End-to-End Delay Analysis of Wireless ECG over Cellular Networks

Man-Ki Yoon, Jung-Eun Kim, Kyungtae Kang, Kyung-Joon Park, Min-Young Nam, and Lui Sha
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{mkyoon, jekim314, ktkang, kjp, mnam, lrs}@uiuc.edu

ABSTRACT

In this paper, we present a medical-grade network architecture based on the wireless cellular technology of CDMA2000 1xEV-DO (Evolution-Data Optimized). It can provide highly reliable communication with a bounded delay by exploiting its inherent channel access structure and cooperative packet scheduler at the access network. Additionally, we propose a way of analyzing the worst-case end-to-end delay over the candidate cellular architecture, and apply it to wireless ECG (Electrocardiogram) to examine its applicability. Our simulation study shows how the proposed architecture can provide acceptable quality of service for wireless medical applications.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design Studies
; C.4 [**Performance of Systems**]: Reliability, availability, and serviceability
; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Design, Performance, Verification

Keywords

Wireless medical application, end-to-end delay analysis, wireless cellular network, CDMA 1xEV-DO, wireless ECG

1. INTRODUCTION

Recently, the design paradigm of medical systems is shifting from wired to wireless owing to increased convenience and mobility support of wireless technologies. With this increasing demand, IEEE 802.1x-based networks are being deployed as wireless medical networks [1, 3, 9].

WiMD'09, May 18, 2009, New Orleans, Louisiana, USA.

Figure 1: Wireless ECG [2]

On the other hand, wireless cellular technologies, such as CDMA2000 1xEV-DO, have not been paid much attention as a wireless medical network regardless of its support for high mobility and superior security. Since wireless cellular networks are already widely deployed, it will be of great convenience to utilize the infrastructure for medical applications. Most of all, the cellular technologies can provide highly reliable communication while bounding the end-to-end delay, which is one of the most important QoS (Quality of Service) metric.

In this paper, we consider network architecture for wireless cellular technology as a viable solution to the wireless medical environment. In this architecture, we adopt a forward error correction (FEC) mechanism, mostly used in broadcasting due to its good predictability of delay bound, based on Reed Solomon (RS) coding to support reliable communication. Furthermore, we introduce the Advanced Encryption Standard (AES) to the security layer to protect the health-related data from security threats[1]. Then, we propose an end-to-end delay analysis technique for wireless medical applications by taking into account FEC and AES together. As a case study, we apply our framework to wireless ECG. Our simulation study shows that the proposed architecture is a promising solution for wireless medical networks.

Although there have been some recent studies on wireless medical technologies, to the best of our knowledge, there has been little work on the detailed design and analysis of wireless cellular technology as a network architecture for medical applications.

The remainder of this paper is structured as follows: The

[1]Both usage of the RS and the AES are defined in 3GPP2 standard[11, 12].
[2]The figure is referenced from www.dcontinuum.com.

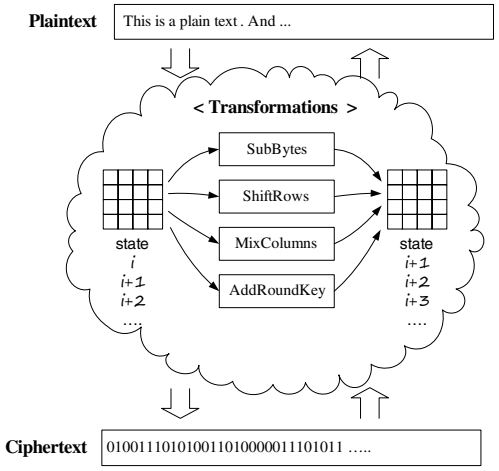

Figure 2: General Concept of AES

next section summarizes the background knowledge. In Section 3, we propose a medical network architecture based on the wireless cellular technology. Section 4 presents a worst-case end-to-end delay analysis technique for wireless medical applications. In Section 5, we perform a delay analysis of wireless ECG to examine the applicability of the proposed architecture. Finally, Section 6 concludes the paper.

