Adaptive Error Recovery in cdma2000 1xEV-DO Mobile Broadcast Networks

Kyungtae Kang, Yongwoo Cho, Hosang Park, and Heonshik Shin

School of Electrical Engineering and Computer Science
Seoul National University, Seoul, 151-744, Korea
E-mail: {ktkang, xtg05, phosanna, shinhs}@cslab.snu.ac.kr

Abstract. We analyze the performance of MAC-layer Reed-Solomon error-recovery in the cdma2000 1xEV-DO Broadcast and Multicast Services (BCMCS) environment, with respect to the size of the error control block (ECB) and the air-channel condition, and establish the relationship between ECB size, error-recovery capacity and service latency. Real-time traffic, such as voice and video streaming, is very sensitive to delay, but can stand a certain level of packet loss. We therefore propose an adaptive error-recovery scheme which adjusts the size of the ECB to reflect the environment of the mobile nodes so as to meet the required service quality (target bit error-rate), while reducing the latency of real-time applications, decoding complexity, and memory requirement. Extensive simulation results show the effectiveness of our adaptive approach compared to the current static scheme. Our scheme achieves near-optimal service latency while meeting the target service quality criterion, and also reduces energy consumption during error recovery.

Keywords: cdma2000 1xEV-DO BCMCS, Adaptive error-recovery, Reed-Solomon, error control block, service latency.

1 Introduction

Work has begun, in both the Third Generation Partnership Project (3GPP) and the 3GPP2 group, on enhancing 3G networks to support broadcast and multicast services (BCMCS) [1] [2] [3]. The 3GPP2 group recently baselined the specification for the cdma2000 high-rate broadcast packet-data air interface [4] [5]. In order to provide a high-speed broadcast service in the BCMCS environment, the efficient handling of data transmitted over wireless radio channels must be considered.

In general, 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. BCMCS therefore use Reed-Solomon [6] coding as a forward error-correction scheme, which can efficiently conceal error clusters. Usually, the Reed-Solomon scheme is combined with an appropriate data interleaving mechanism to increase performance. In BCMCS, this interleaving is controlled by the size of the ECB (error control block). Bursty errors can be corrected much more efficiently with a larger ECB. But, as the size of the ECB increases, mobile node have to buffer a lot of data. This delays the applications that access this data, and increases the memory required at the mobile node and the computational complexity of the decoding process [4]. Reducing delay is very important for real-time applications, such as voice and video streaming, but a certain level of packet loss is tolerable. On the basis of extensive simulation, we derive the performance of Reed-Solomon coding with respect to error capacity and service delay as the size of the ECB changes, within the current cdma2000 1xEV-DO broadcast environment.

Additionally, we propose an adaptive error-recovery scheme for cdma2000 1xEV-DO broadcast networks. Our scheme varies the ECB size to suit changing channel conditions at the mobile nodes. Its aim is to achieve the smallest ECB necessary to meet the required level of service while controlling the latency. The condition of the stationary channel, which drives adjustments to the ECB, is obtained from heuristics which use a moving average (MA) or a weighted average (WA) to flatten out large fluctuations in the bit error-rate.

2 Background

2.1 Error Recovery in Current BCMCS

Unlike the unicast cdma2000 1xEV-DO standard [7], BCMCS do not use an error-control scheme based on ARQs (automatic repeat requests), because there is no reverse link to carry the ACK/NAK signal from the access terminal to the access network. Instead, error control is provided by a forward error-correcting product code, comprising an inner turbo code and an outer Reed-Solomon code. 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 specifies an additional 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 provides the channel structure for the broadcast channel [4] [5].

