This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Seamless Multimedia Broadcasting over cdma2000 BCMCS Networks

Yongwoo Cho, Kyungtae Kang and Heonshik Shin
School of Electrical Engineering and Computer Sciences
Seoul National University, Seoul, Korea 151–744
Email: {xtg05, ktkang, shinhs}@cslab.snu.ac.kr
Telephone: (822) 880–7298

***Abstract*— Recent advances in communication and mobile computing have made it possible for cellular networks and digital broadcasting systems to converge on the basis of their common multimedia applications. We analyze the real-time characteristics of a cdma2000 1xEV-DO broadcast and multicast service (BCMCS) system, and propose a system architecture to provide seamless multimedia services. We found that jitter in the processing time of the Reed-Solomon decoder can lead to a discontinuity in multimedia streaming, and we therefore estimated the Reed-Solomon decoding time on a common mobile platform for a given bit error-rate or fading margin in the channel. Next we evaluated the jitter bound and the best cache size for real-time applications in BCMCS. This analysis allows us to suggest a system architecture based on the Jitter-EDD model to ensure the continuity of real-time services for multimedia applications over a 3G cellular broadcast network, and to provide a better quality of service.**

I. Introduction

As a result of recent advances in wireless and portable computing technology, the paradigm of data communication is shifting from wired to wireless environments. Spatial limitations are being overcome as the utility of mobile computing is enhanced. However, as the characteristics of the transmission medium change, new considerations emerge. The convergence of networking and broadcasting, within a framework of digital data communication services, is another contemporary trend. Digital cellular networks and digital broadcasting systems are now starting to merge, based on their common multimedia applications, even though they have different technological roots. Among many milestones passed, one is the recent publication by the Third Generation Partnership Project (3GPP) and the 3GPP2 of specifications for enhancing 3G networks to support multimedia broadcast and multicast services [1] [2].

The most common scenario for a 3G cellular broadcasting service is the transmission of n streams of data to m mobile stations to support real-time applications such as multimedia broadcasting and multicasting. Errors in 3G broadcasts are mainly the result of packet loss during delivery over the air channel, rather than packets being dropped during congestion control, which is more significant in packet switching networks. To repair transmission errors, broadcast services use forward error correction (FEC) techniques such as Reed-Solomon (RS) coding, instead of the automatic retransmission request (ARQ) used by unicast services.

This works was in part supported by BK21 Program.

We have studied MPEG video transmission over 3G cellular broadcast networks [3] [4] [5], and discovered that, using the current configuration, the video decoder can miss its timing reference because of excessive delay, caused by jitter in the processing time of the RS decoder. We aim to show that real-time scheduling can provide an effective solution to this problem.

Packet scheduling and jitter control are major topics in real-time communication. We have already proposed [6] a dynamic packet scheduler for 3G broadcasting networks, based on the earliest deadline first (EDF) scheduling algorithm, but this study ignored the processing time of the RS decoder. Verma et al. [7] proved that, if a connection does not violate its traffic specification, then its end-to-end delay and jitter requirements are guaranteed by the Jitter-EDD discipline, although their model requires a three-step channel establishment procedure to obtain the bandwidth, processing power and buffer size information needed to confirm the delay bound. Unlike unicast networks, 3G broadcasting environments such as broadcast and multicast service (BCMCS) and multimedia broadcast multicast service (MBMS) do not currently allow those procedures to be performed, and for this reason Verma et al.'s model cannot be applied directly. We now propose a new model that enables us to calculate the end-to-end delay bound and the delay jitter in the context of the processing power of cdma2000 1xEV-DO BCMCS mobiles and, using this model, we can solve the MPEG timing problem. Our model can also provide guidelines for the selection of design parameters for 3G broadcast networks.

II. Background

We will now provide some background to the BCMCS environment: the traditional model of communication delay will be introduced, and its difference from the BCMCS situation will be explained. We will also summarize the well-known real-time communication analysis technique of Jitter-EDD, which can guarantee jitter bounds to processes running on a packet switching network. Then the architecture of BCMCS and the MAC-layer FEC system will be introduced. Finally the real-time characteristics of the MPEG system and the guidelines on jitter range that are contained in the MPEG specification will be described.

1-4244-0353-7/07/$25.00 ©2007 IEEE

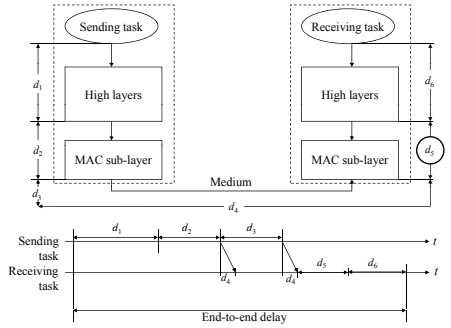

Fig. 1. Traditional end-to-end communication delay model.

A. End-to-end Communication Delay

Traditionally [8], the communication delay between distant tasks can be split into several intermediate delays, as shown in Fig. 1. The symbols $d_1, \ldots, d_6$ have the following meanings:

- d_1: Delay in crossing the upper layers during the sending task, including the application, presentation and transport layers of the OSI model.
- d_2: Queuing delay in the MAC sublayer of the sending node.
- d_3: Delay in physical transmission across the medium.
- d_4: Delay in propagation of a bit across the medium to the receiving node.
- d_5: Delay in reception, and processing time in the MAC sublayer of the receiving node.
- d_6: Delay in crossing the upper layers during the receiving task.

