Congestion-Aware, Loss-Resilient Bio-Monitoring Sensor Networking for Mobile Health Applications

Fei Hu, Yang Xiao, *Senior Member, IEEE*, and Qi Hao, *Member, IEEE*,

***Abstract*—Many elder patients have multiple health conditions such as heart attacks (of various kinds), brain problems (such as seizure, mental disorder, etc.), high blood pressure, etc. Monitoring those conditions needs different types of sensors for analog signal data acquisition, such as electrocardiogram (ECG) for heart beats, electroencephalogram (EEG) for brain signals, and electromyogram (EMG) for muscles motions. To reduce mobile-health (m-health) cost, the above sensors should be made in tiny size, low memory, and long-term battery operations. We have designed a series of medical sensors with wireless networking capabilities. In this paper, we report our work in three aspects: (1) networked embedded system design, (2) network congestion reduction, and (3) network loss compensation. First, for networked embedded system design, we have designed an integrated wireless sensor network hardware / software platform for multi-condition patient monitoring. Such a system integrates ECG/EEG/other sensors with Radio Frequency Identification (RFID) into a Radio Frequency (RF) board through a programmable interface chip, called *PSoc*. Second, for network congestion reduction, the interface chip can use compressive signal processing to extract bio-signal feature parameters and only transmit those parameters. This provides an alternative approach to sensor network congestion reduction that aims to alleviate "hot spot" issues. Third, for network loss compensation, we have designed wireless loss recovery schemes for different situations as follows. (1) If original sensor data streams are transmitted, network congestion will be a big concern due to the heavy traffic. A receiver-only loss prediction will be a good solution. (2) If the signal parameters are transmitted, the transmission loss mandates a 100% recovery rate. We have comprehensively compared the performance of those schemes. The proposed mechanisms for m-health system have potentially significant impacts on today's elder nursing home management and other mobile patient monitoring applications.**

***Index Terms*—Mobile-health (m-health), sensor networks, network congestion, particle filtering, wavelets**

I. Introduction

CURRENT healthcare systems, structured and optimized for reacting to crisis and managing illness, are facing new challenges, e.g., a rapidly growing population of elderly and rising health care spending. For example, according to the U.S. Bureau of the Census, the number of adults age 65 to 84 is expected to double from 35 millions to nearly 70 millions by 2025 when the youngest Baby Boomers retire [1]. Also, overall health care expenditures in the U.S. reached $1.8 trillion in 2004 with almost 45 million Americans uninsured. In addition, a recent study found that almost one third of U.S. adults, most of whom held full-time jobs, were serving as informal caregivers - mostly to an elderly parent [1]. It is projected that U.S. health care expenditures will reach almost 20% of the Gross Domestic Product (GDP) in less then 10 years, threatening the wellbeing of the entire economy [1].

TABLE I
Multiple Healthcare Parameters that could cause alerts

Alert Type for Patients with Multiple Health Conditions	Detection Parameter that Goes beyond Normal Range
Low SpO_2	SpO_2<90% (default values, adjustable)
Bradycardia	HR>40 bpm (default values, adjustable)
Tachycardia	HR>150 bpm (default values, adjustable)
HR change	$\|\Delta$HR per 5 minutes$\|$>19%
HR stability	Max HR variability from past 4 readings >10%
BP change	Systolic or diastolic change>±11%

One critical issue for healthcare providers is the challenge of designing patient-centered processes that can improve the quality and safety of healthcare delivery. Such a patient-centered healthcare should enable improved patient-provider interactions through advanced communication technologies for patients with multiple chronic illnesses. Telemedicine through wireless communication networks can allow remote patient monitoring and convenient patient-provider electronic information exchange at any time and any place.

A patient with multiple health conditions needs a special telemedicine platform that is able to automatically collect multiple health parameters from the patient's body and then timely send an alert to a remote healthcare office if the value of any parameter is beyond normal ranges. Those health parameters include Heart Rate (HR), blood oxygenation level (SpO_2), Blood Pressure (BP), and so on. Table I shows a partial list of physiological conditions that may cause medical alerts [2], [3], [4], [5].

Recently, a promising wireless Telemedicine technology, called *Medical Sensor Networks (MSNs)* [6], [7], [8], [9], [10], [11], [12] , [46], [47], [48], has been proposed to closely monitor changes in patients' vital signs and provide feedbacks to help maintain an optimal health status. As shown in Fig. 1, an MSN sensor typically includes a sensing chip to sense healthcare parameters, a microcontroller to perform local data processing (such as data compression) and networking operations (such as communicating with a neighbor sensor), and a radio transceiver to wirelessly send/receive healthcare data. The entire MSN sensor is powered by batteries with a lifetime

Manuscript received 30 July 2008; revised 25 November 2008. The work is supported in parts by the U.S. National Science Foundation (NSF) under grants CNS-0716455 and CNS-0716211.

F. Hu, Y. Xiao, and Q. Hao are with Univ. of Alabama, Tuscaloosa, AL, USA; F. Hu and Q. Hao are with Dept. of Electrical and Computer Engineering, and Y. Xiao is with or Dept. of Computer Science (e-mails: fei@eng.ua.edu, yangxiao@ieee.org, qh@eng.ua.edu).