Each logical channel uses error control blocks (ECBs) with M MAC packets per ECB row. The variables N and K represent the number of octets and security-layer octets in a Reed-Solomon code word. R is the number of parity octets: the Reed-Solomon decoder can recover up to R octet erasures in each code word. 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. The ECB is designed to provide a structure such that, in the event of a physical-layer packet erasure, octets in the same position are lost from all affected Reed-Solomon code words. 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,

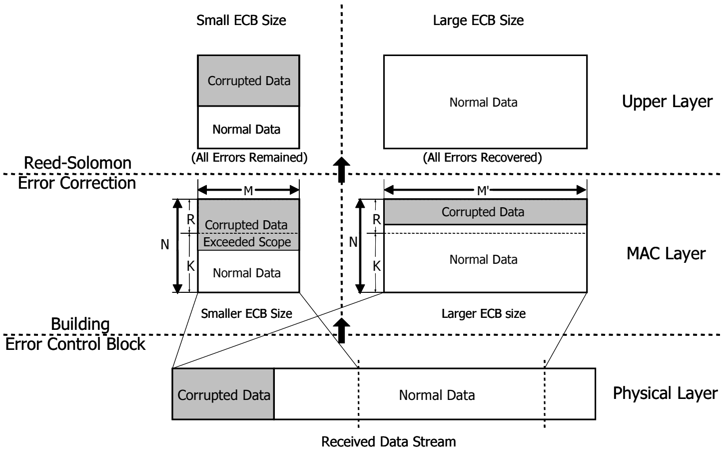


Fig. 1. Correlation between ECB size and error-recovery capacity.

and K can take a value of 28, 26 or 24 when $N = 32$, or a value of 14, 13 or 12 when $N = 16$ [4] [5].

One of the most significant environmental factors affecting channel condition is fading. This is correlated with the burstiness of errors. In slow-moving conditions, the error occurrence pattern tends to be more bursty than in fast-moving conditions. A Reed-Solomon code of (N, K, R) cannot recover any lost data if the corrupted portion of a code word 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. This situation is shown schematically in Fig. 1. 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 $M \times 125$ octets. The value of M for a given ECB has to be less than or equal to 16. As the value of M increases, the time-diversity also increases and thus a mobile node which is in a time-varying shadow environment is able to recover more corrupted data. However, the amount of storage required at the mobile node also increases. Therefore, the value of M is an important consideration in achieving better error-recovery capacity for mobile nodes with less system resource.

3 Proposed Adaptive Error Recovery Scheme

As explained in the previous section, the bit error-rate after Reed-Solomon decoding can be controlled by adjusting the value of M. In current BCMCS, this value is fixed, generally to 16, to increase the error recovery capacity without considering the service delay. However, we can reduce the average service delay by dynamically reconfiguring the Reed-Solomon ECB according to the channel condition of the mobile nodes. Ideally, adapting the value of M to the current

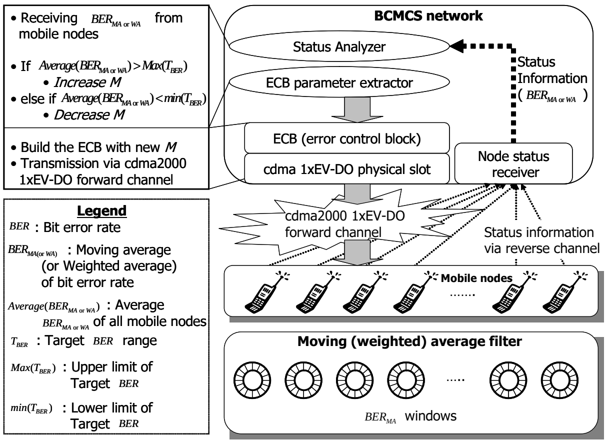


Fig. 2. Service implementation scenario.

network environment in this way would provide a full solution, reducing the service latency while guaranteeing the target bit error-rate. However, in reality it is not possible for the network to sense the environment around the mobile nodes directly. Instead, to detect the tendency of the current environmental status of the mobile nodes, we use a feedback mechanism in which each node reports its bit error-rate as an indicator of its receiving channel environment. Because the status of each mobile node depends on its previous situation, prediction of subsequent status can be achieved by the heuristic feedback algorithm which we will go on to describe.

