_{case1}, \text{ where} \\ \Phi_{case1} &= \sum_{\kappa=RM+1}^{(R+1)M-1} P_{burst}(\kappa)(1-\alpha)^{NM-\kappa-1}\Theta \\ &+ \sum_{\kappa=(R+1)M}^{NM-1} P_{burst}(\kappa)(1-\alpha)^{NM-\kappa-1}\kappa. \\ E_{case2} &= \Delta_{1,2} \times (1-\beta)^{NM-1}. \\ E_{case3} &= \Delta_{3,4} \times \sum_{\lambda=1}^{(N-R-1)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi_{case3} \\ &+ \Delta_{3,4} \times \sum_{\lambda=(N-R-1)M-1}^{(N-R)M-2} (1-\alpha)^{\lambda-1}\alpha \times \Phi'_{case3}, \end{aligned}$$

where

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

and

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

$$E_{case4} = \Delta_{3,4} \times \Phi_{case4},$$

where

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

The variable Θ is κ modulo M. Finally, as the physical-slot error rate ($\varepsilon_{physical}$) changes, the error rate for transmitting packets using the RS scheme can be expressed as

$$\varepsilon_{upper} = (E_{case1} + E_{case2} + E_{case3} + E_{case4})/NM. \quad (5)$$

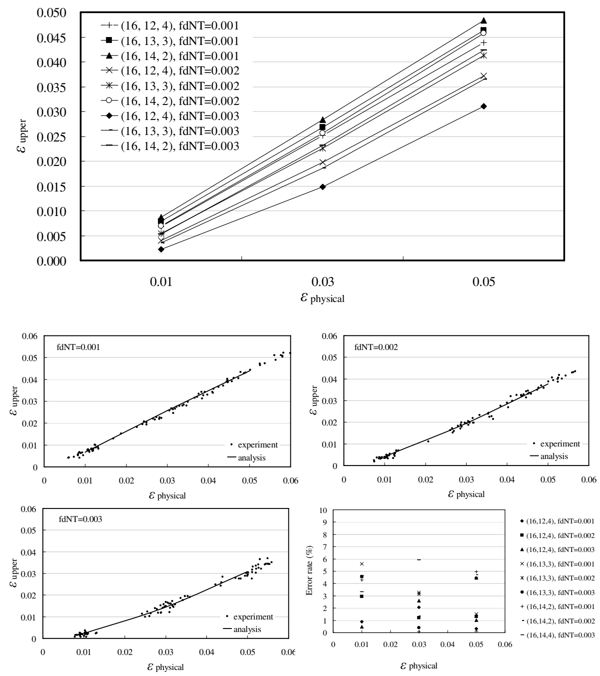


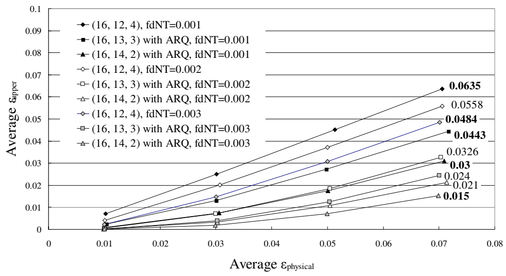
Figure 4. Packet error rate in the upper layer with varying channel condition and error rate: experimental and analytic results.

4 Performance evaluation

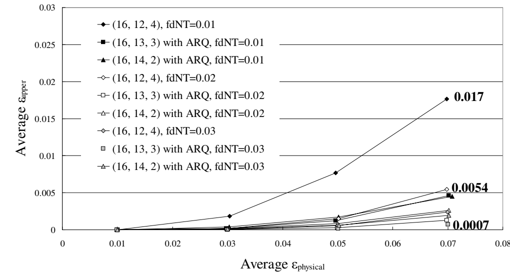
We used the four Foreman test bench video sequences streamed at 30 frames per second, with a total of 10,000 frames. The number of subscribers to each video sequence is uniformly distributed, and the average physical-layer packet error rates of all mobile nodes varies from 1% to 5%. In these experiments, we used QPSK modulation with a 1228.8 kbps data-rate forward channel, and thus the value of N_{BL} is 256 bytes. We compared our Hybrid-ARQ scheme with the original RS-based scheme employed in BCMCS. In our comparison, the RS codes were used with 16 MAC packets per ECB row ($M = 16$).