Digital Object Identifier 10.1109/JSAC.2009.090509.

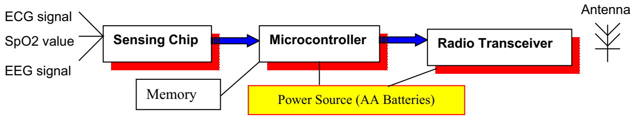


Fig. 1. MSN sensor mote components

of several months. Because the power storage is limited, it is very important to use *low-energy* MSN networking operations.

The MSN sensors can improve the healthcare quality greatly since the automatic, and wireless healthcare data transmission can avoid patients' frequent doctor visits and labor-intensive manual healthcare parameter collections. Such sensors are also important to capture medical emergency events. For instance, many serious heart problems affecting older people are transient and infrequent and can go unnoticed even by the patients. A sudden slowing of the heart rate that leads to a fainting spell may last less than a minute and occur only once or twice a week. This is often enough to make driving a car dangerous but not frequent enough for a doctor to spot it during a checkup or even by using a portable 24-hour electrocardiogram (ECG) recorder (called a Holter Monitor). Another problem, the uncoordinated quivering of the small upper chambers of the heart, a leading cause of stroke in people over 70, can be both infrequent and without obvious symptoms. Therefore, patient-triggered ECG recorders could miss it. However, our MSN ECG sensor can automatically collect ECG data and trigger an alert to the doctor if the ECG data mining software finds out an anomaly.

Note that an MSN sensor is different from traditional *wearable/portable medical devices*, but this does not always indicate that they are small and have wireless communication capability. Most such appliances are much heavier and larger than an MSN sensor that can be easily attached to a patient's body.

We have conducted the practical design of an MSN that has the following characteristics: (1) Our MSN is able to continuously collect multiple healthcare parameters from patients with multiple health conditions, shown in Fig. 2(a). If a patient has multiple chronic diseases, it is important to monitor multiple body parameters as shown in Table 1, instead of just one of them. Each sensor node can detect, sample, and process one or more physiological signals. For example, an ECG sensor can be used for monitoring heart activity, an electromyogram (EMG) sensor for monitoring muscle activities, an electroencephalogram (EEG) sensor for monitoring brain electrical activity, a blood pressure sensor for monitoring blood pressure, and a breathing sensor for respiration. Our system allows the use of an aggregation sensor, as shown in Fig. 2(a), to integrate all sensor data into database records and send out. We used PDA (Personal Data Assistant) to serve as such an aggregation node, shown in Fig. 2(b). (2) Our MSN can be applied in large US Nursing Homes through self-managed, relay-based wireless communication architecture. We have built an MSN hardware/software system that is suitable to large Nursing Homes with a radius of 300∼1000 feet. Each patient's sensor has limited wireless communication range (typically less than 100 feet) due to the low-power transceiver and tiny antenna.

Besides the design of MSN hardware (such as RFID, EEG/ECG sensors, RF chips, etc.), we have comprehensively investigated the following two ways to transmit bio-signals, as shown in Fig. 3. (1) The traditional way (shown in Fig. 3 top) is to simply transmit the bio-signals without any intelligent data processing. The continuous stream (distributed in different packets) keeps flowing in the multi-hop sensor networks. However, the wireless link can damage some signal values. Because any piece of signals could indicate some important medical symptom (such as a heart attack), the network needs to recover the lost data. To reduce the network congestion, we abandon traditional TCP-style retransmission and use destination-only loss compensation based on prediction algorithms. (2) Another congestion-aware transmission (shown in Fig. 3 bottom) is to compress the medical data through feature parameters extraction. The parameters are then forwarded to the destination. However, the loss recovery could not use prediction-based schemes since we need 100% error recovery. We thus have proposed a set of complete error resistant network protocols (including network-assisted, localized recovery and enhanced erasure codes) to guarantee the final signal reconstruction.

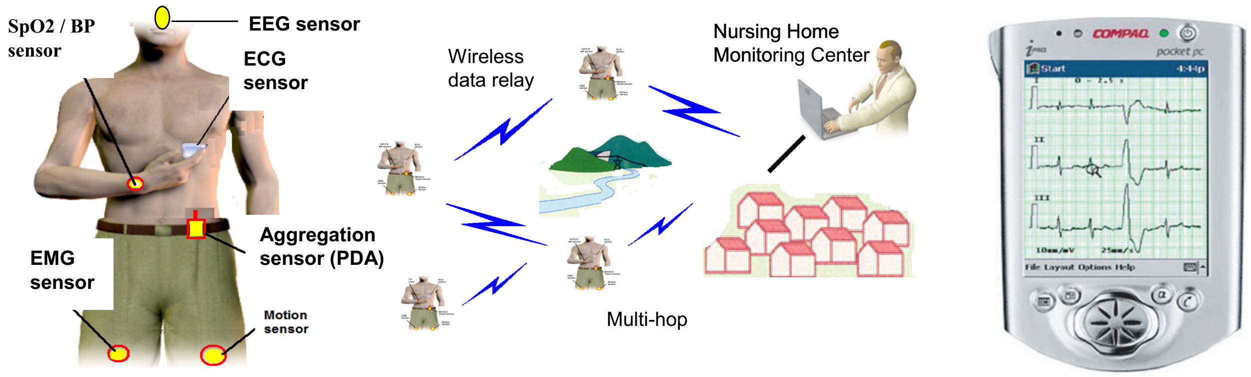
The rest of the paper is organized as follows. Section II provides system architecture and hardware design methodologies. In Section III, we discuss a way to send out raw ECG/EEG streams and to recover lost data through our proposed nonlinear prediction algorithms. Section IV describes our parameterized transformation for compressive signal transmission and corresponding complete loss recovery schemes. Experiment results are presented in Section V. Finally, we conclude this paper in Section VI.