2. BACKGROUND

Electrocardiogram (ECG) is a recording of electrical impulses in the heart for the purpose of detecting abnormal activity of the heart. Those electrical impulses are measured from several leads, called *electrode*, and displayed in voltage difference between each pair of leads. In wireless ECG applications, the patient's heart-beat data, represented by electrical signals, are continuously transmitted to ECG monitoring application through wireless network. As an example of data information of ECG application, [5] recommended the number of leads as from 2 to 32, each of which takes samples of 8, 16, or 32 bits size at 200Hz $\sim$ 500Hz. Additionally, the allowable end-to-end delay of a wireless ECG application should be less than few seconds, e.g. 2 secs.

The Advanced Encryption Standard (AES) is a block cipher, which takes a fixed length of input, called *block*, and produces a corresponding encrypted output block of the same length. To be specific, the AES encrypter repeatedly performs a certain number of transformation rounds that converts an intermediate result, called *state*, into the next state, as illustrated in Figure 2. Each round consists of four operations; SubBytes, ShiftRows, MixColumns, and AddRoundKey transformation. In the *SubBytes* transformation, every byte in the state is substituted by new byte value using the Rijndael substitution table, *S-Box*. The *ShiftRows* transformation operates on each row of the state so that each row is cyclically shifted to the left in encryption, or right in decryption. In this process, each n^{th} row is shifted $(n-1)$ bytes to the left or right. In *MixColumns* transformation, each column of the state is considered as a polynomial

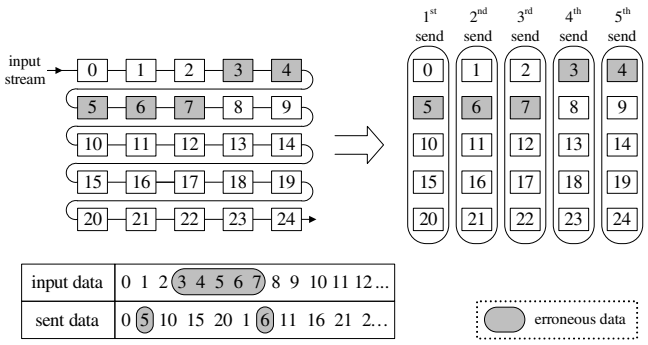

Figure 3: Block Interleaving in RS Coding

over $GF(2^8)$ and is multiplied with a special matrix, and then the result of this matrix multiplication replaces the original column. In the *AddRoundKey* transformation, each byte of the state is combined with the corresponding byte of the round key, which is a cipher key derived from the main key by using the key schedule, using bitwise XOR[3].

Forward error correction (FEC) is an error detection and correction code to improve reliability of communication. As the name implies, decoder at the receiver side can correct as well as detect errors without requesting retransmission of data by adding redundant bits to the original data by the encoder, which means that it can provide bounded delay. Thus, it is suitable for transmitting continuous streaming data which requires seamless services like medical applications.

Reed Solomon (RS) coding [2] is a block-based error correction code and is used as an outer FEC coding. This RS coding is combined with block interleaving scheme to make the bursty errors looks like sparse errors as illustrated in Figure 3. Thus the RS coding is suitable for wireless communication network in which errors occur as bursty. Meanwhile, increasing the interleaver's width makes bursty errors be more sparse, that is, it is more easier to correct those errors. However, it requires more time for memory management and therefore increase total delay. Accordingly, there is a trade-off between reliability and latency.

An RS code is specified by a tuple (N, K, R) where each parameter in the tuple is defined as follows :

N = the maximum length of a codeword in symbols

K = the data length in symbols

R = the parity length in symbols

RS decoder can correct not only up to $R/2$ errors but also up to R erasures. Since the demodulator at the physical layer sets a flag indicating the locations of erroneous symbols, RS decoder can easily detect those symbols and therefore correct them. In this paper, thus, we assume that we use the RS erasure decoder where the error correction ability is R.

[3]The detailed procedures of AES can be found in [4].

