Energy Characterization of Reed-Solomon Decoding in 3G Broadcasting

Junhee Ryu, Kyungtae Kang, Yongwoo Cho, Joonho Lee, and Heonshik Shin

School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea 151–744
Email: {jhryu, ktkang, xtg05, jlee108, shinhs}@cslab.snu.ac.kr

Telephone: (822) 880–7298
Fax: (822) 874–3104

***Abstract*— 3GPP2 has recently introduced the Broadcast and Multicast Services (BCMCS) architecture for cdma2000 1xEV-DO wireless networks to enable service providers to broadcast multimedia content such as MPEG-4 video. For MAC-layer forward error correction the BCMCS scheme uses the Reed-Solomon (RS) decoding process, which consumes a considerable amount of energy on mobile phones. To address this problem, we first characterize the energy consumption of the decoder with respect to its components: the data decoder, error locator and erasure decoder. Based on this detailed energy characterization we propose an analytic energy model which takes account of different levels of bit error rate in the forward traffic channel. This model is then verified experimentally on an ARM microprocessor-based testbed. Our results will help design energy-efficient BCMCS systems for 3G cellular networks.**

I. Introduction

Recently, great efforts have been made to provide a broadcast service environment for next-generation wireless networks. The Third Generation Partnership Project-2 (3GPP2) has produced technical specifications for the cdma2000 high-rate broadcast packet-data air interface [1] [2] which can support broadcast and multicast services (BCMCS) [3] [4] [5]. In designing a mobile phone device to support multimedia in a 3G broadcast environment, energy conservation is more critical than the performance of a particular application program, especially because the high-speed packet data services supported by third-generation systems are expected to consume more energy than conventional circuit-switched voice services. However, strict limits on the allowable form factor and weight of mobile devices limit the size of battery that can be specified. Many techniques to measure and reduce energy consumption at the chip level have therefore been proposed, but there has been no research on the energy used at the computing system level by mobiles operating in a 3G broadcast environment.

Our study focuses on the characterization and analysis of the energy consumed by a mobile station receiving a broadcast service, and we propose a model of its energy consumption, focusing especially on error recovery. As a wireless radio network is prone to errors, and these tend to occur in bursts, error control is an essential task in a mobile designed to receive broadcast services. Forward error correction (FEC) is commonly associated with video broadcast applications (unlike the unicast service in cdma2000 1xEV-DO [6] [7]), due to the strict delay requirements of video streams, combined with the semi-reliable nature of a wireless network. In current cdma2000 1xEV-DO BCMCS, Reed-Solomon (RS) coding is used for FEC at the medium access control (MAC) layer [1] [2]. Because error recovery has to be performed on all the multimedia data received by the mobile, the decoding process must be a major target in reducing energy consumption.

Our experiments were performed on the ARM7TDMI-based testbed, including a 4K I-cache and an 8K D-cache, which are shown to be appropriate sizes for RS decoding. Energy consumption was measured using the SEE [8], which proved to be a reliable tool in earlier work. It measures the energy consumed by each computational component of the RS decoder, which allows us to construct our energy model. We have obtained this energy data for different air channel conditions and a large range of RS coding parameters. From the results of these experiments, we are able to suggest guidelines which should have a significant impact on the design of low-energy mobiles receiving broadcast services.

The remainder of this paper is organized as follows: In Section II, we present the background to our study and, in Section III, we introduce the energy model of the RS decoding process which we will use in Section IV to derive and analyze the total energy consumption of the RS decoder during error recovery under a range of simulated air channel conditions. The verification of the energy model by experiment is described in Section IV, and in Section V we draw conclusions.

II. Background

A. Reed-Solomon Codes

Bose-Chaudry-Hocquehen (BCH) codes are a large class of powerful random error-correcting cyclic codes. The Reed-Solomon code is one of the best-known classes of non-binary BCH codes. It allows the correction of corrupted information using parity data, and is commonly utilized as an outer code for data communication or data storage.