In order for a task to receive a message in time, it is necessary to determine and guarantee these intermediate delays. Generally, the delays d_1 and d_6 do not depend on the network, which is the case in the current context; and d_2 and d_5 have traditionally been assumed to be negligible fixed delays, supposing that all messages are immediately passed to or from the upper layers. However, it has been suggested [7] that d_2 can be considered as the main variable factor in a packet switching network. Additionally [3] [4] [5], d_5 also varies to a significant degree with the channel condition, while d_2 can easily be regulated by the scheduling process at the base station in a BCMCS environment. The transmission delay d_3 depends on the network bit-rate and the length of the message, and the propagation delay d_4 depends on the network. An upper bound on the latter can be guaranteed by reserving the medium at the appropriate time; how the medium is reserved depends on the MAC protocol of the particular network.

B. Delay Jitter Control in a Packet Switching Network

In classic earliest-due-date (EDD) scheduling, each packet is assigned a deadline, and the packets are sent in order of increasing deadlines. Delay-EDD [9] is an extension in which the server negotiates a service contract with each source. Its essence lies in the assignment of deadlines to packets. The server sets a packet's deadline to the time at which it should be sent, assuming it has been received at the time specified in the contract. This is equivalent to the expected arrival time added to the delay bound at the server. For example, if a client commits to send packets every t_1 seconds, and the delay bound at the server is t_2 seconds, then the k^{th} packet from the client will be assigned a deadline of t_1k+t_2. By reserving bandwidth at the peak rate, Delay-EDD can assure a hard delay bound on each channel.

Jitter-EDD [7] extends Delay-EDD to provide delay-jitter bounds (i.e. a bound on the minimum as well as on the maximum delay). After a packet has been served, it is stamped with the difference between its deadline and the actual dispatch time. A regulator at the entrance to the next switch holds the packet for this period before it becomes eligible to be scheduled. This mechanism provides the required minimum and maximum delay guarantees.

The idea of inserting a delay can also be effective in the BCMCS environment, although in this case the delay is not produced by indeterministic queuing at a switch, but produced by the time required for RS decoding at the receiver. In the case of BCMCS, the base station sends the broadcast packet in its pre-scheduled time slot, making the sender-side delays deterministic. A detailed description will be presented in the following section.

C. cdma2000 1xEV-DO BCMCS and FEC in the MAC layer

BCMCS in cdma2000 1xEV-DO radio access networks provide point-to-multipoint transmission of data from a single source to all users, or to a group of users in a specific area. The main purpose of BCMCS is to allow efficient use of the cdma2000 radio interface for delivery of content streams to mobile stations across an operator's network. In these services, the subscribers in a certain group receive data from the base station simultaneously, because all subscriber nodes and the base station are synchronized with a global clock. As a result, the delay d_1 can be ignored and we can determine d_2 by time-slot scheduling. Unlike the unicast cdma2000 1xEV-DO standard which uses ARQ, BCMCS utilize a forward error-correcting product code for error control, comprising an inner turbo code and an outer RS code. The architecture and description of BCMCS can be obtained from the 3GPP2 standard [2] and related articles [10] [11].

Fig. 2 shows the structure of error recovery using RS coding within the context of data transmission in current BCMCS. The broadcast framing protocol fragments higher-layer packets at the access network; the broadcast security protocol provides encryption of framing packets; and the broadcast MAC (medium access control) protocol defines the procedures used to transmit over the broadcast channel, and additionally specifies an outer code which, in conjunction with the physical-layer turbo code, forms the product code.

Each logical channel of the BCMCS uses error control blocks (ECBs) with M MAC packets per row (see Fig. 2). The variables N and K represent the number of octets and security-layer octets in an RS codeword. R is the number of parity octets: the RS decoder can recover up to R octet erasures in each codeword. 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 ECB is designed to provide a structure such

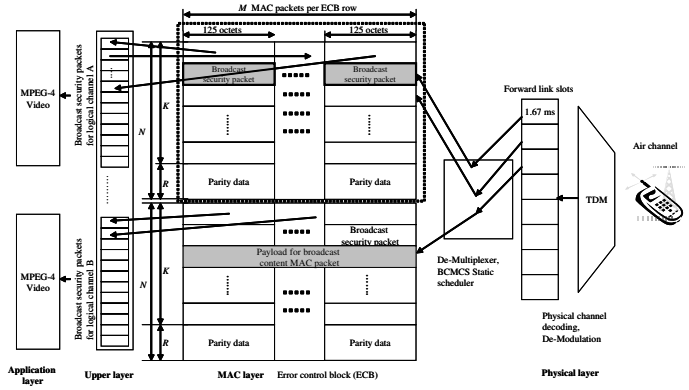

Fig. 2. Structure of the ECB and the data transmission scheme for BCMCS.

that, in the event of a physical-layer packet erasure, octets in the same position are lost from all affected RS codewords. The possible values of N in BCMCS are 32, 16 and 1, and K can take a value of 28, 26 or 24 when $N = 32$, or a value of 14, 13 or 12 when $N = 16$ [2] [10].