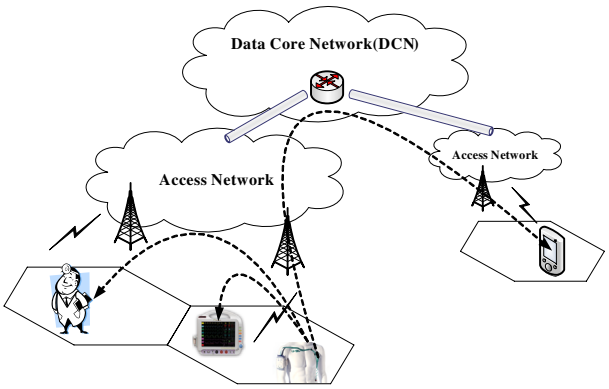

Figure 4: Network Architecture for Wireless ECG

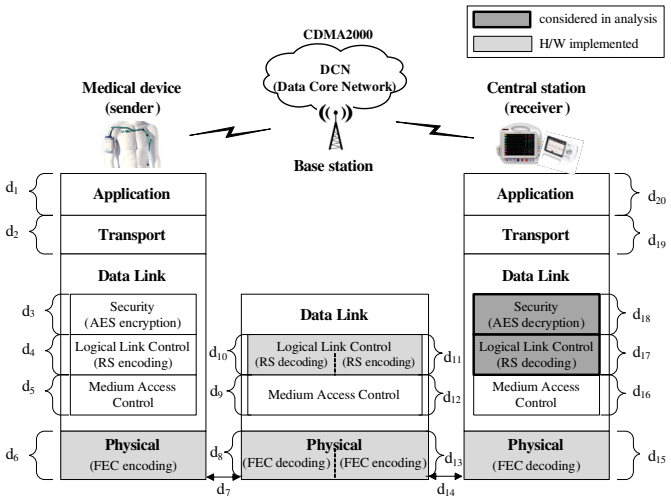

Figure 5: Protocol Stacks for Medical Application

3. NETWORK ARCHITECTURE FOR WIRELESS MEDICAL APPLICATIONS USING WIRELESS CELLULAR TECHNOLOGY

Wireless cellular technologies such as CDMA2000 1xEV-DO, HSDPA (High-Speed Downlink Packet Access) and OFDMA (Orthogonal Frequency-Division Multiple Access) can be a good candidate for reliable wireless medical network since these exploit TDM (Time Division Multiplexing) in combination with CDMA or OFDMA, so that these can deterministically bound the worst-case transmission delay. In addition, these technologies have high mobility with offering appropriate data rate for medical applications. Thus, medical devices constituting a medical application can do reliable communication while moving in medical circumstances by using wireless cellular technologies.

Thus, in this section, we consider an abstracted wireless cellular network architecture for reliable wireless medical application. This architecture consists of the wireless communication network using cellular technology, especially CDMA 2000 1xEV-DO, and medical devices which communicate with each other via such wireless cellular network, as shown in Figure 4. In addition to this, a base station relays data from a device to another. Moreover, if the sender and the receiver are located at different cells with each other, the DCN (Data Core Network) plays the role of wired communication between each side of the base station[4].

Figure 5 shows the protocol stack of medical devices and that of base station. In this model, each medical device has four layers - Application, Transport, Data Link, and Physical. The application layer represents a network process that provides audio, video, or electronic signal streaming which need seamless services. Once data provided by the application layer is delivered to the transport layer, the data is encapsulated by RTP (Real-Time Transport Protocol) or UDP, both of which are sensitive to delays introduced by network latency. The data link layer is divided into three sub-layers - Security, LLC (Logical Link Control), and MAC (Medium Access Control). The AES cipher in the security layer encrypts and decrypts to protect the health-related data from security threats such as network tampering, sniffing, etc. The LLC layer plays a role of RS coding and block interleaving to increase the reliability of data communication between devices. And the MAC layer uses TDM based channel access method to bound the transmission delay. Finally, the physical layer carries out inner FEC such as turbo coding and modulation like QPSK (Quadrature Phase-Shift Keying), 8-PSK (Phase-Shift Keying), 16-QAM (Quadrature Amplitude Modulation), etc.

4. END-TO-END DELAY ANALYSIS FOR WIRELESS ECG

One of most important QoS metrics in medical applications is end-to-end delay from a medical device to another because most of medical applications require reasonably bounded latency. This E2E delay is highly dependent on channel condition, network scheduling method, protocols used in devices, required level of reliability, and so on. For simplicity, however, we only consider delays due to the processing in network protocol layers shown in Figure 5.