II. System Architecture

A. *PSoc-based sensor-mote Interface*

This work proposes a novel approach of interfacing medical sensors to a RF mote (which includes wireless communication and a Microcontroller) through a highly versatile Programmable System on Chip (PSoC) Mixed-Signal array from Cypress [13], shown in Fig. 4.

The reasons for using PSoc instead of the built-in interface in our built RF mote (to be discussed in Section II-B) are as follows. PSoc extends the computational capabilities of the RF mote, while keeping its power consumption in check. A PSoC *Mixed-Signal Array* is a low-power programmable *systems-on-chip (SoC)* that allows programming of analog and digital (mixed-signal) components that are typically used in embedded systems. It also has a built-in microcontroller which integrates and controls all of the programmed components. Because of the extended computational capabilities through using this mixed signal array, complex computations such as filtering or triggering as well as application-specific data compression or suppression can be implemented at individual nodes. This can further reduce the data throughput over the network and result in reduced transmission time and improved network traffic. On the other hand, the built-in interface of our built RF mote still relies on TinyOS programming, which needs a deep understanding of many NesC sub-libraries. Furthermore, to save space and cost, our RF mote does not



(a) WSNs for M-Health

(b) Aggregation Node

Fig. 2. (a) Wireless Sensor Networks (WSNs) for M-Health and (b) Aggregation Node

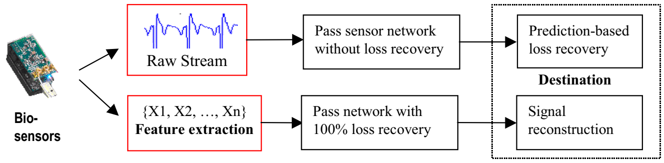

Bio-sensors
Raw Stream
{X1, X2, …, Xn}
Feature extraction
Pass sensor network without loss recovery
Pass network with 100% loss recovery
Prediction-based loss recovery
Destination
Signal reconstruction

Fig. 3. Two approaches to transmit medical sensor signals

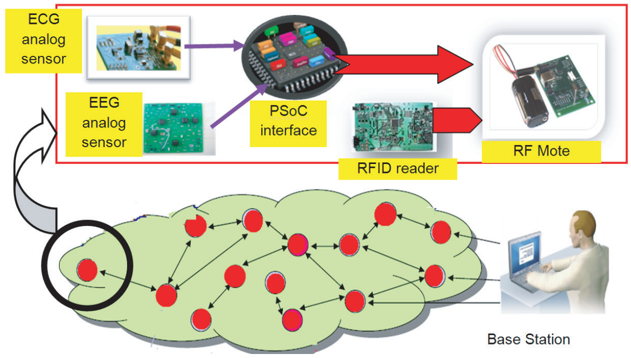


Fig. 4. PSoc-based sensor-mote interface

have enough memory space (<16K bytes) to build multiple interface routines. However, the PSoc allows multiple digital and analog interfaces. Its programming style is straightforward and the debugging tools are based on friendly GUI (Graphical User Interface). PSoc also has enough memory space to handle extra medical signal processing, which is needed in our medical signal processing algorithms, explained in Sections III and IV.

The PSoC design software is different from those used for programming traditional fixed-function microprocessor because of PSoC's configurable analog / digital hardware blocks. The PSoC Designer *Integrated Development Environment (IDE)* developed by Cypress Microsystems [13] is an innovative software development environment. It is a GUI-based design suite that simplifies the designing of the configurable blocks, provides the libraries for developing customized application code, and also provides advanced debugging tools to support the programming of the PSoC chip.

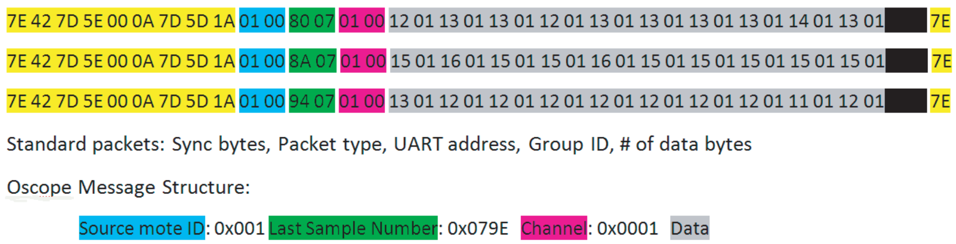
The raw data packets that are transmitted from PSoC to our designed RF mote (Section II-B) using UART port have format shown in Fig. 5. The data has to be sent in a little-endian format. The *synchronization byte* (0x7E), packet type (0x42), *UART address* (0x007E = 7D 5E 00, because of escaping byte), Message type *UART* (0A) and *Group ID* (7D) are highlighted in Fig. 5. The highlighted bytes on the third column are the CRC (Cyclic Redundancy Check) error correction bytes.

