Proactive Reed-Solomon Bypass (PRSB): A Technique for Real-Time Multimedia Processing in 3G Cellular Broadcast Networks

Yongwoo Cho, Kyungtae Kang, Joonho Lee and Heonshik Shin
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea 151–744
{xtg05, ktkang, jlee108, shinhs}@cslab.snu.ac.kr

Abstract

We analyze the execution time of MAC-layer Reed-Solomon error recovery in cdma2000 1xEV-DO broadcast and multicast services (BCMCS), with respect to cache size and air channel condition. We observe that a deteriorating channel condition often causes an intolerable delay for real-time tasks because of the error recovery process, even when sufficient cache memory is available to increase CPU throughput. To overcome this problem, we propose proactive Reed-Solomon bypass (PRSB), which determines whether the Reed-Solomon scheme should be used to correct errors in the real-time communication environment. The proposed scheme utilizes the cache effectively, thus preventing reset of the entire decoding process and allowing real-time multimedia applications to meet their deadline constraints. Simulation results show that the PRSB greatly improves the overall playback quality of a video.

1 Introduction

Broadcast and multicast services (BCMCS) are currently being standardized by various mobile wireless standard bodies such as 3GPP and 3GPP2 [8]. These services will allow users simultaneously to receive a variety of content on their handsets over cellular links.

A wireless radio channel has a much higher error-rate than that expected on a wired link. Additionally, errors occur in clusters or bursts, with relatively long error-free intervals between them. Error control is therefore mandatory when data are sent over wireless access networks to an end-user. Forward error correction is the scheme most commonly used when sending a broadcast video stream over a lossy network such as a wireless radio channel. In BCMCS, Reed-Solomon (RS) coding is used as a method of forward error correction which can efficiently conceal error clusters [1][2]. When utilizing RS, the timing constraints which are characteristic of multimedia streams must also be considered. A video frame is useful only if it arrives at the mobile node before its playback time. However, the current cdma2000 1xEV-DO chipset used for unicast transmission is not designed to minimize delay, and timing problems are therefore inevitable.

In this study, we analyze the execution time of an RS forward error correction (FEC) scheme and propose an efficient way of satisfying timing constraints using a cache. Additionally, we suggest a lower bound on the channel condition which will satisfy the timing constraints of real-time multimedia applications, based on the observation that latency increases as the condition of a channel deteriorates. We also propose a proactive Reed-Solomon bypass (PRSB) scheme to prevent significant quality degradation due to buffer corruption when a long delay occurs while using an RS scheme to decode video data.

The remainder of this paper is organized as follows: in Section 2, we introduce the background to our research and, in Section 3, we explain the role of RS in the current chipset environment. The improvement of RS performance by means of a cache is explained in Section 4 and our proposed PRSB scheme is discussed in Section 5. Section 6 concludes the paper.

2 Background

2.1 FEC for 3G cellular broadcast networks

Unlike the unicast cdma2000 1xEV-DO standard, broadcast services do not use an error control scheme based on automatic repeat requests (ARQs). Instead, error control is provided by FEC using RS encoding, which we will review briefly in this section.

RS is an algebraic code belonging to the class of Bose-Chaudry-Hocquehen multiple-burst-correcting cyclic codes that operates on bytes of fixed length. This is a large class of powerful random error-correcting cyclic codes.

If the parity part of a sequence is of length R bytes, RS codes allow the correction of up to R corrupted bytes, if their positions are known, or up to $R/2$ bytes if the position of the errors is unknown. The RS code in cdma2000 1xEV-DO uses 8-bit bytes, and a number of parity bytes. The maximum sequence length that can be generated is 255 bytes, but in practice shorter sequences are used.