The burstiness of errors caused by bad channel conditions can be an important factor in selecting an appropriate data interleaving interval, determined by the width of the ECB, which is $125 \times M$ octets, as shown in Fig. 2. As the value of M increases, the time-diversity also increases and thus a mobile station which is in a time-varying shadow environment is able to recover more corrupted data. The organization of BCMCS is such that the value of M for a given ECB has to be less than or equal to 16.

D. Real-time Constraints Imposed by an MPEG System

An MPEG system is one of the best-known multimedia systems that can utilize cdma2000 1xEV-DO BCMCS. The transport stream system target decoder (T-STD) model of the MPEG system provides a guideline for managing timing constraints and buffers in an MPEG transport stream (TS) decoding process [12] [13]. According to the MPEG specifications, the TS should be constructed so that the time interval between the bytes containing the last bit of the PCR_base field in successive occurrences of the program clock reference (PCR) in the TS packets containing the PCR Packet ID (PCR_PID) for each program is no more than 100 ms:

$$|t(i) - t(i')| \leq 100 \text{ ms}, \tag{1}$$

for all values of i and i', which are the indexes of the bytes containing the last bit of consecutive PCR_base fields in the TS packets of the PCR_PID for each program. There should be at least two PCRs, from the specified PCR_PID within a TS, between consecutive PCR discontinuities, so as to facilitate phase locking and to permit extrapolation of byte delivery times. Therefore, an MPEG system should receive each TS packet with a maximum jitter of 100 ms.

III. Time Taken by Reed-Solomon Decoding

As described in the previous section, the delay in the MAC sublayer of the receiving station largely depends on the processing time of the RS decoder. Moreover, experimental results from our previous studies [3] [4] [5] show that the execution time of an RS decoder varies significantly under different error conditions. We will now summarize its operation and suggest a simple model to estimate its execution time.

A. The Reed-Solomon Decoding Process

Reed-Solomon is an algebraic code belonging to the class of BCH (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 an RS codeword is of length R bytes, then the decoder can correct up to R corrupted bytes if their positions are known, or detect and correct up to $R/2$ bytes if the positions of the errors are unknown. In cdma2000 1xEV-DO, an RS codeword consists of a sequence of 8-bit bytes, containing a configurable number of parity bytes.

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. An 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 [14]: 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}, which 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 R is the error-correcting capability of the code, then $2R$ syndromes can be computed, as follows:

$$\begin{aligned} S_j &= \gamma(\alpha^j) \\ &= e_{i_1}(\alpha^j)^{i_1} + e_{i_2}(\alpha^j)^{i_2} + \cdots + e_{i_\nu}(\alpha^j)^{i_\nu} \\ &= e_{i_1}(\alpha^{i_1})^j + e_{i_2}(\alpha^{i_2})^j + \cdots + e_{i_\nu}(\alpha^{i_\nu})^j \\ &= e_{i_1} X_1^j + e_{i_2} X_2^j + \ldots + e_{i_\nu} X_\nu^j, \end{aligned} \tag{2}$$

where X_ι is defined as $\alpha^{i_\iota} (\iota = 1, 2 \ldots \nu)$.

These syndrome equations can be obtained as a sequence of $2R$ algebraic equations, which can be linearized by defining the error locator polynomial $\Lambda(x)$ as follows:

$$\begin{aligned} \Lambda(x) &= \prod_{\iota=1}^{\nu} (1 - X_\iota x) \\ &= \Lambda_\nu x^\nu + \Lambda_{\nu-1} x^{\nu-1} + \ldots + \Lambda_1 x + 1. \end{aligned} \tag{3}$$

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

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_R \\ S_2 & S_3 & \cdots & S_{R+1} \\ S_3 & S_4 & \cdots & S_{R+2} \\ \vdots & \vdots & \ddots & \vdots \\ S_R & S_{R+1} & \cdots & S_{2R-1} \end{bmatrix} \begin{bmatrix} \Lambda_R \\ \Lambda_{R-1} \\ \Lambda_{R-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_{R+1} \\ -S_{R+2} \\ -S_{R+3} \\ \vdots \\ -S_{2R} \end{bmatrix}. \tag{4}$$

Once the syndromes have been computed, there are a number of ways to find the error locations. In our RS decoder we

use Berlekamp's iterative algorithm, which takes the following form [14]:

1) For each codeword received, compute the syndrome sequence $S_1, \ldots, S_{2R}$.
2) Initialize the algorithm variables: $\kappa = 0, \Lambda^{(0)}(x) = 1, L = 0$ and $T(x) = x$.
3) Set $k = k + 1$ and then compute the discrepancy $\Delta^{(k)}$ as follows:

$$\Delta^{(k)} = S_k - \sum_{i=1}^{L} \Lambda_i^{(k-1)} S_{k-1}. \quad (5)$$

4) If $\Delta^{(k)} = 0$, then go back to Step 8.
5) Modify the connection polynomial:

$$\Lambda^{(k)}(x) = \Lambda^{(k-1)}(x) - \Delta^{(k)} T(x). \quad (6)$$

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 $k < 2R$, then go back to Step 3.
10) Determine the roots of $\Lambda(x) = \Lambda^{(2R)}(x)$. If these roots are distinct and lie in the right field, then determine the error magnitudes, correct the corresponding locations in the codeword, 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 the received codeword has ν errors and f erasures, an erasure locator polynomial is defined as follows:

$$\Gamma(x) = \prod_{\iota=1}^{f} (1 - Y_\iota x) \text{ where } Y_\phi = \alpha^{j_\phi}, (\phi = 1, 2 \ldots, f) \ . \quad (7)$$