If the parity part of a sequence is of length R octets, an RS code can correct up to R corrupted octets if their positions are

1-4244-0419-3/06/$20.00 ©2006 IEEE

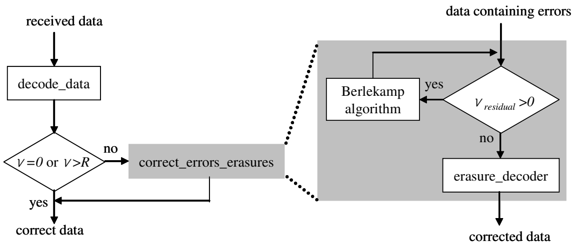

Fig. 1. Flow diagram of the RS decoding process.

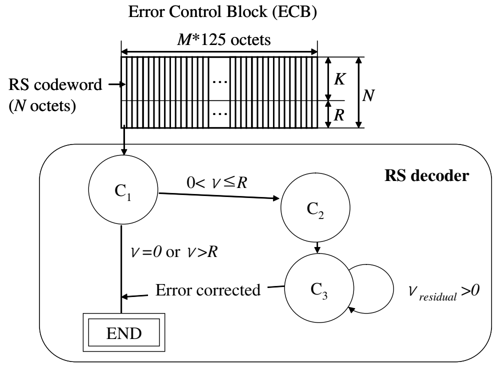

Fig. 2. Computational components of the RS decoding process.

known, or detect and correct up to $R/2$ octets if the positions of the errors are unknown. The cdma2000 standard specifies an RS code with 8-bit octets, and a configurable number of parity octets. The longest sequence that can be generated is 255 bytes, but in practice shorter sequences are used.

B. Error recovery in BCMCS

In BCMCS, the broadcast framing protocol provides for the fragmentation of higher-layer packets at the access network; the broadcast security protocol specifies the encryption of framing packets; and the broadcast MAC 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. As already mentioned, Reed-Solomon was chosen as the outer code for cdma2000 BCMCS, and the broadcast MAC-layer packets have a fixed size of 125 bytes. The protocol is completed by the broadcast physical layer, which specifies the structure of the broadcast channel.

Each logical channel uses error control blocks (ECBs) with M MAC packets per row. The variables N and K represent the number of octets and security-layer octets in a Reed-Solomon codeword. R is the number of parity octets: the Reed-Solomon decoder can recover up to R octet erasures in each codeword. Reed-Solomon 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. To decode a Reed-Solomon codeword correctly, the broadcast MAC protocol needs to receive at least K of the N octets in that codeword. If all K data octets are received without errors, decoding is not needed. The data octets which have been successfully received are simply forwarded to the upper layer of the BCMCS protocol suite. 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 $N = 16$ [1] [2].

One of the most significant environmental factors affecting channel condition is fading. This is correlated with the burstiness of errors. A slow-moving mobile station tends to receive longer bursts of errors than one that is moving more quickly, which escapes more rapidly from shadowed locations where reception is poor. A Reed-Solomon code of (N, K, R) cannot recover any lost data if the corrupted portion of a codeword is larger than R. For this reason, the performance of error correction will drop if the burst-length of errors becomes so large that the ECB cannot interleave them sufficiently.

C. Reed-Solomon decoding

An RS decoder operates as shown in Fig. 1. When a data stream is received, syndrome bytes are created. Their number is proportional to the number of parity octets, regardless of the quality of the data stream. So the time required to generate a syndrome depends only on the number of parity octets, 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. The locations of errors are found using Berlekamp's algorithm [9] [10], and the original data can then be 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 octet error rate at the input to the decoder.

III. Energy model of the RS decoding process

A Reed-Solomon decoder is made up of three computational components, as shown in Fig. 2: the data decoder (C_1) and the two functional units of the Berlekamp algorithm, which are the error locator (C_2) and the erasure decoder (C_3). When the coded data arrives, the Reed-Solomon decoder reads a codeword from the ECB and computes its syndrome, which it uses to decide whether to try erasing the errors or not. If errors are found, the Berlekamp algorithm creates a new syndrome for error correction (in the error locator), which is subsequently used to recover the errors (in the erasure decoder). Erasure decoding is repeated until all the errors have been corrected. Thus C_1 operates on every codeword received, while C_2 is called once per codeword if there are errors. Finally, C_3 is repeatedly invoked to correct errors until no more remain (i.e. $\nu_{residual} = 0$).