An understanding of the encoding part of the RS algorithm is not necessary here because we are going to focus on the evaluation of receiver performance; but decoding will be described briefly. A RS decoder generates four syndrome bytes, which will be all zero if the message has no errors. These syndromes can be computed relatively simply, as follows [3]: Let $\nu(x) = \nu_0 + \nu_1 x + \nu_2 x^2 + \ldots v_{n-1}x^{n-1}$ be the transmitted code vector and let $\gamma(x) = \gamma_0 + \gamma_1 x + \gamma_2 x^2 \ldots + \gamma_{n-1}x^{n-1}$ be the corresponding received vector. Then $e(x) = \gamma(x) - \nu(x) = e_0 + e_1 x + e_2 x^2 \ldots + e_{n-1}x^{n-1}$ is the error pattern added by the channel, where $e_i = \gamma_i - \nu_i$ is a symbol from GF^{2^m}, where GF stands for Galois Field, which is a field of finite order; and so GF^{2^m} is a Galois Field of order 2^m. If we assume that there are ν errors in positions $i_1, i_2, \ldots, i_\nu$, then $e(x) = e_{i_1}x^{i_1} + e_{i_2}x^{i_2} + \ldots + e_{i_\nu}x^{i^\nu}$. If t is the error correcting capability of the code, then $2t$ syndromes can be computed, as follows:

$$\begin{aligned} S_j &= \gamma(\alpha^j) = e_{i_1}(\alpha^j)^{i_1} + e_{i_2}(\alpha^j)^{i_2} + \ldots + e_{i_\nu}(\alpha^j)^{i_\nu} \\ &= e_{i_1}(\alpha^{i_1})^j + e_{i_2}(\alpha^{i_2})^j + \ldots + e_{i_\nu}(\alpha^{i_\nu})^j \\ &= e_{i_1}X_1^j + e_{i_2}X_2^j + \ldots + e_{i_\nu}X_\nu^j, \\ &\quad \text{where } X_\iota \text{ is defined as } \alpha^{i_\iota} (\iota = 1, 2 \ldots \nu). \end{aligned} \tag{1}$$

These syndrome equations can be obtained as a sequence of $2t$ algebraic equations, and they can be translated into a series of linear equations by defining the error locator polynomial $\Lambda(x)$ as follows:

$$\Lambda(x) = \prod_{\iota=1}^{\nu}(1 - X_\iota x) = \Lambda_\nu x^\nu + \Lambda_{\nu-1}x^{\nu-1} + \ldots + \Lambda_1 x + 1. \tag{2}$$

If it is assumed that t errors ($\nu = t$) have occurred, then we obtain the following matrix equation:

$$\begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_t \\ S_2 & S_3 & S_4 & \cdots & S_{t+1} \\ S_3 & S_4 & S_5 & \cdots & s_{t+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_t & S_{t+1} & S_{t+2} & \cdots & S_{2t-1} \end{bmatrix} \begin{bmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \Lambda_{t-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_{t+1} \\ -S_{t+2} \\ -S_{t+3} \\ \vdots \\ -S_{2t} \end{bmatrix}. \tag{3}$$

Once the syndromes have been computed, there are a number of ways to find the error locations. In our RS code, we use Berlekamp's iterative algorithm which takes the following form [3]:

1. Compute the syndrome sequence $S_1, \ldots, S_{2t}$ for the received word.
2. Initialize the algorithm variables:
$$\kappa = 0, \Lambda^{(0)}(x) = 1, L = 0 \text{ and } T(x) = x.$$
3. Set $k = k + 1$ and then compute the discrepancy $\Delta^{(k)}$ as follows:
$$\Delta^{(k)} = S_k - \sum_{i=1}^{L} \Lambda_i^{(k-1)} S_{k-1}.$$
4. If $\Delta^{(k)} = 0$, then go to Step 8.
5. Modify the connection polynomial:
$$\Lambda^{(k)}(x) = \Lambda^{(k-1)}(x) - \Delta^{(k)}T(x).$$
6. If $2L \geq k$, then go to Step 8.
7. Set $L = k - L$ and $T(x) = \Lambda^{(k-1)}(x)/\Delta^{(k)}$.
8. Set $T(x) = x \cdot T(x)$.
9. If $k < 2t$, then go to Step 3.
10. Determine the roots of $\Lambda(x) = \Lambda^{(2t)}(x)$. If the roots are distinct and lie in the right field, then determine the error magnitudes, correct the corresponding locations in the received word, and stop.
11. Otherwise, declare a decoding failure and stop.