Fig. 6 describes our program flow of PSoC in order to send the data in the TinyOS message structure described above. Since the data payload and the CRC change with each packet, a 13-integer array, *iData* (Fig. 6), containing only the data payload, is initialized with the constant source mote as 0x0001, cumulative data count as 0x0000, ADC channel number (remains constant) as 0x0001, and the sampled data as all zeros. Ten sampled values are then added to the *iData* array. A cumulative count variable is used to keep track of the last sample number transmitted. This count is loaded into the *iData* array after the samples have been loaded. The *iData* array is now required to be converted to a 26-character array because all the sampled data is to be transmitted in the little-endian format. Once the new character array is formed, the CRC is initialized with a pre-calculated value of the standard bytes because they remain the same with every packet. The PC now jumps to the UART sending function that initially sends out the standard bytes (highlighted in Fig. 5, first column) then the character array that contains the data payload followed by the CRC.

B. RF Mote Design

Our original RF mote was based on TelosB motes from Crossbow Inc. [15]. It is an ultra-low-power wireless module intended for sensor networks applications. Regarded as the next-generation mote platform, it offers 10kB on-chip RAM and also provides IEEE 802.15.4 Chipcon radio [16] with an integrated on-board antenna providing up to 125 meters of range. Constructed around a TI MSP430 microcontroller [17], the TelosB has on-board ADC peripherals with expansion bays, to which the customized sensor board is connected.

However, we found out a few problems when using TelosB. First, its unit price is high in terms of large-scale MSN deployment [15]. Secondly, its power lifetime is around 3∼6 months depending on how often the ECG signal is transmitted



```
7E 42 7D 5E 00 0A 7D 5D 1A 01 00 80 07 01 00 12 01 13 01 13 01 12 01 13 01 13 01 13 01 13 01 14 01 13 01 [redacted] 7E
7E 42 7D 5E 00 0A 7D 5D 1A 01 00 8A 07 01 00 15 01 16 01 15 01 15 01 16 01 15 01 15 01 15 01 15 01 15 01 [redacted] 7E
7E 42 7D 5E 00 0A 7D 5D 1A 01 00 94 07 01 00 13 01 12 01 12 01 12 01 12 01 12 01 12 01 12 01 11 01 12 01 [redacted] 7E
```

Standard packets: Sync bytes, Packet type, UART address, Group ID, # of data bytes

Oscope Message Structure:

Source mote ID: 0x001 Last Sample Number: 0x079E Channel: 0x0001 Data

Fig. 5. Data format to be sent to RF mode

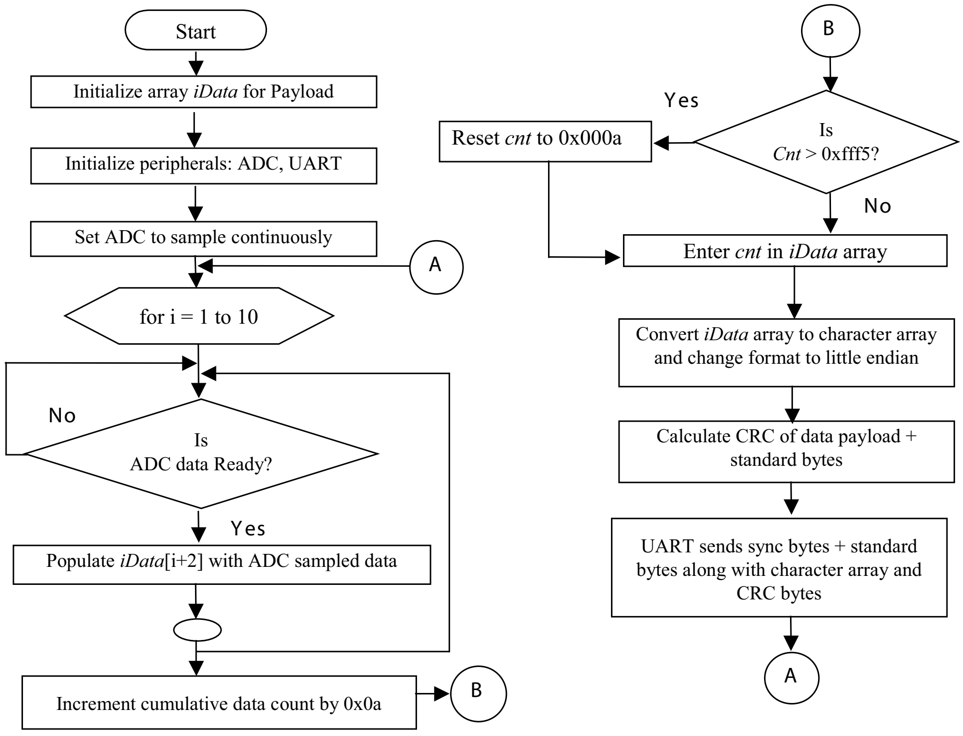

Start
Initialize array *iData* for Payload
Initialize peripherals: ADC, UART
Set ADC to sample continuously
A
for i = 1 to 10
No
Is
ADC data Ready?
Yes
Populate *iData*[i+2] with ADC sampled data
Increment cumulative data count by 0x0a
B
B
Yes
Reset *cnt* to 0x000a
Is
Cnt > 0xfff5?
No
Enter *cnt* in *iData* array
Convert *iData* array to character array and change format to little endian
Calculate CRC of data payload + standard bytes
UART sends sync bytes + standard bytes along with character array and CRC bytes
A

