Performance Study of Real-Time ECG Transmission in Wireless Networks for Telecardiology Applications

Álvaro Alesanco, José García and Robert S. H. Istepanian

***Abstract*— An ECG coding and transmission methodology for real-time telecardiology applications using wireless networks is presented. The ECG signal is segmented into blocks which are coded and packetized for transmission. The compression method is based on the wavelet transform (WT) and the use of Set Partitioning in Hierarchical Trees (SPIHT) algorithm. Transmission simulations using TCP protocol through wireless channels with different characteristics are presented and discussed. Specific values for relevant parameters in the TCP protocol are proposed to enhance the quality of the communications.**

I. Introduction

Mobile Health (m-Health) is a new and evolving area of e-Health that exploits the recent advances in wireless communications for healthcare. m-Health can be defined as "mobile computing, medical sensor, and communications technologies for health-care" [1].

Telecardiology is one of the most mature and successful areas in m-Health. The most popular field in telecardiology deals with the transmission of electrocardiographic (ECG) signals over different kinds of communications networks. The inclusion of wireless networks opens a wide range of possibilities for ECG transmission in telecardiology projects. There are two different ways in which ECG data can be transmitted in a telecardiology environment: real-time and store-and-forward (pre-recorded) transmission. Store-and-forward ECG transmission has been widely used in earlier telecardiology systems where real-time ECG interpretation and/or supervision was not crucial. One of the main challenges in telecardiology is found on patient monitoring and early diagnosis in the area of emergency e-Health, where real-time ECG transmission is required. Emergency e-Health is synonym of m-Health since the main way to communicate an ambulance with a hospital is through a wireless channel. Transmission in wireless channels is much more challenging than in wired ones mainly due to the high bit error rate (BER) they may present.

Wireless networks have evolved from technologies such as Global System for Mobile communication (GSM) or General Packet Radio Service (GPRS) to the recently developed Third Generation (3G) Universal Mobile Telecommunications System (UMTS), thus permitting wider use of new wireless telemedicine services. Although 3G wireless networks increase considerably the available bandwidth, an efficient ECG coding method is still required, especially when the ECG shares the bandwidth with other clinical information and other users within the wireless network. Data compression methods based on orthogonal transforms, and specifically the wavelet transform (WT) coding approach have been extensively used for compression purposes due to its highly efficient energy packing and high compression gains [2].

During the last years, the increasing popularity of the Internet has made that the TCP/IP protocol stack has been implemented in the network and transport layers in almost every communication network. Wireless networks which formerly were implemented as circuit-switching have migrated to packet-switching thus implementing the TCP/IP protocol stack. This is the case of 3G wireless networks. In this way, the information has to be packetized to achieve real-time functionality in current wireless networks.

The aim of this paper is two-fold. Firstly, it describes an automatic ECG coding approach suitable for real-time transmission in wireless networks. It is based on the Set Partitioning In Hierarchical Trees (SPIHT) algorithm. Secondly, it covers the packetization and transmission issues using TCP/IP protocol stack in order to be used in current packet-switching wireless networks. Wireless errors effects are also analyzed from the cardiologists point of view providing a complete view of the real-time ECG transmission.

II. Coding methodology

In this approach, the ECG coding method is divided into the following stages, illustrated in the block diagram in Fig. 1:

i) Pre-processing.
ii) Wavelet transform.
iii) Coefficients coding.

A. *Pre-processing*

The ECG data stream is divided into fixed length blocks and every block is coded as an independent entity. Each lead in the block is treated independently through all the coding process. Then the baseline of the acquired ECG signal is removed. As real-time operation is required, a simple baseline estimation method is used: a third-order Butterworth low-pass filter (cut-off frequency equal to 0.5 Hz) used in forward/backward directions to avoid phase distortion. After

This work was supported by project TSI2004-04940-C02-01 from Comisión Interministerial de Ciencia y Tecnología (CICYT) and Fondos Europeos de Desarrollo Regional (FEDER) and project FIS PI051416 from Fondo de Investigación Sanitaria (FIS).