Once errors have been detected, they must be deleted and the original data recovered. For erasure decoding, assuming a received vector with ν errors and f erasures, an erasure locator polynomial is defined as follows:

$$\Gamma(x) = \prod_{\iota=1}^{f}(1 - Y_\iota x), \text{ where } Y_\phi = \alpha^{j_\phi}, (\phi = 1, 2 \ldots, f). \tag{4}$$

Finally, decoding is achieved by the following steps:

1. Compute an erasure polynomial $\Gamma(x)$ using the erasure information provided by the receiver.
2. Replace the erased coordinates with zeros and compute the syndrome.
3. Compute the modified syndrome polynomial:
$$\Phi(x) = (\Gamma(x)[1 + S(x)] - 1) \bmod x^{2t+1}.$$
4. Apply the Berlekamp algorithm to find the connection polynomial $\Lambda(x)$, using the modified syndrome coefficients Φ_i, $i = f + 1, \ldots, 2t$.
5. Find the roots of $\Lambda(x)$, and thus the error locations.

6. Determine the magnitudes of the errors and erasures. First it is necessary to define the error/erasure locator polynomial as follows:

$$\Psi(x) = \Lambda(x)\Gamma(x).$$

Then the error and erasure values are

$$e_{i_k} = \frac{-X_k\Omega(X_k^{-1})}{\Psi'(X_k^{-1})}, f_{i_k} = \frac{-Y_k\Omega(Y_k^{-1})}{\Psi'(Y_k^{-1})},$$
$$\text{where } \Lambda(x)[1+\Phi(x)] = \Omega(x) \bmod x^{2t+1},$$
$$X_\phi = \alpha^{i_\phi}, (\phi = 1, 2 \ldots, \nu)\,. \tag{5}$$

As shown above, the RS code consists of two major parts, error detection and error recovery. Error detection is always performed and does not require much computation. Error recovery is only performed when there are errors, and requires complicated and time-consuming computation. Thus, for time-critical applications, it is reasonable to skip error recovery if the errors are not critical.

2.2 Real-time constraints of MPEG video streams

MPEG is one of the best-known multimedia formats that can utilize cdma2000 1xEV-DO BCMCS. The transport stream system target decoder (T-STD) model of an MPEG system provides a guideline for managing timing constraints and buffers in an MPEG transport stream (TS) decoding process [5][6]. There are three types of decoders in the T-STD: video, audio, and systems. For proper decoding of an MPEG TS, the system requires a transport buffer (TB) to store the de-multiplexed packetized elementary stream (PES), and a main buffer (B) to feed appropriate access units into each decoder. These buffers are required to provide exact timing references to synchronize the system target decoding processes, which include de-multiplexing of each elementary stream (ES) received from the transport stream (TS), decoding of the de-multiplexed ES, and presentation of decoded video and audio. By using the same $system\_clock\_frequency$ (SCF), these three decoders (system, video and audio) can be operated synchronously and will have a presentation unit available at the correct time. A SCF is transmitted to the system target decoder (STD) by means of the program clock reference (PCR), which is prepared during the system encoding process. The PCR then generates a decoding time stamp (DTS) and a presentation time stamp (PTS) respectively. Fig.1 shows how the PCR, DTS and PTS contribute to the timing of TS decoding.