In summary, 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:

$$\Phi(x) = (\Gamma(x)[1 + S(x)] - 1) \bmod x^{2R+1}. \quad (8)$$

4) Apply the Berlekamp algorithm to find the connection polynomial $\Lambda(x)$, using the modified syndrome coefficients Φ_i, $i = f + 1, \ldots, 2R$.
5) Find the roots of $\Lambda(x)$, and thus the error locations.
6) Determine the magnitudes of the errors and erasures using the error/erasure locator polynomial, which is

$$\Psi(x) = \Lambda(x)\Gamma(x). \quad (9)$$

Then the error and erasure values are

$$e_{i_k} = \frac{-X_k \Omega(X_k^{-1})}{\Psi'(X_k^{-1})}, \ f_{i_k} = \frac{-Y_k \Omega(Y_k^{-1})}{\Psi'(Y_k^{-1})}, \quad (10)$$

where

$$\Lambda(x)[1 + \Phi(x)] = \Omega(x) \bmod x^{2R+1},$$
$$X_\phi = \alpha^{i_\phi}, (\phi = 1, 2 \ldots, \nu).$$

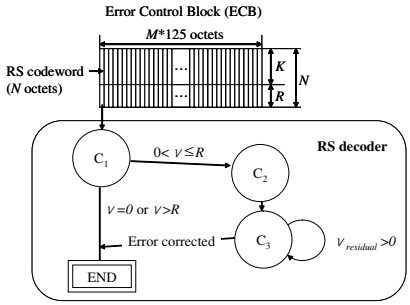

Fig. 3. Simplified execution-time model of the RS decoder.

B. Execution Time Model of the Reed-Solomon Decoder

Having briefly surveyed the RS decoding procedure, we now focus on finding a correlation between the error condition and the execution time of the decoder.

In the BCMCS environment, the RS decoder is able to obtain the numbers and positions of errors from the physical layer. Using this error information, the decoder can decide whether to continue the decoding process to erase errors, or not. The decoder should continue decoding and erase the corrupted data when a codeword includes at least one error, but not when there are more than R. If these conditions are set, the subsequent steps taken by the decoder can be simplified into two stages. First a series of operations is performed to build the syndromes. Then the decoder loops through a second series of operations that are repeated until no more residual errors remain.

Fig. 3 shows our simplified model of the time taken by RS decoding. There are three computational components in the model, C_1, C_2, and C_3, which have the following functions:

- C_1 decodes a codeword and decides whether to try erasing the errors or not.
- C_2 builds the corresponding syndromes.
- C_3 are the erasure loops.

After receiving the codewords, the decoder invokes C_1 to decode them and then decides what to do next, depending on the value of ν. Thus, C_1 is performed once for each codeword received. If the number of errors is greater than zero and less than or equal to R, then C_2 should be invoked. From C_2, the system obtains the syndromes that correspond to the error pattern in those codewords. C_3 is then repeatedly invoked until no more errors remain and the decoding of that codeword is complete.

Based on this model, we constructed a simple equation that expresses the total time taken by RS decoding (C_{CW}), split between the computation components C_1, C_2, and C_3, and depending on the number of errors in each codeword (ν):

$$C_{CW}(\nu|\nu = 0, \nu > R) = C_1$$
$$C_{CW}(\nu|0 < \nu \leq R) = C_1 + C_2 + \nu \times C_3. \quad (11)$$

As mentioned in Section II-C, an ECB is composed of $125 \times M$ codewords, and each codeword may contain a different number of errors. So we can determine the total execution time for an ECB if we know its composition, which we can

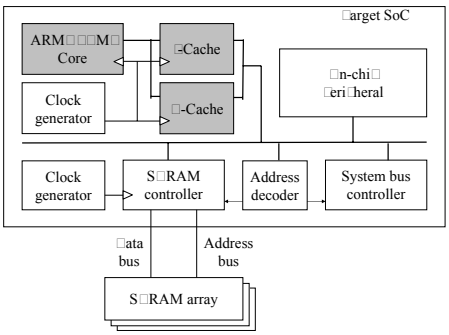

Fig. 4. Block diagram of the experimental environment.

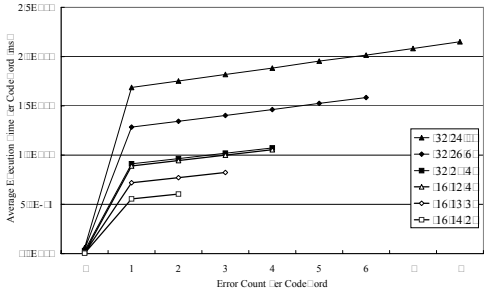

Fig. 5. Execution time of the RS decoder.

determine using the following equation:

$$C_{ECB} = \sum_{\xi=1}^{125M} C_{CW}(\xi) = 125MC_1 + \sum_{\nu=1}^{R} N(\nu)(C_2 + \nu C_3), \quad (12)$$

where ξ denotes a codeword and $N(\nu)$ is the number of codewords containing ν octet errors.