Fig. 2 shows the service scenario with which we have experimented. The BCMCS network broadcasts a data stream to the mobile nodes via the cdma2000 1xEV-DO forward traffic channel. Each mobile node records its error count at a predefined interval as a BER_{MA} or BER_{WA} (moving average or weighted average of the bit error-rate), implemented as a circular queue. Each node periodically reports the moving average (MA) of the bit error-rate at a window, or the weighted average (WA), to the network via the cdma2000 1xEV-DO reverse channel. The network then selects a new value of M on the basis of the average of the status information collected from the nodes. An $Average(BER_{MA})$ or $Average(BER_{WA})$ larger than the maximum boundary of the target BER range ($Max(T_{BER})$) suggests that the environmental status in the current service area is unsatisfactory. In this case, the network increases M in order to improve its capability to conceal errors. Conversely, an $Average(BER_{MA})$ or $Average(BER_{WA})$ that is lower than $Min(T_{BER})$ indicates that the current network status is more than sufficient for the required level of service, and that M can be reduced so as to reduce the total latency of service, the memory required and the computational complexity. The network then builds a new ECB in the MAC layer of cdma2000 1xEV-DO, using the adjusted value of M. The

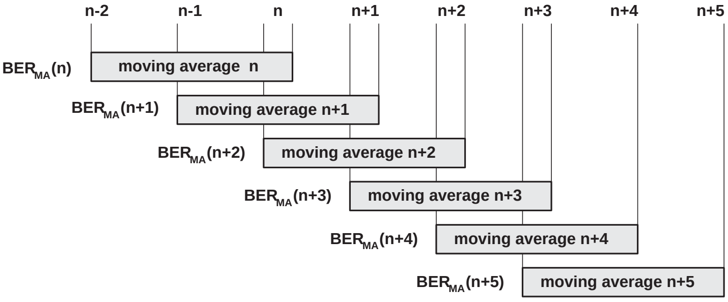


Fig. 3. Simple moving average.

data-stream built by this procedure is transferred across the cdma2000 1xEV-DO forward channel. By repeating these procedures, the network can adapt its transmission configuration to suit the environmental situation.

To control the size of the ECB according to the average channel condition of mobile nodes, we employ the MA as a low-pass filter, and its output is used to determine the appropriate ECB size. By damping changes in the BER, it can be used to determine the channel condition more reliably. Fig. 3 shows the basic idea of an MA with a given window size w, which is calculated as follows:

$$\begin{aligned} BER_{MA}(\tau_j, n) &= \frac{1}{w} \sum_{i=n-w+1}^{n} BER(\tau_j, i) \\ Average(BER_{MA}) &= \frac{1}{\eta} \sum_{j=0}^{\eta-1} BER_{MA}(\tau_j, n) \\ BER(\tau_j, i) &:= i^{th} \text{ BER of } \tau_j \\ BER_{MA}(\tau_j, n) &:= n^{th} \text{ moving average of } \tau_j \\ Average(BER_{MA}) &:= \text{average } BER_{MA} \text{ of all mobile nodes} \\ \eta &:= \text{number of mobile nodes.} \end{aligned} \tag{1}$$

Each node feeds back recent information about its environmental condition, measured as a value of BER_{MA}, to the network, which then determines the general status of the mobile nodes by averaging the BER_{MA} values. If the general status of the network environment is getting worse, then M will be increased; if the situation takes a favorable turn, then M will be reduced so as to shorten delays in the system. However, it takes a long time for the value of M to decrease once it has increased following the deterioration of the channel condition, and in the meantime the service latency will become longer. We therefore deploy another method, a weighted average, which works to reduce a large value of M more quickly, by giving more priority to the tendency of the recent BER values reported by the mobile nodes. This weighted average (WA) can be calculated as follows:

$$\begin{aligned}
BER_{WA}(\tau_j, n) &= \omega \cdot BER_{WA}(n-1) + (1-\omega) \cdot BER(\tau_j, n) \\
Average(BER_{WA}) &= \frac{1}{\eta} \sum_{j=0}^{\eta-1} BER_{WA}(\tau_j, n) \\
BER(\tau_j, i) &:= i^{th} \text{ BER of } \tau_j \\
BER_{WA}(\tau_j, n) &:= n^{th} \text{ weighted average of } \tau_j \\
Average(BER_{WA}) &:= \text{average } BER_{WA} \text{ of all mobile nodes} \\
\omega &:= \text{weight.}
\end{aligned} \tag{2}$$

4 Performance Evaluation

4.1 Simulation Environment

In our experiments, we used QPSK modulation with a 1228.8 kbps data rate forward channel. We also used the simple threshold model suggested by Zorzi [8] to simulate the behavior of errors which arise in data transmission over fading channels. This model can be represented as a binary Markov process [9] 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.