A. Alesanco and J. García are with the Communications Technologies Group (GTC), Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain alesanco@unizar.es

R. S. H. Istepanian is with the MINT Research Centre, Kingston University, UK.

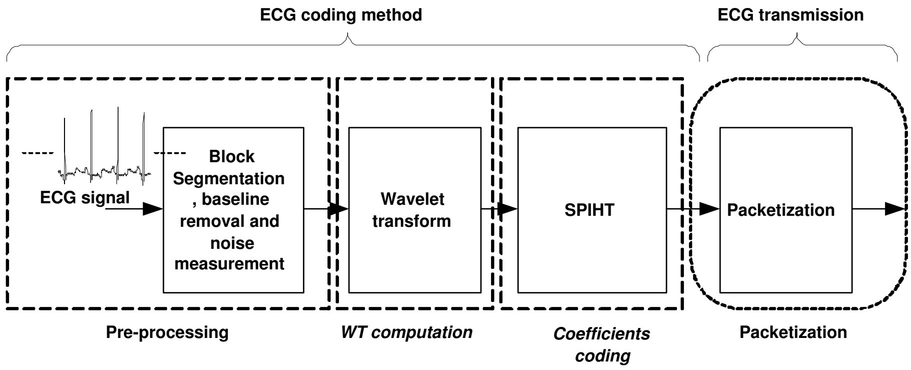

Fig. 1. ECG wavelet coding and packetization process.

the estimation, baseline is subtracted from the signal. In [3] a novel methodology for selecting automatically the compression threshold was introduced. To perform the automatic selection, the noise of the ECG block had to be measured. To accomplish with this, a simple third-order high pass Butterworth filter with a cut-off frequency of 20 Hz was used. In [3], the noise was measured in the repolarization interval, which is a part of the ECG which frequency components are below 20 Hz. This was possible since QRS detection was carried out to compress the signal beat to beat instead of block to block. Because the method presented here is based in block segmentation instead of beat segmentation, we do not know where the repolarization interval is placed. Since it is probable that a QRS appears in the block and this complex has frequencies above 20 Hz, it is not possible to measure noise with the aforementioned filter in the complete block. To palliate this problem, the noise is measured dividing the block into smaller windows of length 0.12 ms, which is the maximum duration of a QRS considered as normal. Once the noise has been calculated in every window, we select the median of them as the noise of the block. Tests carried out with different signals and block morphologies have shown that this method obtains measurements very close to the values obtained if the noise were measured in the repolarization interval.

B. Wavelet transform and coefficients coding: SPIHT algorithm

SPIHT was firstly presented in [4] as an efficient method for coding wavelet coefficients in image compression. In [2], the algorithm was introduced for ECG compression, obtaining very good results when being compared with other ECG compression methods. The principles of the SPIHT algorithm are partial ordering of the transform coefficients by magnitude with a set partitioning sorting algorithm, ordered bit plane transmission and exploitation of self-similarity across different layers. By following these principles, the encoder always transmits the most significant bit to the decoder. For a complete explanation of the SPIHT algorithm, we refer to [2] and [4].

C. Guaranteeing quality using SPIHT

When compressing biomedical signals, it is very important to have control over the distortion introduced in the compression process. The reason is simple: the higher the compression rate, the higher the distortion introduced and a highly distorted signal could be not useful from a clinical point of view. Thus, controlling the distortion must be an essential part of any compression algorithm. Traditionally, the measure of distortion in reception has been performed using two mathematical indices: Root Mean Square (RMS) error and Percentual RMS difference (PRD). In this paper, we use RMS error so as to determine the compression threshold, as we will see later on.

In order to reconstruct the signal, the decoder initially sets the reconstructed coefficients vector, $\tilde{c}$ to zero and updates its components according to the coded information. Using the fact that the euclidean norm is invariant to the wavelet transform it is easy to see that the reconstruction error can be measured also in the wavelet domain using the coded coefficients. Thus, RMS and PRD indices can be calculated in order to controll the reconstruction error using the wavelet coefficients, c, by

$$D_{RMS/PRD} = \sqrt{\frac{\sum_{n=1}^{N} (c(n) - \tilde{c}(n))^2}{F_N}} \tag{1}$$

where F_N is the normalization factor with value N or $\sum_{n=1}^{N} c[n]^2$ for RMS and PRD respectively.