C. Experimental Environment

The execution time of the RS decoder was measured using the SEE (SNU Energy Explorer) [15]. The block diagram of our experimental environment is shown in Fig. 4. In the following experiments, both the ARM7 core and the SEC 128 Mbit SDRAM array (K4S280832A) were operated at a clock speed of 100 MHz, and we used 8K instruction and data caches, with 4-way associativity. This organization is abstracted from the general ARM-based embedded system and MSM6000 chipset provided by QUALCOMM, which also uses the ARM7 TDMI core [16].

D. Experimental Results

We measured the average execution time for each computational component, and the total time spent by the RS decoder on each codeword. We repeated this experiment 100 times with randomly generated codewords to get average values. The average execution time of each computational component is given in Table I, and the total execution times per codeword are plotted in Fig. 5.

We found that, once the type of RS code to be used has been chosen, the values of C_1 and C_2 are affected very little by the contents of each codeword. However, factors such as the positions of the error octets do affect C_3 to some degree. The main reason for this seems to be that Berlekamp's algorithm invokes a different style of iteration depending on the characteristics of the codeword, as described in Section III-A. Nevertheless, Table I shows that the total execution time per codeword mainly depends on C_2, and also that the standard deviation (σ) of C_3 is not large, so we will use the average values (μ) of C_3, in order to keep our analytic model as simple as possible.

TABLE I
AVERAGE EXECUTION TIME OF EACH COMPUTATIONAL COMPONENT.

RS code	C_1	C_2	C_3	
	(ms)	(ms)	μ (ms)	σ
(32,24,8)	0.0609	1.5594	0.0663	0.0033
(32,26,6)	0.0451	1.1790	0.0600	0.0016
(32,28,4)	0.0295	0.8282	0.0537	0.0012
(16,12,4)	0.0119	0.8235	0.0548	0.0005
(16,13,3)	0.0081	0.6760	0.0349	0.0302
(16,14,2)	0.0043	0.5346	0.0164	0.0284

IV. Error Pattern Analysis

A. Gilbert's Channel Model of Rayleigh Fading

We used the Gilbert channel model [17] [18] [19] to simulate data errors which arise in transmission over fading channels. Fading in the air channel is assumed to have a Rayleigh distribution. 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 with relatively long error-free intervals between them.

By choosing different values for the input bit error-rate (BER) and for f_dT (which is the Doppler frequency normalized to the data-rate, where f_d is the Doppler frequency, equal to the mobile velocity divided by the carrier wavelength [20]), we can model different degrees of correlation in the fading process. 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 values of f_dT (fast fading).

In the equations that follow, α is the probability that the i^{th} bit is corrupted, given that the $(i-1)^{\text{th}}$ bit was transmitted accurately, and β is the probability that the i^{th} bit is accurate, given that the $(i-1)^{\text{th}}$ bit was corrupted. The steady-state BER (ε) is then obtained as follows:

$$\varepsilon = \frac{\alpha}{\alpha + \beta} \ . \quad (13)$$

If the Rayleigh fading margin is F, the average BER can be expressed as

$$\varepsilon = 1 - e^{-\frac{1}{F}} \ . \quad (14)$$

Using this equation and the equations which follow, we can now derive values for α and β. The average lengths of

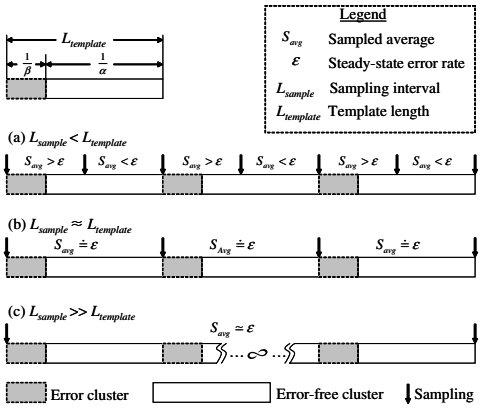

Fig. 6. The effect of sampling interval.

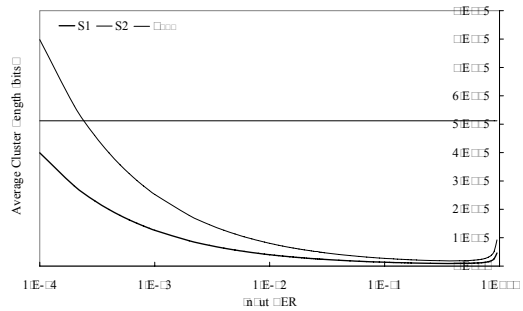

Fig. 7. Average values of $L_{template}$ and L_{ECB}.

sequences of error bits and sequences of normal bits are given by $1/\beta$ and $1/\alpha$, where

$$\beta = \frac{Q(\theta, \rho\theta) - Q(\rho\theta, \theta)}{e^{\frac{1}{F}} - 1}, \text{ and } \theta = \sqrt{\frac{2/F}{1-\rho^2}} \ . \qquad (15)$$

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

$$Q(x, y) = \int_y^\infty e^{-\frac{x^2+w^2}{2}} I_0(xw) w \ dw \qquad (16)$$

is the Marcum-Q function. Thus, the relationship between the given BER and the Markov parameter can be represented as

$$\beta = \frac{1-\varepsilon}{\varepsilon}[Q(\theta, \rho\theta) - Q(\rho\theta, \theta)] \ , \qquad (17)$$

where

$$\theta = \sqrt{\frac{-2\log(1-\varepsilon)}{1 - J_0^2(2\pi f_d T)}} \ .$$

B. Experimental Environment

In the following experiments, we used values of 1.0×10^{-4} (S1) and 5.0×10^{-5} (S2) for $f_d T$, which represent fast and slow fading conditions respectively, with a reference channel data-rate of 409.6 Kbps and a carrier frequency of 900MHz.