The timing of each process in the system decoder is determined by the dynamic PCR and the static $transport\_rate$. If $PCR(i)$, which is an input of the system decoder, has been corrupted, then the SCF will also be corrupt, as follows:

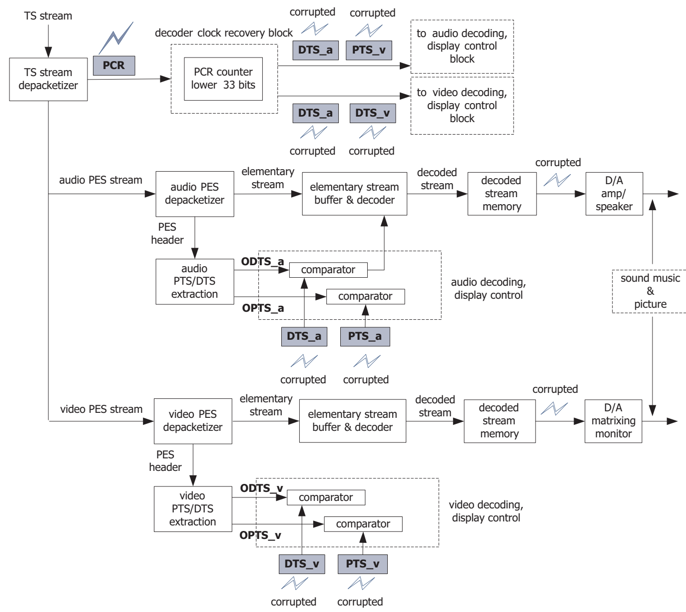


Figure 1. Timing model of TS decoding.

$$SCF_{corrupted} = \frac{PCR(i)}{t_{corrupted}(i) - \frac{i-i''}{transport\_rate}}. \tag{6}$$

This corruption of the SCF will sequentially corrupt all the timing references, causing a system crisis, because all the rates in the system decoder will be modified. Once a system crisis has occurred, the system decoder will stop all the decoding processes, initialize every buffer and timing count, and then obtain a fresh PCR from which to calculate the SCF, allowing the system decoder to be re-initialized.

These problems can be avoided if the timing follows MPEG rules. According to the MPEG specifications [5][6], the TS should be constructed so that the time interval between the bytes containing the last bit of the PCR's base field in successive occurrences of the PCR in the transport stream packets containing the PCR packet identifier (PCR_PID) for each program is no more than 100ms:

$$|t(i) - t(i')| \le 100 \text{ ms}, \tag{7}$$

for all i and i', where i and i' are the indexes of the bytes containing the last bit of consecutive $program\_clock\_reference\_base$ fields in the transport stream packets of the PCR_PID for each program. There should be at least two PCRs, from the specified PCR_PID within a transport stream, between consecutive PCR discontinuities to facilitate phase locking and to permit extrapolation of byte delivery times. Therefore, in an MPEG system the delay should not exceed 100ms, from arrival to system decoder.

3 RS in the Current Environment

QUALCOMM has developed the MSM5000 Mobile Station Modem™ (MSM) chipset and system software solution to support field trials and the first commercial deployments of 3G CDMA 1x networks. The MSM5000 device integrates functions that support a dual-mode CDMA/FM subscriber unit. Subsystems within the MSM5000 device include a CDMA processor, a digital FM (DFM) processor, a QUALCOMM-designed DSP for voice compression, an ARM ARM7TDMI microprocessor. There are also several peripheral interfaces that are used to support other functions, demonstrating that the current chipset can be connected to peripheral devices such as RAM without a cache memory. RS decoding can be executed on the ARM7TDMI core with some external memory using the current MSM5000 chipset.