From a consideration of this whole process, the total energy consumption ($E_{codeword}$) during the RS decoding of a codeword can be predicted by summing the expected energy required by each component to deal with the number of errors

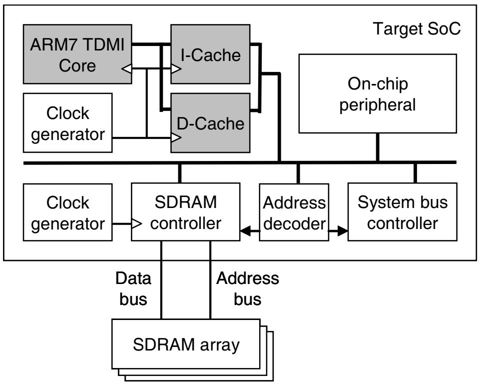

Fig. 3. Testbed for RS decoding.

(ν) contained in that codeword:

$$E_{codeword}(\nu) = \begin{cases} E_{C_1} & (\text{if } \nu = 0, \nu > R) \\ E_{C_1} + E_{C_2} + \nu E_{C_3} & (\text{if } 0 < \nu \leq R), \end{cases} \quad (1)$$

where E_{C_i} is the expected energy consumption of computational component C_i.

A. Measurement of basic energy consumption

1) Testbed for RS decoding: The execution time and energy consumption of RS decoding were measured using the SEE (SNU Energy Explorer) [8]. In the following experiments, both the ARM7TDMI core and the SEC 128 Mbit SDRAM array (K4S280832A) were operated at a clock speed of 100 MHz, and the cache we used has 4-way associativity. This organization is abstracted from the general ARM7-based embedded system. Fig. 3 shows the block diagram of our RS decoding testbed.

Fig. 4 shows how the I-cache miss rate varies with cache size, while the decoder is correcting the maximum number of octet errors per codeword that can be dealt with by each coding scheme. In the figures, 32-24-8-ν denotes a (32,24,8) code with ν octet errors in each codeword, and similarly for other codes. As the number of errors increases, the I-cache miss rate decreases. The Berlekamp algorithm runs more often but, because it is compact, it achieves many cache hits, so the absolute miss rate is low in all cases. The cache miss rate

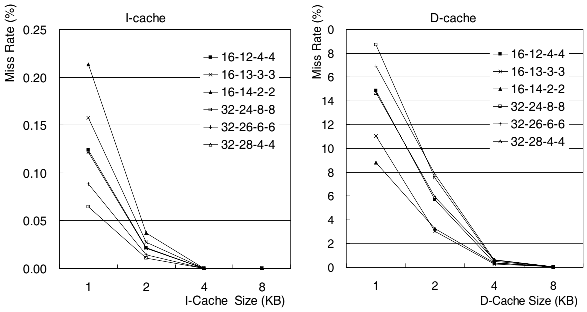

Fig. 4. Cache miss rates.

TABLE I
EXPECTED EXECUTION TIME OF EACH COMPUTATIONAL COMPONENT.

	Execution time (μs)		
RS code	C_1	C_2	C_3
(16,12,4)	11.9	823.5	54.8
(16,13,3)	8.1	658.6	52.3
(16,14,2)	4.3	501.9	49.1
(32,24,8)	60.9	1559.4	66.3
(32,26,6)	45.1	1179.0	60.0
(32,28,4)	29.5	828.2	53.7

TABLE II
EXPECTED ENERGY CONSUMPTION OF EACH COMPUTATIONAL COMPONENT.