C. Sufficiency of the Interleaving Space

As described in Section II-C, the BCMCS system uses the ECB to scatter error clusters into a sparse pattern so as to maximize the error recovery performance of the RS decoder. We suggest that, so long as sufficient interleaving space is provided in the codes, the Rayleigh distribution of an error cluster is converted into a random distribution. We will now present a model to support this contention.

Fig. 6 shows the relation between the sampling interval (L_{sample}) and the sampled average value (S_{avg}). As mentioned in Section IV-A, the average lengths of sequences of uncorrupted and error bits are $1/\beta$ and $1/\alpha$ respectively. The fluctuating pattern of the channel condition is made up of a repetition of these error and normal clusters. The average length of one sequence of normal bits and one sequence of error bits, which we will call a template ($L_{template}$), can be expressed as follows:

$$\begin{aligned} L_{template} &= \frac{1}{\alpha} + \frac{1}{\beta} \\ &= \frac{1}{(1-\varepsilon)[Q(\theta, \rho\theta) - Q(\rho\theta, \theta)]}. \end{aligned} \qquad (18)$$

We now consider the relative sizes of L_{sample} and $L_{template}$. If L_{sample} 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 localized into a few samples and S_{avg} for these samples will be exaggerated. The other samples will contain relatively few errors, and S_{avg} will naturally be underestimated for these samples.

As L_{sample} converges with $L_{template}$, the fluctuation of S_{avg} will stabilize. After this point, the distribution of errors in the samples will approach more and more closely to the steady-state error-rate as L_{sample} increases further; and finally $S_{avg} \to \varepsilon$ as $L_{sample} \to \infty$.

In BCMCS, as described in Section II-C, the errors in each channel are interleaved by the ECB. Thus, the size of the ECB (L_{ECB}) can be considered to be equivalent to L_{sample}, and is defined as follows:

$$L_{ECB} = 125MN\frac{bits}{octet}, \qquad (19)$$

where $bits/octet$ is the bit-count of the transmission octet of the specific communication system.

Fig. 7 shows the theoretical variation of $L_{template}$ with the input BER, and the corresponding curve for L_{ECB} when M is 16 and N is 32. The value of $L_{template}$ under condition S1 is smaller than L_{ECB} when the BER is between 1.0×10^{-4} and 1.0. Within that range, we can expect the channel to be in a situation similar to that shown in Fig. 6 (b) or (c). But, if the BER is below 2.4×10^{-4}, then the value of $L_{template}$ for S2 is larger than L_{ECB}, and the situation is similar to that in Fig. 6 (a). However, in both those cases, the increase in execution time caused by a relatively small number of error clusters is marginal, and our results show that the maximum execution time will not be affected. Thus, we are able to assume that the BCMCS system can select an appropriate ECB size parameter to achieve sufficient interleaving under most channel conditions, when the influence of the channel on the error distribution may be considered to be random.

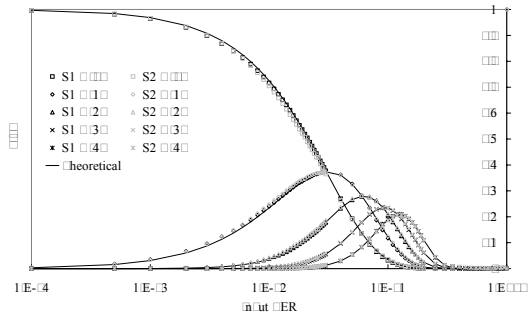

Fig. 8. Error count distribution.

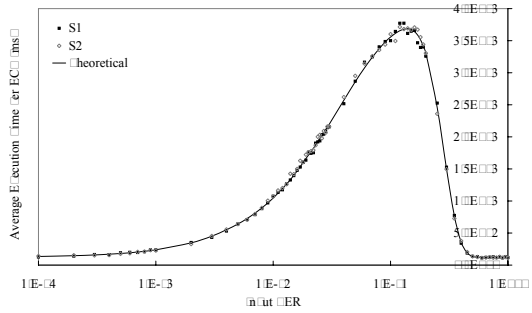

Fig. 9. Experimental execution times.

D. Error Probability Function after Adequate Interleaving

As explained in the previous section, the pattern of error clusters occurring in the channel can be regarded as random after sufficient interleaving has been provided. In this case, the probability that a codeword contains i errors can be expressed by the following binomial equation:

$$P(\nu = i) = \binom{N}{i}\varepsilon^i(1-\varepsilon)^{N-i}. \tag{20}$$

The residual error-rate ($\varepsilon_{residual}$) after RS decoding, which is an indicator of error recovery performance, can then be derived as follows:

$$\varepsilon_{residual} = \sum_{\nu=R+1}^{N}\frac{\nu}{N}\binom{N}{\nu}\varepsilon^{\nu}(1-\varepsilon)^{N-\nu}. \tag{21}$$

E. Simulation Results

Fig. 8 shows how the simulation results relate the input BER to the probability that a codeword contains a certain number of errors. We performed this simulation by generating error maps for the two values of f_dT that correspond to conditions S1 and S2. Each map contains 2.0×10^6 codewords so as to stabilize the results. We counted the number of codewords containing errors, and divided the result by the total number of codewords to get the average probability of an error occurring. Then we compared the results with the theoretical curves, generated from Eq. (20). Fig. 8 shows that the results for a fast-moving mobile (S2) follow the theoretical curve closely, and the results for slow fading (S1) are almost as good.