3.1 Experimental environment

We now analyze how the execution time of RS decoding, with no cache, depends on the channel condition. We generated a template data-stream with errors that model those of an actual air channel, by injecting errors generated by a channel error model. In this study, we used the simple threshold model suggested by Zorzi [9] to simulate the behavior of data errors which arise in transmission over fading channels. This model can be represented as a binary Markov process in which the receiver is deemed to have received a data bit when the fading envelope of that bit is more than some threshold value. If the fading envelope is below the threshold, receipt fails. 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. Fading in the air channel is assumed to have a Rayleigh distribution. By choosing different values for the physical-layer bit error-rate ($\varepsilon_{physical}$) and for f_dT (the Doppler frequency normalized to the data-rate), we can model different degrees of correlation in the fading process of radio channels. The value of f_dT determines the correlation properties, which are related to the mobile speed for a given carrier frequency. When f_dT 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 value of f_dT (fast fading). In these experiments, we set the value of f_dT to 0.00002, which represents moderate fading.

The performance of RS was measured using the SEE (SNU Energy Explorer)[7]. In our simulation, the ARM7TDMI core and the SEC 128 Mbit SDRAM array (K4S280832A) were operated at 200 MHz and 100 MHz respectively. The target application is assumed to be a multimedia service and the proportion of CPU time used by RS decoding is assumed to be 5%. We used a bit-rate of 294 kbps for processing the multimedia stream.

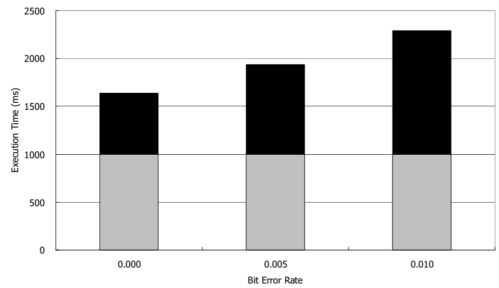


Figure 2. Execution time of RS (no cache).

3.2 Execution time of RS without a cache

Fig. 2 shows the performance of RS decoding with varying bit error rates. In the figure, grey and black areas indicate the timing deadline of the upper-layer service and the excess time required for RS decoding, respectively. In this environment, we see that RS decoding is not possible in real-time. However, a system with a multimedia processor could adjust the CPU reservation time for RS decoding. A slight performance gain could also be achieved by optimizing the RS decoding algorithm, although RS decoding will always be a time-consuming task.

4 Improving the Performance of RS

4.1 Proposed system architecture

For time-constrained real-time applications such as MPEG video, the latency caused by accessing a buffer is a critical issue in service quality, as already mentioned. That

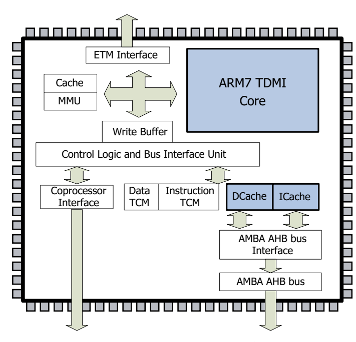
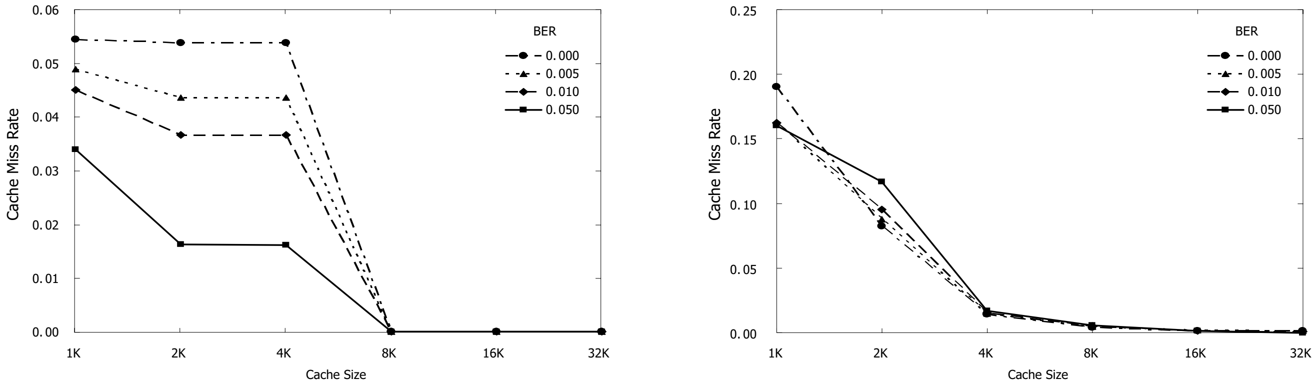


Figure 3. Example chip solution with a cache.