To illustrate how the worst-case end-to-end delay of a medical application is estimated, we use wireless ECG as an example application. In this example, a wireless ECG continuously sends patient's heart-beat represented by electrical signals to an ECG monitor, both of which are located in the same cell[5]. Figure 5 also shows possible sources of delays, $d_1, d_2, ..., d_{20}$, each of which represents the time required for processing of each layer. Among them, we only need to con-

[4]This paper is based on the concept of [8]. In that work, a central station(receiver) and a base station are considered as an unified entity, since they are connected with wired network. However, in this paper, we consider them as separate entities connected with each other through wireless cellular network.

[5]If they are located in different cells, a delay for passing through the DCN will occur. Since the DCN is a wired network, however, we can neglect the propagation delay. Also, the routing delay can be obtained by using existing routing delay analysis.

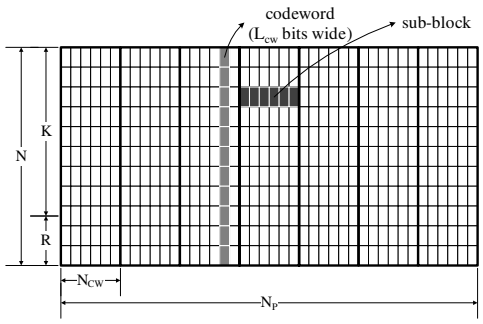

Figure 6: An Example of ECB

sider the delays introduced by outer RS decoding (d_{17}) and AES decryption (d_{18}), both of which at the receiver side, since all except these delays are not dependent on the channel condition, so that each of these can be bounded as a constant value or negligible. To be specific, we assume that the delays of application and transport layers (d_1, d_2, d_{19}, and d_{20}) are constant. In addition, the time required to encrypt using AES (d_3) and to encode using RS (d_4 and d_{11}) can be regarded as constant, because these delays are dependent on not probability of error occurrence, but data rate. The MAC access delays (d_5, d_9, d_{12}, and d_{16}) are also bounded as constant values since the MAC of CDMA2000 1xEV-DO is based on TDM. Specifically, if the data rate of the channel, the number of time slots, and the size of a data and that of a time slot are known, we can calculate the worst-case channel access delay, which is also a constant. Moreover, if we use hardware implemented turbo or convolution coding in the physical layer, these introduced delays (d_6, d_8, d_{13}, and d_{15}) are negligible. Similarly, the time to RS decode at the base station's LLC layer (d_{10}) is also negligible, because most of infrastructure base stations have a capability to exploit special-purpose hardware supporting RS coding. Finally, the propagation delays (d_7 and d_{14}) between a device and the base station can be disregarded as well, considering the cell radius of wireless cellular network and communication distances between them.

Recall that we will only deal with delay analysis for outer RS decoding ($d_{RS,de} = d_{17}$) and AES decryption ($d_{AES,de} = d_{18}$). However, the delays due to AES encryption ($D_{AES,EN} = d_3$) and RS encoding ($D_{RS,EN} = d_4$) will be considered, although these are constant. All other delays are summed and represented by D_c, and we will not illustrate how these are obtained. We can therefore simply express the worst-case end-to-end delay of a medical application operating in our network architecture as follows:

$$d_{e2e} = d_{RS,de} + d_{AES,de} + D_{RS,EN} + D_{AES,EN} + D_c. \quad (1)$$

4.1 Worst-Case Delay Analysis for RS decoding

Prior to RS decoding, a certain amount of data is buffered in ECB to be decoded as a codeword, and this operation causes a significant buffering delay, $d_{RS,buf}$. To derive an equation for estimating the buffering delay, we assume that the ECB consists of N rows, each of which has N_p sub-blocks, and a sub-block contains N_{cw} codewords, L_{cw} bits wide each, as shown in Figure 6. If the data rate of the MAC payload is denoted by μ_p, the buffering delay of this ECB therefore can be computed as follows:

$$d_{RS,buf} = \frac{N \times N_p \times N_{cw} \times L_{cw}}{\mu_p}.$$

Figure 3 in [7] depicts how RS decoder decodes and corrects a received codeword. When a codeword is given to a decoder, component named C_1 first examines whether the codeword has any erroneous symbol in it. If it does not have any error, the codeword is decoded and then forwarded to the upper layer, LLC. Otherwise, if it is capable to be corrected, the corresponding syndromes are built in component C_2, and every erroneous symbol is corrected by component C_3 until no more errors remain. If the number of errors in the received codeword, denoted by n_e, exceeds the decoder's error correction capability R, the codeword is decoded and forwarded to SEC without performing correction processes C_2 and C_3. Thus, the worst-case delay in RS decoding occurs when a received codeword contains $n_e = R$ erasures, that is the maximum number of errors that can be fully corrected by (N, K, R) RS decoder. If we represent the computation time of each component as $C_{RS,1}$, $C_{RS,2}$, and $C_{RS,3}$ respectively, the worst-case delay for RS decoding a codeword, $d_{RS_{cw},de}$, is

$$d_{RS_{cw},de} = C_{RS,1} + C_{RS,2} + R \times C_{RS,3}.$$

Accordingly, the worst-case delay to RS decode, including buffering, for an ECB is defined as follows:

$$\begin{aligned} d_{RS,de} &= N_p \times N_{cw} \times d_{RS_{cw},de} + d_{RS,buf} \\ &= N_p \times N_{cw} \times \{C_{RS,1} + C_{RS,2} + R \times C_{RS,3} + \frac{N \times L_{cw}}{\mu_p}\}. \end{aligned} \quad (2)$$

Notice that N_p affects the delay as well as the level of block interleaving. As this increases, we can achieve higher reliability, but the delay becomes longer. Meanwhile, if the time required to RS encode for codeword, $d_{RS_{cw},en}$, and the data rate of application, $\mu_{p'}$, are given, we can obtain the worst-case delay for RS encoding as follows :

$$D_{RS,EN} = N_p \times N_{cw} \times \{d_{RS_{cw},en} + \frac{N \times L_{cw}}{\mu_{p'}}\}. \quad (3)$$

4.2 Worst-Case Delay Analysis for AES decryption

The AES cipher in SEC sub-layer takes a block of encrypted data to decrypt. If the given block has at most R erroneous symbols, then the block passes through an initial round, a certain number of main rounds, and then a final round, to be finally decrypted, as shown in Figure 7. Otherwise, however, if the number of erroneous symbols of the block, n_e, exceeds the maximum number of correctable erroneous symbols of RS decoder, R, the AES cipher doesn't carry out decryption at all. Thus, the worst-case scenario of AES decryption for a data occurs when every cipher blocks

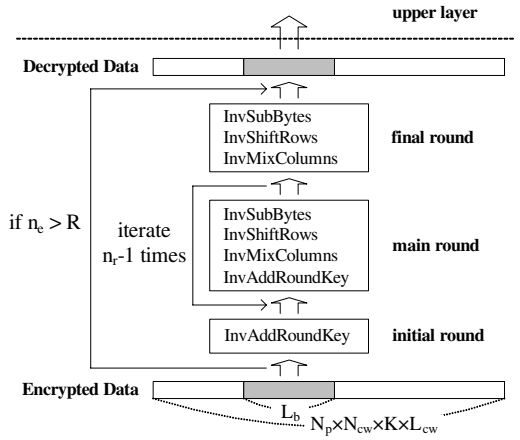

Figure 7: The Process of AES Decryption

constituting the data has either no error or at most R errors, and the delay of this case can be simply represented as follows:

$$d_{AES_{block},de} = d_{AES_p,initial} + (n_r - 1)d_{AES_p,main} + d_{AES_p,final}, \quad (4)$$

where n_r is the number of rounds, which is varying with the combination of block length and key length, and $d_{AES_p,x}$ is the delay for carrying out each part x. Since each part of AES is composed of one or several transformations, we can rewrite the Equation (4) as follows:

$$\begin{aligned} d_{AES_{block},de} &= C_{AES_t,invArk} \\ &+ (n_r - 1)(C_{AES_t,invSb} + C_{AES_t,invSr} + C_{AES_t,invMc} \\ &+ C_{AES_t,invArk}) + (C_{AES_t,invSb} + C_{AES_t,invSr} + C_{AES_t,invArk}) \\ &= (n_r + 1)C_{AES_t,invArk} + n_r(C_{AES_t,invSb} + C_{AES_t,invSr}) \\ &\quad + (n_r - 1)C_{AES_t,invMc}, \end{aligned}$$

where each $C_{AES_t,y}$ is the worst-case computation time of each transformation y. These $C_{AES_t,y}$ can be estimated by using execution time profiling, however it is out of the scope of our concern.