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 values of f_dT (fast fading). A value of f_dT of 0.00001 is taken to correspond to slow fading; and values of 0.00002 and 0.00003 are taken to correspond to fast fading. In computing the MA and WA, we set the values of $Min(T_{BER})$ and $Max(T_{BER})$ to 1% and 3% respectively, the size of the window to 10, and the weight (ω) to 0.7. We varied the value of $\varepsilon_{physical}$, while fixing the RS code at (16,12,4). These settings were then applied to simulation scenarios in which ten mobile nodes move around a service area in a pattern that yields values of f_dT between 0.00001 and 0.00003.

4.2 Experimental Results

Performance Analysis of Reed-Solomon Coding with respect to M Our first experiment investigated the relationship between the average value of

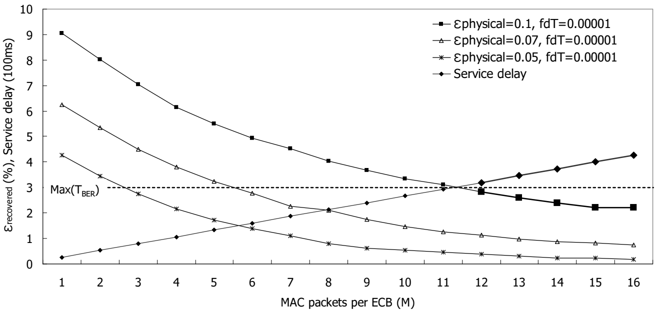


Fig. 4. Average $\varepsilon_{recovered}$ when $f_dT = 0.00001$.

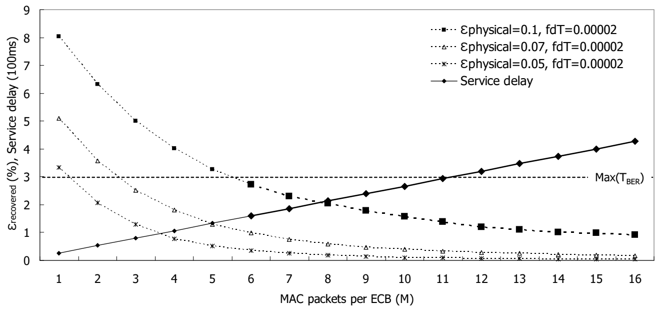


Fig. 5. Average $\varepsilon_{recovered}$ when $f_dT = 0.00002$.

$\varepsilon_{recovered}$ and the number of MAC packets per ECB (M) in a cdma2000 1xEV-DO broadcast environment. We will use $\varepsilon_{physical}$ to denote the physical-layer bit error-rate, and $\varepsilon_{recovered}$ to denote the bit error-rate of data carrying (excluging parity-carrying packets) after Reed-Solomon decoding. For each $\varepsilon_{recovered}$ and f_dT, the average $\varepsilon_{recovered}$ using Reed-Solomon coding is inversely proportional to M. Additionally, the rate of reduction of the average value of $\varepsilon_{recovered}$ is greater in fast fading, because the error bursts are shorter than they are in slow fading. Bursts of errors can be sufficiently interleaved in an ECB when their lengths are short, and thus the average $\varepsilon_{recovered}$ increases considerably with even a small increase of M. Because M is related to service latency and the memory requirement, both latency and storage must be sacrificed to reduce the average value of $\varepsilon_{recovered}$. The service delay increases linearly with the size of the ECB, as shown in Figs. 4, 5 and 6.