(a) Icache (b) Dcache

Figure 5. Cache miss rate against cache size.

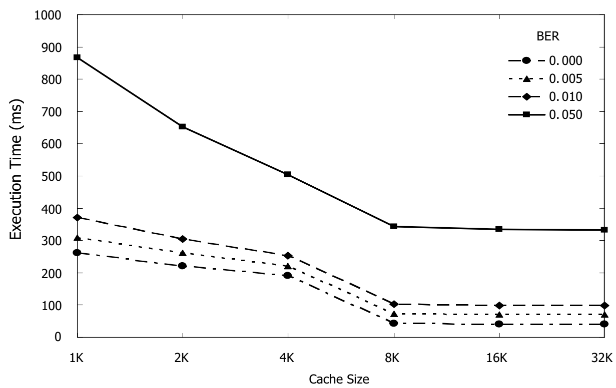


Figure 4. Execution time against cache size.

is why we have analyzed the performance of error recovery with the non-caching structure that is most widely used in current cdma2000 1xEV-DO chip solutions such as the MSM5000. However, this chip structure cannot satisfy the timing constraints of MPEG video because execution times extend during error recovery by the RS decoder. The current chip solution is tuned for unicast services which do not require FEC, and the chipset design does not allow for the constraints on spatial and temporal locality of data and instructions which occur during RS decoding. Current cdma2000 1xEV-DO unicast services exploit ARQs to correct errors.

We have analyzed execution times for the current chip architecture with the addition of a cache. Fig. 3 shows an example system, designed for MPEG broadcasting, which incorporates an ARM7TDMI core and a cache.

4.2 Performance evaluation

We evaluated the execution time taken by the RS decoding routine. The channel error model suggested by Zorzi, described in Section III, is also used in this experiment. As already mentioned, the execution time grows as the bit error rate (BER) increases. This is due to operations that involve extensive calculation, such as the Berlekamp algorithm in the RS code. We have already explained that the RS algorithm needs to perform complicated matrix computations for error recovery. These will increase the load on the CPU in general; in addition, these computations are very much reliant on spatial and temporal locality. When calculating matrix multiplicity, for example, it is possible to reuse previous values, and so the execution time will be reduced if the matrix is in the cache.

We simulated the proposed system architecture and measured the miss ratio of each cache and the execution times, while varying the sizes of both the instruction and data cache between 2k and 64k. Fig. 4 shows that cache size and execution time are inversely proportional up to a cache size of 8k, but after that there is negligible improvement. Even with a cache, a higher BER implies more computation time. For example, with a cache size of 8k and a BER of 0.05, decoding requires about 350ms; when the BER is 0.005, it requires less than 100ms. This is because a higher BER increases the requirement for error recovery, which is a bottleneck in the computation.

Next, we looked at the cache miss rate for different sizes of both the instruction and data cache. The simulation results for the instruction cache (Icache) are shown in Fig. 5(a). We see that the miss-rate decreases as the cache size increases, and almost reaches zero when the cache is larger than 8k. This indicates that an 8k cache is big enough to load all the code for the RS calculation. The performance of the data cache (Dcache) (Fig. 5(b)) shows a similar pattern. Again the miss-rate is negligible beyond 8k. We conclude that both the instruction and data cache only need to be 8k-16k for the RS algorithm.