Fig. 6. Flowchart describing the PSoC sending sampled data

back to the server. Ideally, we hope that a cardiac patient can carry such a low-cost ECG sensor for at least one year without worrying about power exhaustion. Thirdly, its radio components cannot be enhanced since we cannot use a better radio transceiver / antenna to reach a longer distance.

Due to the above reasons, we have used Ember CPU-RF chips [18] to build our own RF motes shown in Fig. 7(a). The RF mote size is a little larger than 2 AA batteries. The heart of the RF board is the Micro Central Unit (MCU) / ZigBee [19] Transceiver unit. To design such a RF mote, we could use either a separate MCU / Transceiver, or use a SoC (System-on-Chip) that incorporates the two components together. The SoC option was chosen as it would be cheaper to implement, decrease programming complexity, and create an easier Printed Circuit Board (PCB) layout, as there will be fewer parts to layout. The Ember EM250 SoC [18] was selected for use in the ZigBee Data Forwarding Unit (DFU).

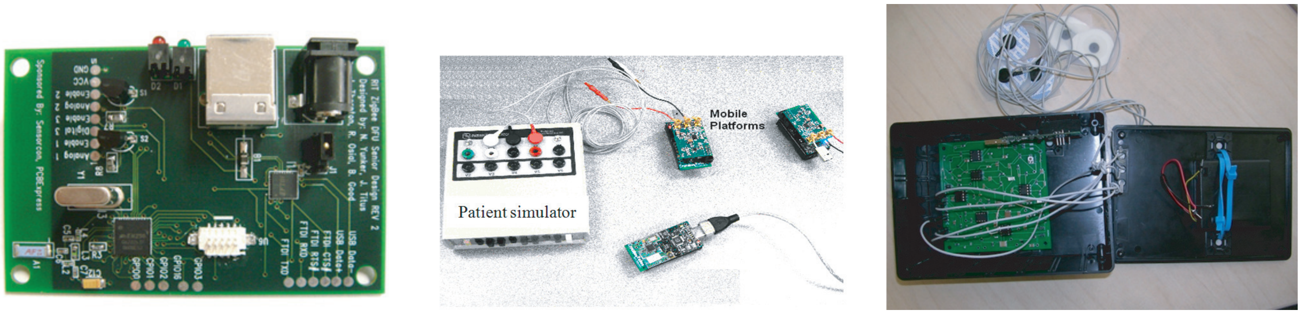
C. ECG Sensor

Our ECG sensor board design is assisted by Harvard University CodeBlue team [20]. The ECG lead extensions from the sensor board are pin-compatible and color coded to standard 3-Lead ECG monitoring systems. While there are different flavors of physiological chest leads, this system was designed to match any 3-Lead ECG Snap Set Lead wires. The Snap Set may be used to collect data by attaching to the appropriate jellied ECG conductive adhesive electrodes if humans were to be used for testing purposes. An alternative would be ECG signal simulators, as explained next.

The ECG generator used in this project is the Model 430B, 12-lead ECG generator as shown in Fig. 7(b). This generator provides a complete PQRST waveform at six preset rates (60, 75, 100, 120, 150, and 200 BPM (beats per minute)) as well as six preset amplitudes (0.1, 0.2, 0.5, 1.0, 2.0, and 5.0 mV). It is also capable of generating square waves using its 5 ECG snaps plus 10 banana jacks. This provides a good testing interface because this project will be upgraded to a 12-lead monitoring system in the future. Fig. 7(b) also shows the connection between 430B ECG simulator and our designed RF motes.

D. EEG Sensor

Recently we have designed a low-cost, portable, wireless EEG sensor platform that includes two primary elements:



(a) Our built RF Board (b) Model 430B Patient Simulator (c) Analog EEG Board with Electrodes

Fig. 7. (a) Our built RF Board, (b) Model 430B Patient Simulator, and (c) Analog EEG Board with Electrodes

the analog board (i.e., EEG sensor shown Fig. 7(c)) and the digital board (also called "wireless mote" or "RF board"). Fig. 8 shows the system architecture. We implemented only a two-channel case. More channels could be easily extended based on the same circuit principle. Two electrodes collect the EEG channel data from the brain and one electrode goes to the right leg. The input EEG analog signals need to pass DC (Direct Current) filter, amplifier, A/D converter and high-frequency filter. After the analog board filters and conditions the EEG signal, the A/D output is processed by the PSoc chip. The ESD (*Electro-Static Discharge*) circuit uses diode to achieve IC (Integrated Chip) protection and user wearing safety. The analog EEG signal amplification circuit is a crucial part. We did not use general instrumentation or linear amplifiers because we have found out that multi-stage combinational amplifier could provide a more accurate output gain. The Bessel Low-pass filter is adopted to filter out low-frequency signals. The digital board runs wireless communication protocols.