However, if we abandon a fixed value of M, and instead control M dynamically to suit the channel condition of the mobile nodes, while also satisfying the target bit error-rate, then we can reduce the average $\varepsilon_{recovered}$ with relatively little service delay. For example, a target BER ($Max(T_{BER})$) can be satisfied if the value of M is more than 12, as represented by the bold line in Fig. 4. There-

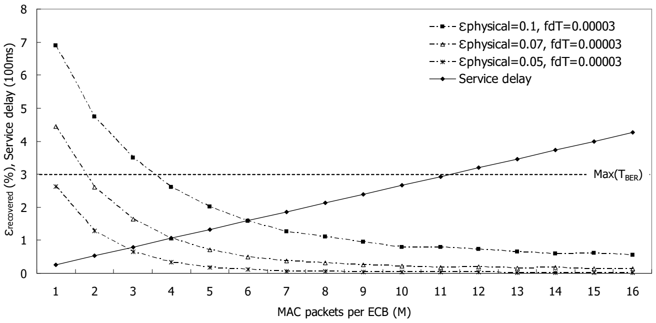


Fig. 6. Average $\varepsilon_{recovered}$ when $f_dT = 0.00003$.

fore, if we set M to 12, we can minimize the service delay while guaranteeing the required service quality. When $M = 12$, the service delay is about 100ms less than it is when $M = 16$. Another example, in Fig. 5, shows that $\varepsilon_{recovered}$ is less than $Max(T_{BER})$ when M is more than 6. Thus, by constructing an ECB with M set to 6, we can again minimize the service latency while still achieving the required quality. This is especially important for real-time applications in which a short service delay is especially desirable. By dynamically optimizing the size of the ECB (selecting the minimum M which satisfies the target service quality), we can approach the ideal solution. As mentioned earlier, we propose two heuristic methods to achieve this, based on the MA or the WA, and the performance of these two adaptive schemes is experimentally evaluated in the following section.

Performance of the Proposed Adaptive Scheme This experiment was conducted to show that our adaptive schemes, using either the MA or the WA method to adjust M, achieve results close to the ideal solution. This 'ideal solution' is computed using a minimum value of M chosen so that the average value of $\varepsilon_{recovered}$ corresponds to T_{BER} even though the channel condition changes.

Figs. 7 and 8 show that the value of M in our proposed scheme follows the values arrived at by the ideal solution. From Fig. 7, we can see that the MA scheme requires more time to accommodate to changes in channel condition than the ideal solution. However we see from Fig. 8 that M decreases as quickly as it does in the ideal solution, because the WA method adapts to a more recent channel condition of the mobile nodes. A moving average is a statistical technique used to predict the tendency of a stochastic process, and tends to eliminate short-term fluctuations in a time series and to highlight long-term trends. Thus an MA provides an effective way to predict the occurrence of errors in wireless communication, provided that they have a continuous stochastic pattern. Conversely, a weighted average emphasizes recent movements rather than overall low-frequency trends; it is sensitive to high-frequency fluctuations and is

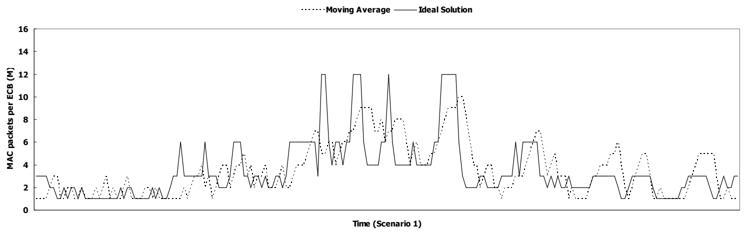


Fig. 7. MAC packets per ECB (M) with the MA method and the ideal solution.