If we denote the size of a block as L_b, the number of blocks n_b which will be decrypted is $\lceil \frac{N_p \times N_{cw} \times K \times L_{cw}}{L_b} \rceil$, since we only need to decrypt K rows of data symbols in ECB. Therefore, the worst-case delay for AES decryption can be approximately[6] computed as follows:

$$\begin{aligned} d_{AES,de} &= n_b \times d_{AES_{block},de} \\ &= \lceil \frac{N_p \times N_{cw} \times K \times L_{cw}}{L_b} \rceil \times \{(n_r + 1)C_{AES_t,invArk} \\ &+ n_r(C_{AES_t,invSb} + C_{AES_t,invSr}) + (n_r - 1)C_{AES_t,invMc}\}. \end{aligned} \quad (5)$$

Note that, the AES standard only allows the block length L_b to be 128 bits, and the key length only to be 128, 192, or 256 bits. In these cases, the number of rounds n_r are 10, 12, and 14, respectively. Meanwhile, the worst-case delay for AES encoding, $D_{AES,EN}$, can be computed with Equation (5), by replacing each $C_{AES_t,invX}$ with $C_{AES_t,X}$, as follows:

$$\begin{aligned} D_{AES,EN} &= \lceil \frac{N_p \times N_{cw} \times K \times L_{cw}}{L_b} \rceil \times \{(n_r + 1)C_{AES_t,ark} \\ &+ n_r(C_{AES_t,sb} + C_{AES_t,sr}) + (n_r - 1)C_{AES_t,mc}\}. \end{aligned} \quad (6)$$

[6]We simply assume that there is no variances of $C_{AES_t,y}$ for a transformation y due to iteration order or cache operation.

Table 1: The Parameters Used in the Example Analysis

Parameter	Value
Number of leads	6
Sampling rate of a lead	500Hz
Sample size	16 bits
Data rate of reverse channel	76.8 kbps
Data rate of forward channel	614.4 kbps
E2E Delay Requirement	3 sec
(N, K, R)	(16,12,4)
μ_p	64.0 kbps
N_p	10
N_{cw}	125
L_{cw}	8 bits
L_b	128 bits
n_r	10

5. CASE STUDY OF WORST-CASE DELAY ANALYSIS FOR WIRELESS ECG

To examine whether the supposed architecture is applicable to practical medical applications, we will now perform a delay analysis for wireless ECG application with the parameters in Table 1. In this example application, each 16 bits of heart-beat samples are collected from 6 leads of ECG at a rate of 500Hz, so that these makes 48.0 kbps of data stream. After passing through (16,12,4) RS encoder, this data stream is expanded to the data rate of 64.0 kbps for MAC payloads, which is enough to be transmitted through 76.8 kbps of reverse channel and 614.4 kbps of forward channel[12]. With this application environment, we assume that an ECB row has 10 sub-blocks, and each sub-block contains 125 codewords, each of which is 8 bits wide. In addition, the size of both AES cipher block and key are 128 bits, thus the number of round is 10 according to AES standard[4]. Finally, we assume that this wireless ECG application requires 3 seconds of end-to-end delay.