It is clear that the guarantee of reconstruction quality can be easily done by controlling the value of the coded coefficients. Guaranteeing reconstruction quality using the SPIHT algorithm can be achieved by stopping the coding process when the desired distortion is reached.

D. Compression working point

When a new compression method is developed, it is common that authors give compression results and compare them with other compression methodologies already available in the literature. Nevertheless, a critical point that it is not treated very often is which should be the working point of the compression method so as to guarantee that the reconstructed signal is clinically useful from the cardiologist point of view. In [3], a new approach on compression was introduced. The compression point should be that one which leads to a reconstruction error equal to the noise present in the ECG being compressed. Doing this we remove excessive noise from the ECG signal at the same time we compress it. Although in that paper the compression method was based on beat segmentation and not in block segmentation and the coefficient coding was not performed using SPIHT, it is easy to extrapolate the main idea and to implement it no matter what compression methodology is being used. Here, we extend the idea to block compression using one of the compression methods that better results have obtained: SPIHT. The compression working point is selected block to block depending on the measured noise of this block. In this way, the compression threshold is set as the value, measured in μV, of the noise. In Fig. 2 an example of the algorithm applied to record 100 of MIT-BIH Arrhythmia database [5] (records with two leads and a sampling frequency of 360 samples per second with a resolution of 11 bits per sample) can be seen. In Fig. 2(a) 5 seconds of the original record can be seen. The reconstructed signal is shown in Fig. 2(b). Consulted cardiologists have pointed out the there is no loss of information between the signals, thus being useful for diagnosis purposes.

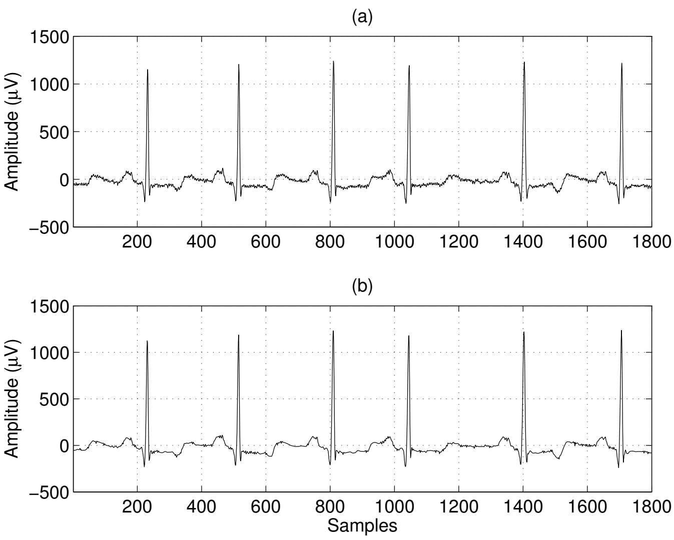

Fig. 2. Five seconds of register 100. (a) Original signal. (b) Reconstructed signal

In Tab. I results obtained when applying the method to lead 1 from eleven records in MIT-BIH Arrhythmia database (we will refer to them as MIT-A database) are shown for completeness. It is interesting to note how compression ranges from 7:1 to 10:1. A typical lossless compression method can achieve compression ratio of about 2:1, so we can see the advantage of using this method. It is very important to note that the introduced distortion is very low, thus guaranteeing quality in any case.

Performance results of SPIHT compared with other algorithms can be found in [2], so we refer those readers interested in them to this paper.

TABLE I

AUTOMATIC COMPRESSION MODE RESULTS FOR THE DATASET.

Rec.	**Noise (μV)**	**Rate (bit/s)**	**RMS (μV)**	**PRD (%)**
100	9.86	545.93	9.82	5.44
101	11.28	515.82	11.23	5.08
102	10.64	530.82	10.60	5.90
103	9.71	596.12	9.67	3.30
107	19.60	559.43	19.50	2.41
109	12.50	528.68	12.45	2.76
111	13.35	498.51	13.31	7.23
115	10.32	548.93	10.28	3.44
117	12.82	398.05	12.77	6.55
118	27.66	438.66	27.51	6.97
119	13.77	510.83	13.72	3.32

E. Real-time performance

As real time operation is required, an important value is the execution time for the proposed coding method. Tests carried out with a laptop (Centrino processor 1.4 Ghz and 512 MB RAM) have shown that the maximum execution times keep below 30 ms for coding and 10 ms for decoding one second of an 8-lead ECG.