V. Ensuring Service Continuity in 3G Broadcast Networks

A. The Effect of Channel Condition on Reed-Solomon Decoding Time

The expected number of codewords containing errors can be expressed as follows:

$$E[N(\nu)|\nu = i] = 125MP(\nu = i). \tag{22}$$

From Equations (12) and (20), we are then able to derive a simple equation which expresses the expected RS execution time per ECB for a given BER:

$$E[C_{ECB}(\varepsilon)] = 125M[C_1 + \sum_{\nu=1}^{R}\binom{\eta}{\nu}\varepsilon^{\nu}(1-\varepsilon)^{\eta-\nu}(C_2 + \nu C_3)]. \tag{23}$$

B. Experimental Results

Using the SEE [15], we measured the average time taken by the RS decoder to process each ECB. Each stream of 5.12×10^6 bits was generated by the 2-state Markov model and interleaved, with a value of L_{ECB} of 5.12×10^5 bits. The RS code was (32,24,8). We measured the time required to decode each ECB 10 times, and took the average value. And then we compared the results with curves generated from Eq. (23). Fig. 9 shows that the execution times for both fading conditions, S1 and S2, almost coincide the with theoretical curves. As described in Section III-D, C_1 and C_2 remain nearly the same for a given RS code, and it is the variation in C_3 that produces the difference between the measured and the theoretical results. This means that we can predict the RS decoding time providing that we have information about the channel error-rate, which can easily be obtained.

C. Estimating Maximum Jitter and Buffer Size

From our experiments, we can predict the time required for RS decoding at a mobile for a given BER. However, service providers manage the coverage area and transmission power level of each base station within a pre-determined range of tolerable error-rates. So it is necessary to estimate the jitter value for a given range of BERs. This involves calculating the execution time range of the RS decoder, which contributes jitter to the end-to-end communication delay. If the BER of the channel is between 0 and ε and the period of an ECB is T_{ECB}, then the jitter value $J(\varepsilon)$ for that ECB can be derived as follows:

$$\begin{aligned} J(\varepsilon|\varepsilon < \varepsilon') &= \frac{E[C_{ECB}(\varepsilon)] - min(C_{ECB})}{T_{ECB}} \\ &= \frac{125M}{T_{ECB}} \times \\ &\quad \sum_{\nu=1}^{R}\binom{\eta}{\nu}\varepsilon^{\nu}(1-\varepsilon)^{\eta-\nu}(C_2 + \nu C_3); \\ J(\varepsilon|\varepsilon \geq \varepsilon') &= \frac{125M}{T_{ECB}} \times \\ &\quad \sum_{\nu=1}^{R}\binom{\eta}{\nu}\varepsilon'^{\nu}(1-\varepsilon')^{\eta-\nu}(C_2 + \nu C_3). \end{aligned} \tag{24}$$

And ε' can be obtained from the following derivatives:

$$\left[\frac{dE[C_{ECB}(\varepsilon)]}{d\varepsilon}\right]_{\varepsilon=\varepsilon'} = 0, \left[\frac{d^2E[C_{ECB}(\varepsilon)]}{d\varepsilon^2}\right]_{\varepsilon=\varepsilon'} < 0.$$

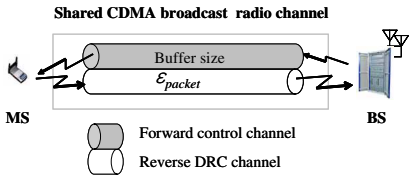

Fig. 10. Information exchange in BCMCS.

The size of buffer needed to store one ECB, and the maximum jitter bound, if that ECB is to reach the upper-layer MPEG system decoder during the correct interval, can be expressed by the following equation:

$$L_{buffer}(\varepsilon) = L_{ECB}(1 + 2J(\varepsilon)). \tag{25}$$

D. System Architecture to Ensure Service Continuity in 3G Broadcast Networks

In a standard BCMCS network, each node reports its error-rate at the arrival of every packet, using the data request channel (DRC) on the reverse link. The base station can use this information, together with inbuilt performance profiles, to determine the end-to-end delay bounds and jitter values for each mobile. The base station can then decide the appropriate buffering level for each service so as to match the current channel conditions of its subscribers. Each mobile can then be told its buffering level by means of an overhead message on the forward control channel. Fig. 10 shows a simple implementation of this idea.

Our model will provide information about the delay and jitter bounds of the service, while bandwidth is determined by time-slot scheduling performed at the base station. A measurement of the processing power available at each mobile station can also be provided to the system. If a base station schedules its service using that information and the Jitter-EDD algorithm, then we can guarantee the continuity of the service between the base station and the output of the RS decoder, using Verma et al.'s proof [7] and our EDF-based scheduling algorithm [6]. As described in Section II-D, if the RS decoder provides the TS within its deadlines, then the MPEG system will meet the rest of the timing constraints.