	Energy consumption (μJ)		
RS code	C_1	C_2	C_3
(16,12,4)	4.2	266.2	17.0
(16,13,3)	3.0	212.4	16.2
(16,14,2)	1.7	161.7	15.2
(32,24,8)	19.8	501.8	20.7
(32,26,6)	14.8	380.0	18.3
(32,28,4)	9.8	266.1	16.8

stabilizes when the size of the I-cache is 4KB, so we use an I-cache of this size in our testbed. Similarly, the miss rate of the D-cache stabilizes around 8KB, which is also apparent from Fig. 4.

2) Energy consumption by the decoding components : Tables I and II show experimental values of the average execution time and energy consumption for each computational component during RS decoding of one codeword when there is a single octet error. The results are obtained by averaging the execution time and energy consumption over 10 trials ($\kappa = 10$) while varying the error location as follows:

$$T_{C_i} = \text{Avg}\left[\frac{1}{{}_N C_1}\sum_{j=0}^{N} T_{C_i}(j)\right]_{\text{trials}=\kappa}, \quad (2)$$

$$E_{C_i} = \text{Avg}\left[\frac{1}{{}_N C_1}\sum_{j=0}^{N} E_{C_i}(j)\right]_{\text{trials}=\kappa}, \quad (3)$$

where $T_{C_i}(j)$ and $E_{C_i}(j)$ are the execution time and energy consumption respectively, when an error occurs in the j^{th} octet of a codeword.

The number of loops performed by the Berlekamp algorithm while generating the polynomial, lambda and the syndrome bytes required for error detection and correction is proportional to the amount of parity information. Generating the polynomial used to erase or evaluate the errors takes a significant amount of time: with the maximum number of errors execution takes 15% longer than it does when there is only one error in each codeword, and energy consumption grows similarly, as shown in Table II.

In summary, each computational component has a characteristic effect on the execution time and energy consumption. The performance of all the components is dependent on the encoding scheme. The number of parity bits has a linear relation to the execution time and the energy consumption of computational components C_1 and C_2, whereas the time and energy used by C_3 are both proportional to the number of parity octets and to the number of errors.

IV. Energy consumption by RS decoding in a simulated channel

In this section, we examine the energy consumed during the decoding of RS codes which have 32 octets in a codeword (32-series codes). We will set the number of packets in each ECB row (M) to 16, so as to maximize error recovery capacity (each packet contains 125 bytes, so there are 125×16 codewords per ECB). We derive the average total energy required to decode a data stream of b_i Kbps, for t seconds continuously with varying input BERs (ε) and mobile speeds ($f_d T$).

A. Performance analysis of the RS decoder

1) Channel model : In this study, we used the Gilbert channel model [11] [12] to simulate the behavior of 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 over a correlated Rayleigh fading channel: these errors occur in clusters or bursts with relatively long error-free intervals between them.

By choosing different values for the input bit error-rate and for $f_d T$ (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 [13]), we can model different degrees of correlation in the fading process. The value of $f_d T$ determines the correlation properties, which are related to the mobile speed for a given carrier frequency. When $f_d T$ is small, the fading process is strongly auto-correlated, which means long bursts of errors (slow fading). Conversely, the errors are weakly auto-correlated for large value of $f_d T$ (fast fading). In the following experiments, we used values of 0.0001 (s_1) and 0.00005 (s_2) for $f_d T$, which correspond to fast and moderate fading conditions respectively, with a reference channel data rate of 409.6 Kbps, and a carrier frequency of 900MHz.

In the equations that are to follow, α is the probability that the i^{th} bit is corrupted, given that the $(i-1)^{\text{th}}$ bit is transmitted successfully, and β is the probability that the i^{th} bit is successful, given that the $(i-1)^{\text{th}}$ bit was unsuccessful. The steady-state error rate ε is then obtained as follows:

$$\varepsilon = \frac{\alpha}{\alpha + \beta} \; . \quad (4)$$

If the Rayleigh fading margin is F, the average bit error-rate

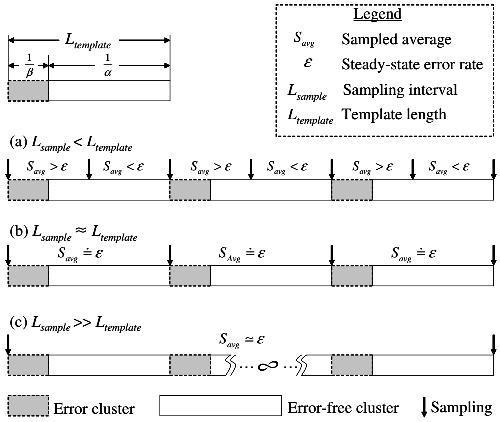

Fig. 5. The effect of sampling interval.

can be expressed as

$$\varepsilon = 1 - e^{-\frac{1}{F}} \; . \quad (5)$$

Using Eq. 4 and the equations which follow, we can now derive values for α and β. The average number of consecutive bit errors is given by $1/\beta$, where

$$\beta = \frac{Q(\theta, \rho\theta) - Q(\rho\theta, \theta)}{e^{\frac{1}{F}} - 1}, \text{ and } \theta = \sqrt{\frac{2/F}{1-\rho^2}} \; . \quad (6)$$

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,

$$Q(x, y) = \int_y^{\infty} e^{-\frac{x^2 + w^2}{2}} I_0(xw) w \; dw \quad (7)$$

is the Marcum-Q function. Thus, the relationship between bit error-rate and the Markov parameter can be expressed as

$$\beta = \frac{1-\varepsilon}{\varepsilon} [Q(\theta, \rho\theta) - Q(\rho\theta, \theta)] \; , \quad (8)$$

where

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

2) Sufficient interleaving by the RS ECB: The BCMCS system varies the size of the ECB to scatter error clusters into a sparse pattern so as to maximize the error recovery performance of the Reed-Solomon decoder. We suggest that, if sufficient interleaving space is provided by the RS codes, the Rayleigh distribution of an error cluster is converted into a random one. We will now present a model to support this contention.

Fig. 5 shows the relation between the sampling interval (L_{sample}) and the sampled average value (S_{avg}). As mentioned in the previous section, the average lengths of sequences of uncorrupted and error bits are $1/\beta$ and $1/\alpha$ respectively. The pattern of fluctuation in the channel condition is made up

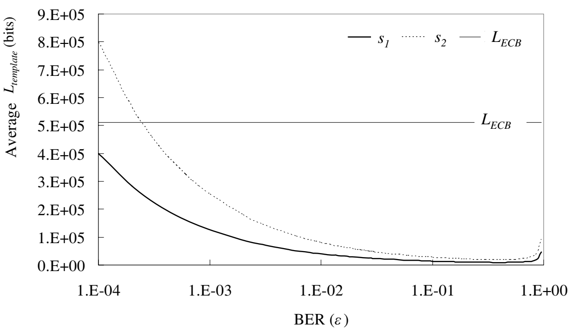

Fig. 6. Average $L_{template}$ and L_{ECB}.

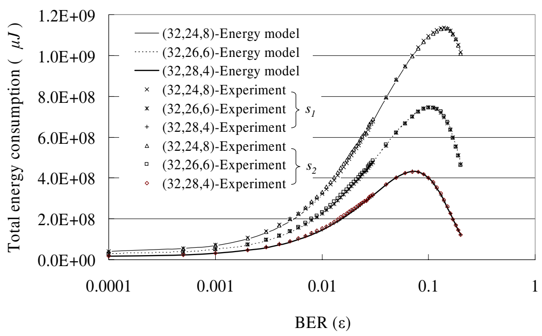

Fig. 7. Total energy consumption to decode a 1-hour 100Kbps video clip.

of a repetition of those error and normal bit clusters, and we will call the average length of one sequence of normal bits and error bits a template ($L_{template}$), which can be expressed as follows:

$$L_{template} = \frac{1}{\alpha} + \frac{1}{\beta} = \frac{1}{(1-\varepsilon)[Q(\theta,\rho\theta) - Q(\rho\theta,\theta)]}. \quad (9)$$

We now consider the relative lengths 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 those samples will be exaggerated. The other samples will contain relatively few errors, and for these S_{avg} will naturally be underestimated.