We now have to obtain the execution times for each RS decoding component and each AES transformation to be obtained. To obtain these values, we performed execution time profiling using IAR® embedded workbench[6] for ARM9-TDMI with a clock speed of 133MHz. In addition, we used the Minsky's version of software implemented RS encoder/decoder[10]. On the basis of these environment, we obtained the expected time required by each component of RS decoding for a codeword, as shown in Table 2. Similarly, we performed same profiling for both AES encryption and decryption, and Table 3 shows the results. Note that in this result, all except MixColumn transformation are symmetric, which means that the internal operation of one of those transformations is same whenever it encrypts or de-

Table 2: Expected Time Required by each RS Decoding Component

Component x	1	2	3
$E(C_{RS,x})$	16.4	377.4	35.9

(unit : μs)

Table 3: Expected Time Required by each AES Transformation

Transform x	$SubBytes$	$ShiftRows$	$AddRoundKey$	$MixColumns$
$E(C_{AES_t,x})$	16.38	0.22	1.62	1.63
$E(C_{AES_t,invX})$	16.38	0.22	1.62	45.85

(unit : μs)

crypts. The Inverse-MixColumn transformation has more operations in it, however, so that it takes more execution time.

With the parameters and the results obtained from execution time profiling, we can approximately estimate the end-to-end delay of our wireless ECG application by using Equation (2), (3)[7], (5), and (6). Table 4 shows the result[8] of worst-case end-to-end delay of the application and each delay part of RS encoding/decoding, AES encryption/decryption, varying with the level of interleaving, N_p. As shown in this result, the end-to-end delay increases with N_p. Also most of the delay is caused by RS coding, which has an overhead of buffering. But most of all, it is important to note that the end-to-end delays with N_p of $1 \sim 4$ are less than 3 sec, which is the requirement of this application.

6. CONCLUSIONS

In this paper, we have proposed an application of the wireless cellular technologies of CDMA2000 1xEV-DO, as a promising solution to wireless medical systems. We have presented an end-to-end delay analysis method for a medical application using CDMA2000 1xEV-DO. Furthermore, from a case study on wireless ECG, we have shown that the proposed wireless cellular technology is promising for wireless medical networks.

7. REFERENCES

[1] S. D. Baker and D. H. Hoglund. Medical-grade, mission-critical wireless networks. *IEEE Engineering in Medicine and Biology Magazine*, 27(2):86–95, March/April 2008.

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

[3] N. Chevrollier and N. Golmie. On the use of wireless network technologies in healthcare environments. In *Proceedings of the Fifth IEEE Workshop on Applications and Services in Wireless Networks(ASWN 2005)*, pages 147–152, June 2005.

[4] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Springer-Verlag, 2002.

[5] N. Golmie, D. Cypher, and O. Rebala. Performance analysis of low rate wireless technologies for medical applications. *Computer Communications*, 28(10):1266–1275, June 2005.

[6] IAR® Embedded Workbench for ARM. http://www.iar.com.

[7] K. Kang, Y. Cho, and H. Shin. Energy-efficient mac-layer error recovery for mobile multimedia applications in 3gpp2 bcmcs. *IEEE Transactions on Broadcasting*, 53(1):338–349, March 2007.

[8] K. Kang, M.-Y. Nam, K.-J. Park, C. Kim, and L. Sha. Quality of service in wireless systems for medical applications. *submitted to IEEE Transactions on Computers*.

[9] E. Kershaw. Wireless networking reshapes the face of patient monitoring systems. *Biomedical Instrumentation & Technology*, 36(3):201–202, May/June 2002.

[10] Henry Minsky's Reed Solomon Encoder/Decoder. http://sourceforge.net/projects/rscode.

[11] R. R. P. Agashe and P. Bender. Cdma2000 high rate broadcast packet data air interface design. *IEEE Communications Magazine*, 42(2):83–89, February 2004.

[12] R. Parry. Cdma2000 1xev-do [for 3g communications]. *IEEE Potentials*, 21(4):10–13, Oct/Nov 2002.

Table 4: Result of Worst-Case Delay Analysis for Wireless ECG

N_r	$D_{RS,EN}$	$d_{RS,de}$	$D_{AES,EN}$	$d_{AES,de}$	d_{e2e}
1	344	317	19	56	736
2	688	634	37	112	1471
3	1033	952	56	168	2209
4	1377	1269	74	224	2944
5	1721	1586	93	280	3680
6	2065	1903	112	336	4416
7	2410	2220	130	392	5152
8	2754	2537	149	447	5887

(unit : ms)

[7]The value of $d_{RS_{cw},en}$ in the Equation (3) is 87.2 μs, which is also obtained from execution time profiling.

[8]D_c is disregarded since it has too small value compared with the main causes.