III. ECG PACKETIZATION AND TRANSMISSION IN WIRELESS ENVIRONMENTS

A. Packetization

After WT coefficients coding, the resultant ECG data is ready to be transmitted. In order to send the coded data through wireless packet-switching networks, packetization has to be performed. Fig. 3 illustrates a typical wireless protocol stack and the protocol data unit (PDU) formats on every layer. It also shows the application layer PDU used for the coded ECG data.

The size of the application PDU depends on the ECG block being coded. For signals in the MIT-A dataset, this value ranges between 600 and 800 bits per lead. This PDU is encapsulated into one transport layer protocol. Transport protocol used here is TCP because it offers reliability in data delivery which is an essential requirement in biomedical data delivery. The radio link protocol (RLP) fragments the IP packet into several blocks which are transmitted independently. It is common that while satisfying the delay requirements, retransmission of RLP blocks is implemented. Nevertheless, in this study it is assumed that the benefits derived from retransmission are embedded in the channel packet error rate (PER) (decreasing it compared with that one could obtain without link layer retransmission). Hence, it is not relevant for the packetization issues at transport layer.

B. Wireless channel model and simulation setup

A first-order Markov model has been used in order to simulate the wireless channel. This simulation process has probed to be a good approximation for multipath fading

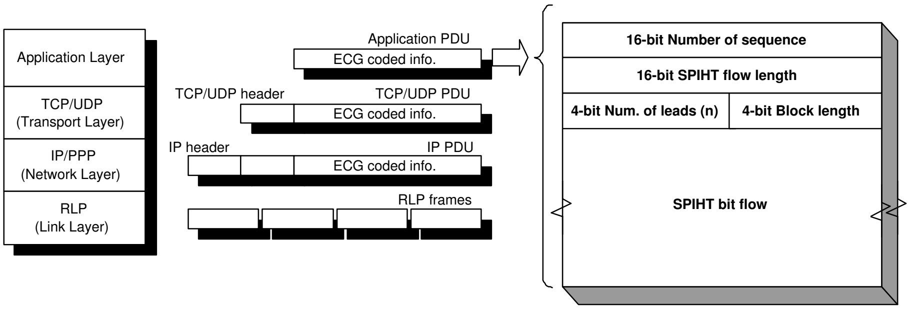

Fig. 3. Wireless protocol stack and ECG PDU.

channles, where error correlation leads to bursty error patterns [6]. All the statistics of the packet errors are determined by the transition matrix of the two-state Markov process:

$$M_c = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix} \tag{2}$$

were p and $(1 - q)$ are the probabilities that the nth packet transmission is successful, given that the $(j - 1)th$ was successful or non-successful, respectively. Note that the steady state PER in this model is $PER = (1-p)/(2-p-q)$ and the average burst length (ABL) is $1/(1-q)$. By choosing different values of p and q we can simulate fading channels with different degree of PER and correlation. It is also worthy to note that if we select $p = 1-q$, independent and identically distributed (i.i.d.) errors will be produced. Thus, this situation characterizes a Bernoulli channel, where error condition in packet n does not depend on error condition in packet $n-1$.

Network round trip time has been simulated with a normal distribution of mean 200 ms and standard deviation 50 ms. These mean values were selected after performing measures in a real 3G wireless network. Although the normal distribution is not probably the best distribution to fit the delay, it has been used for simplicity in this study. The capacity of the wireless network was set up to 64 Kbit/s, a very common value for 3G wireless uplink. We also assume that there are no errors in the ACKs as proposed in [7], which is a reasonable assumption since ACKs length in small compared with ECG data packets.