Fig. 4 shows the performance of the current BCMCS error recovery scheme with a varying average channel condition, and 10 mobile nodes. The results that we present were obtained by averaging 100,000 trials. As Fig. 4 shows, the error capacity of RS declines suddenly for slow-moving nodes (low values of $f_d N_{BL} T$). In slow-moving conditions, the errors occurs are so bursty that a RS ECB does not provide sufficient interleaving, even with a $(16, 12, 4)$ code.

The three graphs in the middle of Fig. 4 show experimental and analytic error rates for a $(16, 12, 4)$ RS code



(a) $f_d N_{BL} T = 0.001, 0.002$ and 0.003.



(b) $f_d N_{BL} T = 0.01, 0.02$ and 0.03.

Figure 5. Average ε_{upper} as $\varepsilon_{physical}$ varies.

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 experimental results are clustered closely around the line corresponding to the analytical model, confirming the accuracy of our analytic model. Fig. 4 shows that the error rate rarely rises above 5% in these experiments.

Next, we compared the error recovery capacity of our proposed Hybrid-ARQ scheme ($(16, 13, 3)$ and $(16, 14, 2)$, both with ARQ) with the original scheme ($(16, 12, 4)$ without ARQ). The experimental results shown in Fig. 5 were obtained by averaging the values of ε_{upper} for 10 mobile nodes. When a mobile node moves very slowly, as depicted in Fig. 5(a), the average value of ε_{upper} is reduced from 0.063 to 0.044, using a $(16, 13, 3)$ code, and to 0.03, using a (16, 14, 2) code, together with Hybrid-ARQ in both cases, while $\varepsilon_{physical} = 0.07$ and $f_d N_{BL} T = 0.001$. When $\varepsilon_{physical} = 0.07$ and $f_d N_{BL} T = 0.003$, the average ε_{upper} is reduced from 0.048 to 0.015, using a (16, 14, 2) code with ARQ. These results show that, if we reserve slots by reducing the coding overhead of RS, and use those slots flexibly for retransmission, the overall error recovery 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. 5(b). When $\varepsilon_{physical} = 0.07$ and $f_d N_{BL} T = 0.01$, ε_{upper} is reduced from 0.017 to 0.0054 by using a (16, 14, 2) code with ARQ. Similarly, ε_{upper} drops by as much as 0.0007 with Hybrid-ARQ when $\varepsilon_{physical} = 0.07$ and $f_d N_{BL} T = 0.03$. These results show the effectiveness of our proposed Hybrid-ARQ scheme.

5 Concluding remarks

We have analyzed 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. Their close agreement suggests that our analytic model is accurate. Additionally, we have proposed the Hybrid-ARQ scheme to conceal errors. By using RS and ARQ in concert, we can reduce the packet error rate in the application layer. 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 can be recovered with a small number of retransmissions. Extensive simulation has shown that Hybrid-ARQ is more efficient than the current scheme.

References

[1] J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, E. Tiedeman, and QUALCOMM. Broadcast and multicast services in cdma2000. *IEEE Communications Magazine*, 45(2):76–83, February 2004.

[2] 3GPP2. Broadcast-multicast service (bcmcs) framework draft document. Technical Report X.P0019 v 0.1.3, 3GPP2, August 2003.

[3] 3GPP2. cdma2000 high rate broadcast-multicast packet data air interface specification. Technical Report C.S0054 v1.0, 3GPP2, March 2004.

[4] P. Agashe, R. Rezaiifar, P. Bender, and QUALCOMM. cdma2000 high rate broadcast packet data air interface design. *IEEE Communications Magazine*, 42(2):83–89, February 2004.

[5] 3GPP2. cdma2000 high rate packet data air interface specification. Technical Report 3GPP2 C.S0024 v3.0, 3GPP2, December 2001.

[6] 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):70–77, July 2000.

[7] M. Zorzi, R. R. Rao, and L. B. Milstein. Error statistics in data transmission over fading channels. *IEEE Transactions on Communications*, 46(11):1468–1477, November 1998.

[8] K. Kang, J. Cho, and H. Shin. Dynamic packet scheduling for cdma2000 1xev-do broadcast/multicast services. *Proc. IEEE Wireless Communications and Networking Conference*, 4, 2005.