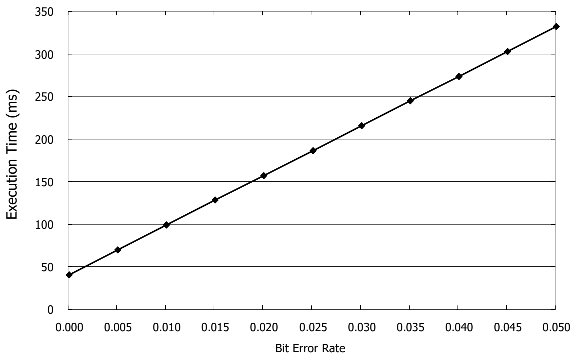


Figure 6. Execution time against BER with a 32KB L1 cache.

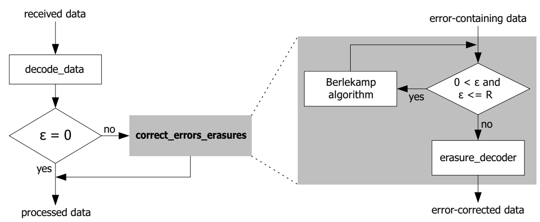


Figure 7. Flow diagram of RS decoding.

5 Proactive Reed-Solomon Bypass Scheme

To investigate the relation between excessive system execution time and BER more deeply, the system performance for different BERs was measured with an L1 cache of 32KB. The experimental environment was unchanged and the CPU reservation time remained at 5%. As shown in Fig. 6, execution time increases with BER. This can be explained by reference to the characteristics of RS decoding and the effect of cache miss-rate, as described in the previous section. The RS decoder operates as shown in Fig. 7. When a data-stream is received, the number of syndrome bytes created is proportional to the quantity of input data, regardless of its quality (Eq. 1). Thus, the time required to generate a syndrome depends only on the quantity of data, and will be stable for a data-stream with a static bit-rate. However, if there is an error in the data-stream, additional work is required for error detection, location and correction. As mentioned earlier, these errors can be found using Berlekamp's algorithm and the data recovered by erasure decoding. The number of times that these procedures need to be executed is proportional to the number of errors, and so the time required for RS decoding is proportional to the BER.

The resulting unpredictable performance has a negative effect on many real-time applications. A particular drawback in an MPEG system is the limit of 100ms on the allowable PCR interval (see Eq. 7). We therefore propose a proactive Reed-Solomon bypass (PRSB) scheme to deal with the unpredictable performance of the RS decoder. This scheme is based on the idea of anticipating whether an RS operation can be completed within the deadline, using the linear relationship between BER and execution time. PRSB then omits RS as often as is required to satisfy the system timing.

We conducted an experiment to explore the trade-off between system timing and recovered BER in a multimedia service. RS decoding continues to use the same BER template during a BER fluctuation. Using the SEE, we measured the timing information and the error dump which will be provided to the MPEG decoder. We recorded the peak signal to noise ratio (PSNR) [4] of the MPEG system with and without PRSB. The results are shown in Fig 8.

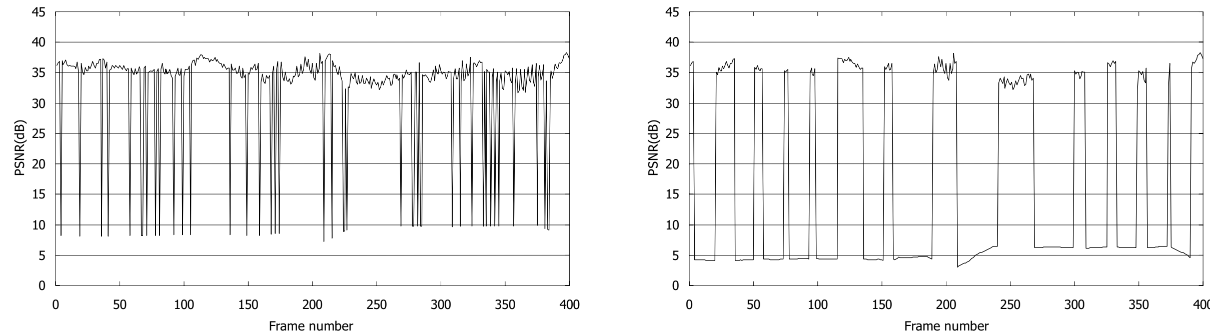
We see that the average PSNR is 16.01 dB without PRSB and 32.40 dB with PRSB. Using PRSB, the received BER is the same as the BER of the input because RS decoding is bypassed when a BER exceeds the threshold. Without RS decoding, the BER increases temporarily and the decoder tries to recover the remaining errors using error resilience features. The PSNR decreases dramatically due to the lower efficiency of error resilience compared with RS; but without PRSB the problem would be even worse.