C. TCP ECG data transmission

TCP is a connection-oriented reliable protocol. If there are packets lost, TCP retransmits them. The way TCP have to ensure delivery is to acknowledge the reception of segments (at TCP level, packets are denoted as segments). Each received packet generates an acknowledge asking for the next packet to be expected unless there was a packet lost. If the receiver is waiting for segment N and segment $N + 1$ is received, an acknowledge asking for packet N is generated. Flow control is implemented using a sliding window mechanism. The window size is measured in bytes. The quantity of information that can be traveling without acknowledge is the size of the sliding window. When an acknowledge is received in the emisor side, the sliding window slides allowing to send another segment. It also implements a congestion control algorithm in order to avoid network congestion. The size of the congestion window is increased when an acknowledge is received but it drops when a packet is lost. The size of the sliding window is calculated as the minimum value between the congestion window and the receiver window (this is the window size recommended by the receiver and sent to the transmitter in the connection initial setup phase). In this way, TCP reacts to the packet drops decreasing the quantity of information sent to the network. Depending on which TCP version is being used, this basic idea is implemented in one way or another. A comprehensive study of TCP operating mechanism can be found in [8]. TCP uses a retransmission timer to ensure data delivery in absence of any feedback from the remote data receiver. When the retransmission timeout (RTO) expires, the retransmission is performed. RTO is updated with the round-trip time (RTT) of the transaction. If many losses are experimented by the TCP connection, the RTO value rises. RTO minimum and maximum values can be set up by the user. TCP implements a reset mechanism which finalizes the transmission if many packets are lost.

In real-time streaming applications it is common to implement a reception buffer to try to palliate delay jitter and to support retransmission of lost data. In our case, this buffer is measured in ECG blocks. The initial delay experimented by the receiver in monitoring the ECG is related to the buffer size.

We introduce several metrics to evaluate the communication and the monitoring process in reception. First of all, in our implementation the use of TCP makes that the bandwidth used for ECG transmission increases due to packet retransmissions. Thus, it is important to measure the bandwidth increment for different channel conditions. Another important parameter is the number of monitoring stops suffered in reception. If the reception buffer runs out of data, a stop will be produced in the ECG monitoring, restarting when the packet arrives. It is also very interesting the mean duration of these stops. These parameters, the number of stops and the mean duration of them are the indices that are related to the clinical quality of the received signal.

1) Tuning TCP for wireless ECG transmission: Tuning TCP is a very important stage before using it to transmit

ECG data. The main parameters that have to be tuned are RTO and retransmission attempts. RFC 2988 [9] gives the recommendations for computing RTO. It recommends that RTO does not be less than 1 second and RTO maximum value does not be less than 60 seconds. Regarding retransmission of lost data, many of the TCP implementations use 3 attempts for connection establishment and 6 for data packets before closing the connection due to excessive errors. For example, in the Windows XP implementation of TCP, RTO minimum value is set up to 1 s, RTO maximum value is equal to 64 s, maximum connection attempts is 3 and maximum data attempts is 6. In order to simulate transmission, we are going to work with the TCP implemented in Windows XP, which includes the TCP SACK version [10], but proposing changes in the parameter settings to improve the performance for real time ECG transmission.

We propose to increment the parameters maximum connection attempts and maximum data attempts to higher values (20 in our implementation) to cope with the high PER we could find in the transmission. In this way, the mechanism for closing the connection is triggered when 20 retransmissions are performed, which is improbable to happen. On the other hand, taking into account that an ECG PDU is produced every 1.42 seconds (one per block of 512 samples), we propose to change the parameters of RTO maximum and minimum values to 1 and 0.1 respectively. By doing this, we enable retransmissions before another ECG packet is produced and also we prevent RTO to increase in a way that highly decreases the transmission of data packets. Note that the proposed changes are recommendable because PDU generation rate of the ECG is very low and it is possible to enable retransmissions before another packet is produced since the time between packets is very high. For other applications such as audio or video transmission, the packet rate is much higher making completely different the problem of retransmitting lost data.

D. Transmission results and discussion

The effects shown in transmission regarding bandwidth increase, monitoring stops and mean duration of the stops are going to be the same no matter what is the signal being transmitted. This is because from the transmission point of view network packets are transmitting data, no matter if it is from one ECG or another. Thus, to show these effects we have used the output packets produced with record 100. In order to obtain results, every point in the following figures have been obtained averaging the results of 60 different realizations simulating the transmission of 10 minutes of ECG signal in a discrete event simulator.