The digital board samples the output signal of the analog board. It accurately produces a digital representation of the EEG waveform. It then encapsulates the data into packets and sends them from the EEG sensor to a remote base station. Fig. 9 shows the entire design procedure of the digital board software. The EEG sensor remains in a passive *listen* state until the radio receives a valid message.

E. RFID Design

There is not much research conducted on the integration of RFID into medical sensor networks even though RFID could play an important role for elder cardiac healthcare. We have proposed to use the following two types of RFID readers in our MSN platforms for better monitoring of cardiac elder patients [47]: (1) Use Ultra-high frequency (UHF) RFID readers for patient tracing. They will be installed in the walls of nursing homes to keep track of cardiac patients' positions in the buildings. UHF readers work at around 900MHz radio frequency and have a tag-reading distance of 50∼150 feet. A UHF reader could send the patients' location information to a backend information database shown in Fig. 10. (2) Use High frequency (HF) RFID readers for medicine-taking management. It is important to monitor the medicine name, taking amount, and taking schedule for elders who may not have clear minds. Some medicine should be taken immediately after an abnormal ECG signal is detected by the medical center. For example, Heparin, a blood thinner, should be taken immediately after a heart attack. A HF reader could be embedded into the patient's ECG sensor. For example, SkyeRead M1-Mini reader [21] could be integrated into a Crossbow sensor. The HF reader could read the medicine tag through the radio frequency of 13.56 MH. HF readers work well within a distance of 5∼15 feet. Because a sensor (with the HF reader) participates in the communications, shown in Fig. 10, the medicine information finally enters the cardiac database.

Fig. 11 shows our RFID control software modules. The MSN gateway software is responsible for server-to-sensors wireless data transmission control, and the RF mote software has all RFID Reader/Tag control functions (such as the collision detection between one reader and multiple tags).

Unfortunately, none of the commercialized RFID readers allow us to reprogram the RFID for our proposed one-to-many reader-tag collision avoidance protocol. In addition, current RFID readers are very expensive.

F. EEG Warning Sub-System

Abnormal heart beats or brain status needs an accurate warning system. The mental disorder warning signals could be triggered in two places of our system, shown in Fig. 12. (1) At the local processing unit, we designed EEG pattern recognition software in PSoc to perform initial, simple classification on normal/abnormal EEG patterns. Abnormal pattern detection will trigger the warning device, which is an audio circuit to be designed in this project. The tiny warning device is integrated with the EEG sensor so that the patient can hear it clearly. (2) At the remote server (doctor's machine), the server runs comprehensive mental disorder detection software. If any serious drowsiness is detected, a warning message will be fed back to the patient's EEG sensor.

III. Wireless-Loss-Resilient Medical Transmission

There will be significant ECG/EEG data loss in a large-scale wireless sensor network. We have used the traditional two ways to achieve reliable sensor transmission in a wireless application [22], i.e., FEC (Forward Error Correction) and ARQ (Auto Retransmission reQuest). Unfortunately, those two schemes can largely shorten our sensors' lifetime due to the

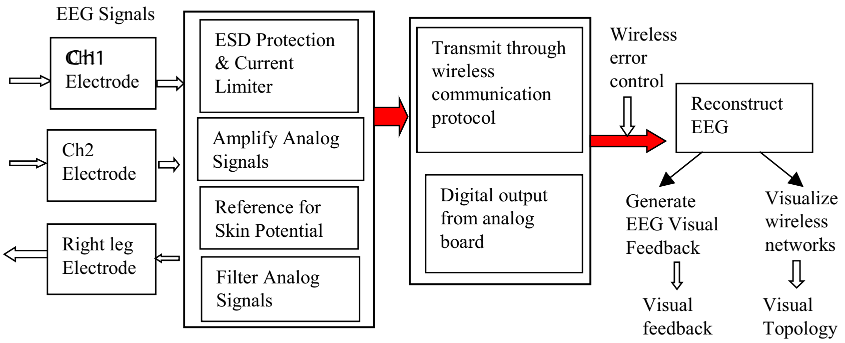

EEG Signals
CH1 Electrode
Ch2 Electrode
Right leg Electrode
ESD Protection & Current Limiter
Amplify Analog Signals
Reference for Skin Potential
Filter Analog Signals
Transmit through wireless communication protocol
Digital output from analog board
Wireless error control
Reconstruct EEG
Generate EEG Visual Feedback
Visualize wireless networks
Visual feedback
Visual Topology