As mentioned in Section 2, the SCF becomes damaged if the PCR does not arrive on time because it was unable to satisfy the input interval limitation of the MPEG system decoder. Once the SCF becomes damaged, it will successively damage the DTS and PTS, and eventually the synchronization of data flow to each buffer will be broken. Then the entire system needs to be re-initialized. Every buffer will be emptied and the clock reset. The buffers will then be filled by a new stream and the PCR will be re-calculated by parsing this stream. The system will continually perform clock feedback to find the SCF. Once this has been recovered, all decoding processes will be restarted by re-synchronizing to the new SCF. As a result of this procedure, a significant delay in stabilization and the PSNR drops, as shown in Fig. 8(b). But the loss in BER that results from using PRSB is less than the loss that will occur due to the system initialization that would otherwise be necessary.

6 Conclusion

In BCMCS, FEC is used as the error correction algorithm for broadcasts, instead of ARQ. We have developed a realistic environment to measure the performance of the RS algorithm which is used for FEC in cdma2000 1xEV-DO. Our RS decoding simulation, conducted under a similar environment to QUALCOMM's MSM5000, shows that the performance of RS decoding is closely related to cache size, and an appropriate cache size has been suggested.

Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)
1533-2306/05 $20.00 © 2005 IEEE



(a) Using the PRSB scheme (BER of 5%)   (b) Without the PRSB scheme (BER of 5%)

Figure 8. PSNR of template.

If a packet delivery is delayed, the entire system has to be re-initialized, which dramatically degrades the quality of service. Therefore, it is critical to synchronize the timing in multimedia applications that use cdma2000 1xEV-DO BCMCS. The PRSB scheme does this by skipping RS decoding for packets whose BER is higher than a threshold. PRSB thus avoids periodic resetting of the entire decoding process, while permitting the quality of the multimedia service to degrade temporarily as a result of the reduced error recovery rate. Extensive simulation results show that the average playback quality is improved by using the PRSB scheme.

It seems certain that hardware for RS decoding of digital media process will eventually appear if cdma2000 1xEV-DO BCMCS are successful. But we expect that PRSB will have a major role as a coordinator of the RS process until such hardware is released.

Acknowledgment

We would like to thank Yongsoo Joo and Hojun Shim of the Computer Systems Laboratory in Seoul National University for helping us to simulate performance evaluations on the SNU Energy Explorer.

References

[1] 3GPP2. cdma2000 high rate broadcast-multicast packet data air interface specification. Standard C.S0054 v1.0, 3GPP2, 2004.

[2] 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.

[3] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

[4] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[5] ISO/IEC. Information technology – generic coding of moving pictures and associated audio information. Standard ISO/IEC 13818, International Standards Organization, 2000.

[6] ISO/IEC. Information technology – coding of audio-visual objects. Standard ISO/IEC 14496, International Standards Organization, 2004.

[7] I. Lee, Y. Choi, Y. Cho, Y. Joo, H. Lim, H. G. Lee, H. Shim, and N. Chang. On the statistics of block errors in bursty channels. *IEEE Design and Test of Computers*, 21(6):572–586, November-December 2004.

[8] 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.

[9] 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 2004.