In Fig. 4 the transmission bandwidth used by the emisor is shown as the PER increases for different ABL values. It can be seen, as it was expected, that as the PER increases, the used bandwidth increases. Note that to calculate this bandwidth we take into account all the packets sent by the transmitter, including retransmissions. Thus, as the PER increases, the retransmission of data to maintain the real time monitoring has to increase. Regarding the effect of the ABL in the bandwidth no significant effect can be detected.

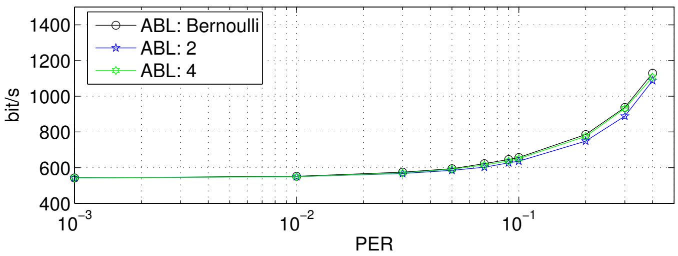

Fig. 4. Transmission bandwith for different PER and ABL values.

The number of stops per minute in the monitoring process can be appreciated in Fig. 5(a) and (b) for a buffer length of 2 and 3 blocks respectively. As it was expected, using a larger buffer leads to a decrease in the number of stops since there is more time available for the retransmission of lost data. On the other hand, using a buffer of 3 blocks means the introduction of an extra initial delay corresponding to the time of 2 blocks (as soon as the third block arrives, the monitoring process starts). In this study, where the block size is 512 samples and the ECG sampling frequency is 360 samples per second, this is translated into 2.84 s. If we take into account also the transmission delay, this value would be around 3 s. Consulted cardiologists have pointed out that this delay would be tolerable but that it would be recommendable not to increase it much more. Low ABL values have a positive effect in the number of stops. The more correlated the channel, the lower the number of stops. The worst results for all the cases are obtained when the channel is not correlated i.e. for a Bernoulli channel. This is an expected value since the errors in a Bernoulli channel are more spread compared with correlated channels thus provoking more stops. Note anyway that as the buffer size increases, this effect is palliated and the difference in the number of stops for different ABLs is less appreciable.

The mean duration of the stops can be seen in Fig. 6(a) and (b) for a buffer length of 2 and 3 blocks respectively. The effect of the ABL is very interesting, increasing the duration of the stops as the ABL increases. This effect is explained if we consider that the larger the ABL, the larger the number of packets lost consecutively and hence the larger the time the receptor has to wait to have information to monitor. On the other hand it can be seen that the PER do no affect highly to the duration of the stops. This effect was expected since the mean duration of the ABL do not depend on the PER.

We have to point out that if standard TCP parameters are used to carry out the communication, the real-time interaction is lost and connection is reset many times being more probable these events as the PER increases, thus causing the monitoring process to fail. With the setting proposed and tested in this paper, we palliate all this faulty effects.

Because the final users of a real-time ECG monitoring are the medical users, MOS tests are currently being carried out in order to collect evaluations of the monitoring process

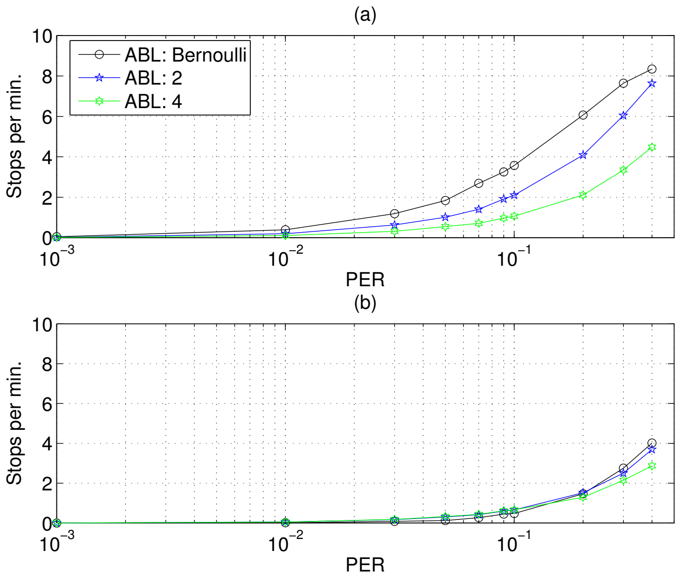

Fig. 5. Number of stops per minute for different PER and ABL values. (a) Buffer length 2 packets. (b) Buffer length 3 packets