Additionally, it is possible to change the buffering level of each service stream to minimize the end-to-end delay in the system. The base station can dynamically determine an appropriate buffering level for each stream, so as to accommodate the largest jitter value associated with any of its subscribers. For example, suppose that a base station is using the (32,24,8) RS code, M is 16, and T_{ECB} is 2,000 ms, allowing a video stream to be broadcast at 409.6 kbps. If the MSM6000-based mobile which is experiencing the worst reception reports the upper bound on its BER as 1.0×10^{-2}, then Eq. 24 can be used to determine its expected jitter for each ECB, which will be around 450 ms. From Eq. 25, the base station can determine the appropriate buffer size, which is 972,884 bits, and then use the forward control channel to notify all the MSM6000-based mobiles.

Our model can also provide some guidelines for planning and designing a network. For example, a service provider needs to decide on an acceptable fading margin to determine the deployment of each cell. Our model can be modified to provide this design consideration:

$$\begin{aligned} E[C_{ECB}(\varepsilon)] &= 125M \times \\ &(C_1 + \sum_{\nu=1}^{R} \binom{\eta}{\nu} (1 - e^{-\frac{1}{F}})^{\nu} e^{\frac{\nu-\eta}{F}} (C_2 + \nu C_3)). \end{aligned} \tag{26}$$

Eq. (23) can also be used to formulate a guideline for deciding the processing power that a mobile needs to deal with a particular BER, or vice versa.

VI. Conclusions

We have presented a model which can be used to ensure service continuity in 3G multimedia broadcasting environments which include RS decoding. We have provided a mathematical model which can predict the jitter of the RS decoding time from the processing power available at a mobile and the channel condition that is is experiencing. This contrasts with Jitter-EDD and its derivatives, which require a three-stage connection establishment procedure to obtain the same information, making its implementation in a broadcasting environments unnecessarily intricate. We expect our model to be practically useful in the planning and design of 3G broadcasting networks and in their daily operation.

References

[1] *Multimedia Broadcast/Multicast Service (MBMS); Architecture and Functional Description*, 3GPP Std. TS 23.246, June 2005.

[2] *CDMA2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification*, 3GPP2 Std. C.S0054, Rev. 1.0, Mar. 2005.

[3] Y. Cho, K. Kang, J. Lee, and H. Shin, "Proactive reed-solomon bypasss (prsb): A technique for real-time multimedia processing in 3g cellular broadcast networks," in *Proc. of the 11th IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications*, Aug. 2005, pp. 532–538.

[4] Y. Cho, K. Kang, Y. Cho, and H. Shin, "Improving mac-layer error recovery for 3g cellular broadcasts," in *Proc. of the 20th IEEE Int'l Conf. on Advanced Information Networking and Applications*, vol. 2, Apr. 2006, pp. 418–422.

[5] J. Lee, K. Kang, Y. Cho, J. Kim, and H. Shin, "Efficient error recovery for multimedia data transmission over 3g cellular broadcast networks," in *Proc. of the 20th IEEE Int'l Conf. on Advanced Information Networking and Applications*, vol. 1, Apr. 2006, pp. 683–688.

[6] K. Kang, Y. Cho, J. Cho, and H. Shin, "Scheduling scalable multimedia streams for 3g cellular broadcast and multicast services," IEEE Trans. Veh. Technol., to be published.

[7] D. C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proc. of TRICOMM'91*, Apr. 1991, pp. 35–46.

[8] F. Cottet, J. Delacroix, C. Kaiser, and Z. Mammeri, *Scheduling in Real-Time Systems*. Wiley, Dec. 2002.

[9] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, pp. 368–379, 1990.

[10] P. Agashe, R. Rezaiifar, and P. Bender, "Cdma2000 high rate broadcast packet data air interface design," *IEEE Commun. Mag.*, vol. 42, no. 2, pp. 83–89, 2004.

[11] J. Wang, R. Sinnarajah, T. Chen, Y. Wei, and E. Tiedemann, "Broadcast and multicast services in cdma2000," *IEEE Commun. Mag.*, vol. 42, no. 2, pp. 76–82, 2004.

[12] *Information technology - Generic Coding of Moving Pictures and Associated Audio Information: Systems*, ISO/IEC Std. 13 818-1, 2000.

[13] *Coding of Audio-Visual Objects - Part 1: Systems*, ISO/IEC Std. 14 496-1, 2004.

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

[15] I. Lee, Y. Choi, Y. Cho, Y. Joo, H. Lim, H. G. Lee, H. Shim, and N. Chang, "Web-based energy exploration tool for embedded systems," *IEEE Des. Test. Comput.*, vol. 21, no. 6, pp. 572–586, 2004.

[16] Msm6000 chipset solution. QUALCOMM Inc. [Online]. Available: http://www.cdmatech.com/products/msm6000_chipset_solution.jsp

[17] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, Sept. 1960.

[18] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Commun.*, vol. 46, no. 11, pp. 1468–1477, Nov. 1998.

[19] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE Trans. Commun.*, vol. 45, no. 6, pp. 660–667, June 1997.

[20] W. C. Jakes, *Microwave Mobile Communications*. Wiley, May 1994.