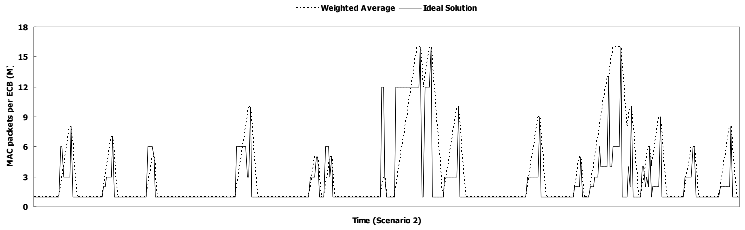
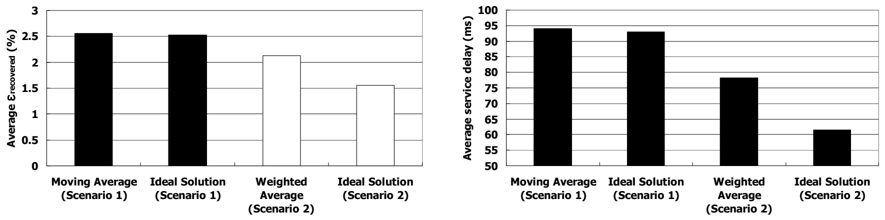


Fig. 8. MAC packets per ECB (M) with the WA method and the ideal solution.



(a) Average $\varepsilon_{recovered}$.

(b) Average service delay.

Fig. 9. Average $\varepsilon_{recovered}$ and service delay of all mobile nodes.

therefore more agile. However, the methods will tend to converge if the recent weight of the WA and the window period of the MA are reduced.

Fig. 9(a) shows that the average value of $\varepsilon_{recovered}$ achieved by our proposed schemes and by the ideal solution both satisfy the target BER. In Fig. 9(a), the average value of $\varepsilon_{recovered}$ is lower than that of $Max(T_{BER})$, and our scheme's average $\varepsilon_{recovered}$ is close to that for the ideal solution. Fig. 9(b) shows the average service delay generated by the proposed adaptation methods and by the ideal solution. We can see from this figure that our proposed heuristic methods provide a near-optimal reduction in service delay, without compromising quality. The differences in average $\varepsilon_{recovered}$ and average service delay between the MA and WA methods stem from the different experimental scenarios.

5 Conclusions

The Reed-Solomon error-correction scheme uses data interleaving mechanisms to increase error-recovery performance. This can be achieved in current BCMCS by adjusting the size of the error control block. By increasing the size of the ECB, we can recover from bursty errors efficiently; however, a larger ECB means increased service latency, an increased memory requirement, and increased computational complexity. To deal with this problem, we adapt the size of the ECB to the value of T_{BER} as the channel condition of the mobile nodes varies. This has been shown to reduce the overall average service delay. We adopted a heuristic feedback mechanism combined with a moving or weighted average to generate a value of M which represents the size of the ECB. This gives simulation results which are as good as those from an ideal solution. To determine the ECB size, our MA method uses an average value of $\varepsilon_{recovered}$ for the mobile nodes obtained from a window of size w, while our WA method uses a statistical approach which reflects more recent changes in $\varepsilon_{recovered}$. We have observed that the these approaches to the dynamic variation of ECB size reduce average service delay to a near-optimal level, while maintaining $\varepsilon_{recovered}$ within T_{BER} and reducing the memory requirement. This is a great improvement on the use of a fixed-size ECB.

References

1. J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, E. Tiedemann, and QUALCOMM, "Broadcast and multicast services in cdma2000," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 76-82, February 2004.
2. 3GPP2 X.P0019 v0.1.3, "Broadcast-multicast services (bcmcs) framework draft doument," August 2003.
3. K. Kang, Y. Cho and H. Shin, "Performance analysis of dynamic packet scheduling within a CDMA2000 1xEV-DO Broadcast Network," *Proc. of IEEE Vehicular Technology Conference*, vol.4, pp.2586-2590, September 2005.
4. P. Agashe, R. Rezaiifar, P. Bender, and QUALCOMM, "Cdma2000 high rate broadcast packet data air interface design," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 83-89, February 2004.
5. 3GPP2 C.S0054 v1.0, "Cdma2000 high rate broadcast-multicast packet data air interface specification," February 2004.
6. R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.
7. 3GPP2 C.S0024 33.0, "Cdma2000 high rate packet data air interface specification," December 2001.
8. 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.
9. H. S. Wang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Transactions on Vehicular Technology*, vol. 45, pp. 353-357, May 1996.