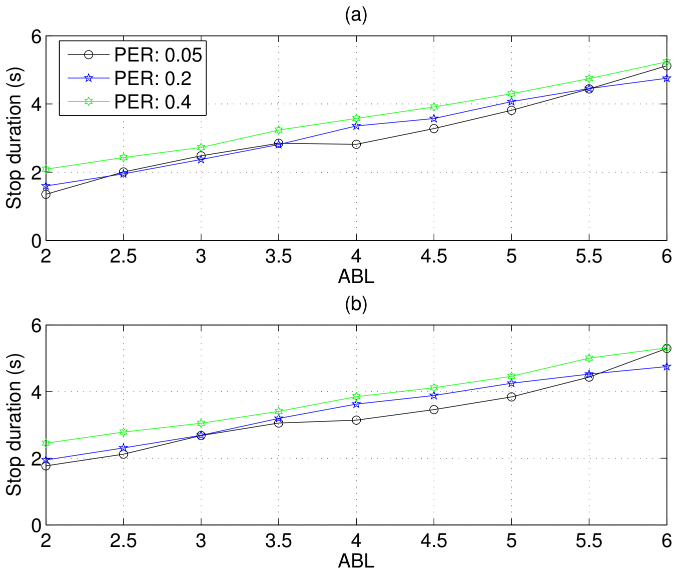

Fig. 6. Mean duration of the stops for different PER and ABL values. (a) Buffer length 2 packets. (b) Buffer length 3 packets

for different PER and ABL. Preliminary results show that even for high PER and ABL, the monitoring process could be carried out without a great disturbance. Anyway, more results are needed to give definitive results.

IV. Conclusions

A complete study including automatic ECG compression, coding and transmission issues for real-time cardiac monitoring using wireless networks has been presented. A simple method for compression based on the SPIHT and automatic thresholding based on the noise measure has been proposed. TCP has been used and a proposal for TCP parameters has been provided for erroneous and lost packets retransmissions to assure information integrity. Simulation results have shown that monitoring stops are more frequent as the PER rises and with low ABL. On the other hand, stops mean duration in independent to the PER but increases as the ABL does. Reception buffer palliates these annoying effects in the monitoring process but since real-time interaction between receiver and transmitter is required, the buffer should not be very large. Ongoing work is underway to clinically evaluate the monitoring process in order to determine the clinical limits for transmission in wireless channels. Preliminary results show that even for high PER and ABL the monitoring process could be carried out.

References

[1] R. Istepanian, E. Jovanov, and Y. Zhang, "Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 405–414, Dec. 2004.

[2] Z. Lu, D. Y. Kim, and W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 849–856, Jul. 2000.

[3] A. Alesanco, S. Olmos, R. S. H. Istepanian, and J. García, "Enhanced real-time ECG coder for packetized telecardiology applications," *IEEE Trans. Inf. Tech. Biomed.*, vol. 10, pp. 229–236, Apr. 2006.

[4] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst.*, vol. 6, no. 3.

[5] "The MIT-BIH arrhythmia database," *http://www.physionet.org/physiobank/database/mitdb/*.

[6] M. Zorzi, R. Rao, and L. Milstein, "On the accuracy of a first-order markov model for data block transmission on fading channels," in *Proc.IEEE ICUPC'95*, pp. 211–215, Nov. 1995.

[7] A. Kumar, "Comparative performance analysis of versions of tcp in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 485–498, Aug. 1998.

[8] W. Stevens, *TCP/IP Illustrated, Volume 1*. Reading. MA.: Addison-Wesley, 1994.

[9] http://www.ietf.org/rfc/rfc2988.txt.

[10] http://www.ietf.org/rfc/rfc2018.txt.