As L_{sample} converges to $L_{template}$, 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} saturates to ε as L_{sample} grows to ∞.

In BCMCS, the errors in each channel are interleaved by making the ECB larger (which also increases the value of M). The size of the ECB (L_{ECB}) can be considered as equivalent to L_{sample}, and is defined as follows:

$$L_{ECB} = (M \times 125) \times N \times \frac{bits}{octet}, \quad (10)$$

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

Fig. 6 shows the theoretical variation of $L_{template}$ with the input BER (ε), and the corresponding curve for L_{ECB} when M is 16 and N is 32. The value of $L_{template}$ for a mobile station moving with a speed s_1 is smaller than the size of the ECB (L_{ECB}), for BERs 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. 5 (b) or (c), which is the result of sufficient interleaving. However, when the BER is lower than 2.4×10^{-4}, the value of $L_{template}$ for a mobile moving at a speed of s_2 is larger than L_{ECB}. This corresponds to a situation similar to that shown in Fig. 5 (a). But, at both speeds, the ECB interleaves the error cluster sufficiently when the value of ε is more than 2.4×10^{-4} and M is 16.

From these results we can see that, under most channel conditions, the BCMCS system can select an ECB size that will achieve sufficient interleaving to randomize the influence of the channel on the error distribution, and from now on we will assume that sufficient interleaving is indeed being achieved.

3) Modeling recovery rate and energy metrics under sufficient interleaving: Using 32-series (N, K, R) RS codes, $M = 16$, and assuming sufficient interleaving, the probability that the codeword contains ν errors, and the residual error rate after RS error recovery ($\varepsilon_{residual}$) can be expressed as follows:

$$P(\nu|\varepsilon) = \binom{N}{\nu}(\varepsilon)^{\nu}(1-\varepsilon)^{N-\nu} \quad (11)$$

$$\varepsilon_{residual} = \sum_{\nu=R+1}^{N} \frac{\nu}{N} \times P(\nu|\varepsilon). \quad (12)$$

Thus the error *recovery rate*, which is an indicator of performance, can also be derived as follows:

$$\textit{recovery rate} = 1 - \frac{\varepsilon_{residual}}{\varepsilon}. \quad (13)$$

If a mobile receives a t-second video clip with a data rate of b_iKbps, the number of codewords, $N(\nu)$, that contains ν octet errors is defined as follows:

$$N(\nu|\varepsilon, b_i, t) = P(\nu|\varepsilon) \times \frac{b_i(Kbps) \times t(seconds)}{K \times (bits/octets)}. \quad (14)$$

And the expected energy consumption of RS decoding for a given steady-state error rate of ε can be expressed as:

$$E[energy|\varepsilon, b_i, t] = \sum_{\nu=0}^{N} E_{codeword}(\nu) \times N(\nu|\varepsilon, b_i, t), \quad (15)$$

This equation allows us to determine the total energy requirement for decoding a given payload.

B. Measurement of energy consumption with a simulated channel

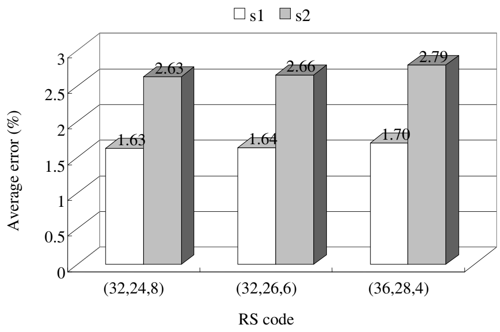

Fig. 8. Average error between the results from the energy model and from experiment.