Fig. 8. System level block diagram of wireless EEG sensor networks

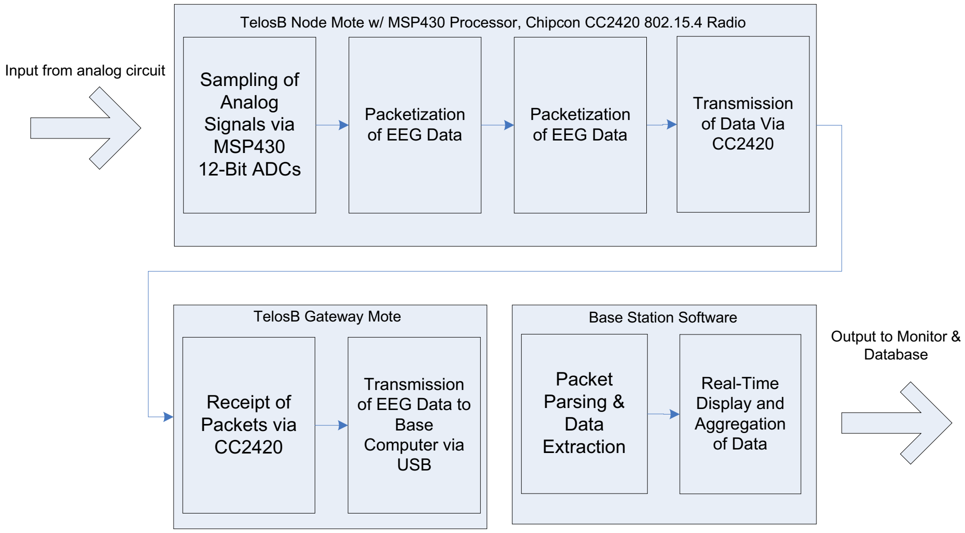

TelosB Node Mote w/ MSP430 Processor, Chipcon CC2420 802.15.4 Radio
Input from analog circuit
Sampling of Analog Signals via MSP430 12-Bit ADCs
Packetization of EEG Data
Packetization of EEG Data
Transmission of Data Via CC2420
TelosB Gateway Mote
Receipt of Packets via CC2420
Transmission of EEG Data to Base Computer via USB
Base Station Software
Packet Parsing & Data Extraction
Real-Time Display and Aggregation of Data
Output to Monitor & Database

Fig. 9. Digital signal /Software Flow Diagram

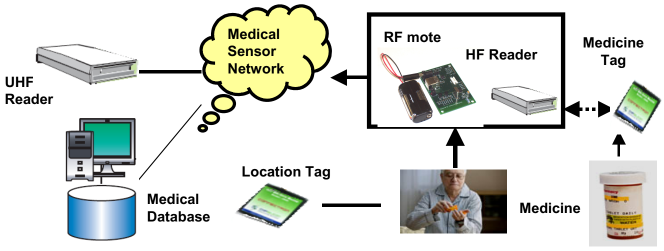


Fig. 10. UHF/HF RFID Readers for healthcare

large wireless communication overhead. FEC adds redundant error-control bits to the source data, and ARQ uses frequent retransmissions. The above two traditional schemes ignored the following simple fact [23], [24]: in resource-constrained MSN, most of node's power energy is consumed in hop-to-hop *wireless communications* instead of in *local CPU processing*. (In Section V, we will provide energy measurement results). The energy cost of transmitting 1 KB for a distance of 100 m is approximately equal to executing 3 Million local instructions by a 4MHz CPU [25].

Thus, this research has resorted to *destination-side* loss recovery approach (i.e., *predicting system future states by a sensor mote itself*). It uses only local processing without resorting source sensors. Unlike FEC and ARQ, our scheme does not increase wireless communication overhead. We have conducted a comparative study of two promising interference-resistance methods, i.e., *Sequential Monte Carlo* (also called *particle filters*) [27] and *Extended Kalman Filtering* (EKF). We hope to obtain insights into which method works better for ECG/EEG loss compensation.

A. Monte Carlo Based Loss Compensation

The motivation for using *particle filters* (or *Sequential Monte Carlo*) is two-fold. For one, particle filters can represent almost arbitrary posterior distributions; they are certainly well-suited to accommodate the types of uncertainty that arise in our distributed control scenario. More importantly, particle filters estimate posteriors over entire paths, not just the current state. Each particle can be thought of as an entire history or trajectory, and the set of all particles represents an approximation of the posterior over trajectories. This property of particle filters is not owned in *Kalman* filters.

We are concerned with the real-time next *state* estimation for non-Gaussian MG control signals. The *unobserved* global state $\{\mathbf{x}_t; t \in N\}$ is modeled as a *Markov* process with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The observations $\{\mathbf{y}_t; t \in N^*\}$ are assumed to be conditionally

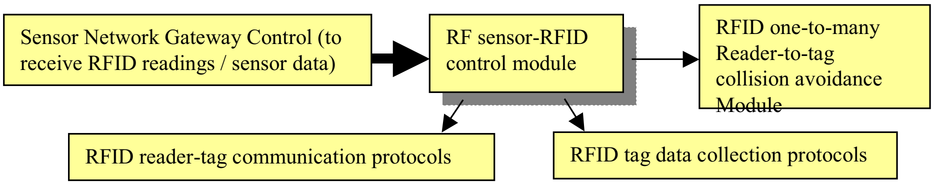

Sensor Network Gateway Control (to receive RFID readings / sensor data)
RF sensor-RFID control module
RFID one-to-many Reader-to-tag collision avoidance Module
RFID reader-tag communication protocols
RFID tag data collection protocols