1) Verification of our energy model: Fig. 7 shows theoretical and simulated results for the total energy required by the RS decoder to service a 1-hour video clip arriving at 100 Kbps, under BER conditions from 0.0001 to 0.2. Experimental values of energy consumption ($E_{exp.}$) follow the theoretical curves closely: the agreement for a mobile moving quickly (at speed s_1) is excellent, and results for a slower mobile (moving at speed s_2) are also very close. The average error of our energy model can be expressed as follows:

$$\text{Average error} = 100 \times \text{Avg}\left[\frac{E_{exp.|\varepsilon,b_i,t} - E[energy|\varepsilon,b_i,t]}{E_{exp.|\varepsilon,b_i,t}}\right]_{\varepsilon_{run}}, \quad (16)$$

where the experimental run of values of ε is

$$\varepsilon_{run} = \{0.0001, 0.0005, 0.001, 0.002, \ldots, 0.03, 0.04, \ldots, 0.2\}.$$

The error values that we obtained in this way are all less than 3%, as shown in Fig. 8.

2) Analysis of the energy consumed by the RS decoder in a simulated channel: The results presented in Fig. 7 show two significant trends. First, the energy consumption increases with the amount of parity information in the RS codes, as we explained in Section III-A.2. Second, as the parity overhead increases, the effective data rate is reduced so that more codewords are needed to handle the same data payload. As a result, it takes more time to process the data and more energy is consumed. The observation that RS codes with less parity consume less energy suggests that an appreciable amount of energy can be saved by choosing the slimmest RS code that still guarantees the bit-level QoS requirement of the video application.

We also see that the RS decoding process uses more energy as the input BER increases. This is a plausible result because the error correction routine has to run more frequently if more errors are input to the RS decoder. However, the energy required for RS decoding actually decreases as ε is incremented past a certain value, which depends on the RS code. This occurs because the RS decoder gives up error correction if the BER of the channel is sufficiently high, and simply forwards the erroneous codeword. When there is little parity information in each codeword, error correction is abandoned relatively early (at $\varepsilon = 0.07$ with (32,28,4) code, for instance), and thus no further energy is used by the computational components responsible for error correction (C_2 and C_3).

V. Conclusions

We have investigated and analyzed the energy consumption of mobiles receiving high-speed broadcast services in a 3G cellular network, focusing on error recovery by the Reed-Solomon decoder that operates in the MAC layer. We have also proposed an analytic energy model which we verified by extensive simulation. The energy consumption of the RS decoding process is mainly determined by three computational components: the first of these decides whether to try erasing the errors or not, and the other two are the error locator and the erasure decoder, which find the locations of any errors and correct them if they exist. The energy consumed by these computational components was measured by running the RS decoding process on an ARM7TDMI testbed while deploying a web-based energy exploration tool (the SEE).

The results presented in this paper will help network operators to configure service parameters, such as RS code and ECB size, so as to achieve a broadcast service that is energy-efficient at the mobiles while guaranteeing the required quality of service.

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] 3GPP2. Broadcast multicast services framework. Standard X.P0019 v0.1.3, 3GPP2, 2003.

[4] J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, E. Tiedeman, and QUALCOMM. Broadcast and multicast services in cdma2000. *IEEE Communications Magazine*, 42(2):76–82, February 2004.

[5] K. Kang, J. Cho, Y. Cho, and H. Shin, "Dynamic scheduling for scalable media transmission over cdma2000 1xEV-DO broadcast and multicast networks," *Lecture Notes in Computer Science*, vol. 3462, Springer, pp. 968-979, April 2005.

[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 Communication Magazine*, pp.70–77, July 2000.

[7] 3GPP2. cdma2000 high rate packet data air interface specification Standard C.S0024-0 v4.0, 3GPP2, 2002.

[8] 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 Design and Test of Computers*, 21(6):572–586, November-December, 2004.

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

[10] Y. Cho, K. Kang, J. Lee, and H. Shin. Proactive Reed-Solomon bypass (PRSB): a technique for real-time multimedia processing in 3G cellular broadcast networks. *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 532–538, August 2005.

[11] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1235-1266, September 1960.

[12] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468-1477, November 1998.

[13] W. C. Jakes, *Microwave Mobile Communications*, Wiley, 1974.