Fig. 11. RFID Control Software Modules

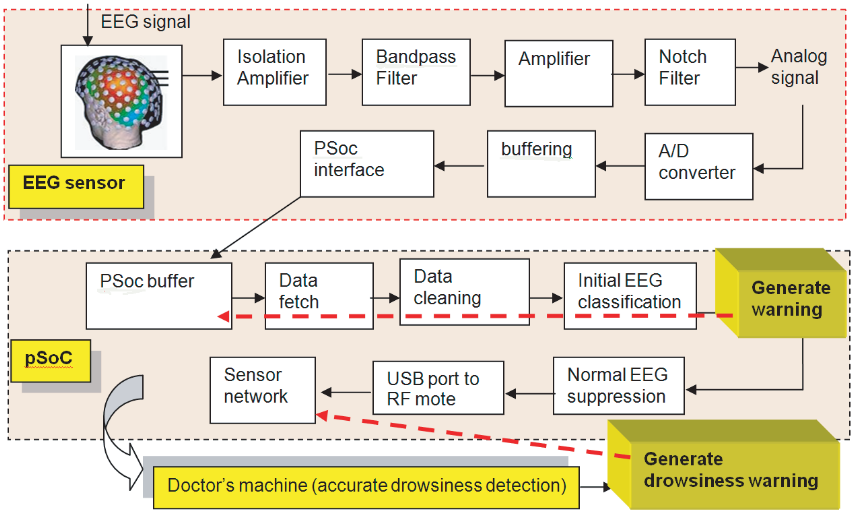


Fig. 12. (In EEG design) Drowsiness Warning System

independent in time given the process $\mathbf{x}_t$ and of marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$. We denote *system state* up to time t as $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \cdots, \mathbf{x}_t\}$ and *observations* up to time t . The measurements $\mathbf{y}_t$ are recorded by $\mathbf{K}$ controllers, and we use y_t^k to denote the subset of observations made by the $\mathbf{k}-\mathbf{th}$ controller. Our goal here is to predict (in real time) the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \cdots, \mathbf{y}_t\}$. We call *state trajectories* as *particles*. There is an *importance weight* associated with each *particle*. At a given time instant, this weight is representative of how well the state trajectory conforms to model dynamics and describes the set of observations, relative to the other particles. Whenever there is a transition between time instants, and a new observation becomes available, each trajectory is extended, and its associated weight is adjusted according to how well it explains the new observation.

If we do not use *networked prediction*, that is, there exists a single controller in the entire system that accepts all sensors' inputs and controls all devices, (this centralized control is not realistic in our scenario), and we may follow generic steps to estimate the posterior distribution [26], [27]:

Step 1 (*particle* initialization): Each particle (total number: N) is sampled from initial distribution. $x_0^{(i)} \sim p(\mathbf{x}_0)$, and every importance weight is initialized to: $\omega_0^{(i)} \sim 1/N$.

Step 2 (*importance* sampling): For each $I = 1, 2, \cdots, N$, $\widetilde{\mathbf{x}}_0^{(i)}$ is sampled from an *importance* distribution $\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$, which may be any distribution. The *importance weights* are evaluated as:

$$\bar{\omega}_i^{(i)} = \frac{p(y_t|\widetilde{x}_t^{(i)})p(\widetilde{x}_t^{(i)}|x_{t-1}^{(i)})}{\pi(\widetilde{x}_t^{(i)}|\widetilde{x}_{0:t-1}^{(i)}, y_{1:t})} \qquad (1)$$

TABLE II
DESCRIPTION OF THE VARIABLES

Variable	Meaning
$\mathbf{x}_k$	Actual state vector
$\mathbf{z}_k$	Actual measurement vector
$\underset{\sim}{\mathbf{x}}_k$	Approximate state vector from 5.3
$\underset{\sim}{\mathbf{z}}_k$	Approximate measurement vector from 5.4
$\hat{\mathbf{x}}_k$	An *aposteriori* state estimate at time k
$\mathbf{w}_k$	Random variable representing process noise
$\mathbf{v}_k$	Random variable representing measurement noise
$\mathbf{A}$	The Jacobian matrix of partial derivatives of f with respect to x
$\mathbf{W}$	the Jacobian matrix of partial derivatives of f with respect to w
$\mathbf{H}$	The Jacobian matrix of partial derivatives of h with respect to x
$\mathbf{V}$	The Jacobian matrix of partial derivatives of h with respect to v

Step 3 (*selection*): N particles $\left\{\mathbf{x}_{0:t}^{(i)}; i = 1, \cdots, N\right\}$ are formed by sampling with replacement from the set $\left\{\widetilde{\mathbf{x}}_{0:t}^{(i)}; i = 1, \cdots, N\right\}$ where the probability of sampling the i-*th* trajectory is $\bar{\omega}_t^{(i)}$. Note that Steps 2 and 3 will be repeated until system convergence.

B. Extended Kalman Filtering

A Kalman filter that linearizes about the current mean and covariance is referred to as an EKF [28]. We have used EKF to recover the damaged ECG data [47]. The information in Table II shows some variables used.

For the EKF, the state estimate is governed by a non-linear differential equation. Assume that f represents the non-linear function that relates the state at the previous time step, k - 1, to the state at the current time step, k. An optional driving function may be included as $\mathbf{u}_{k-1}$, and w represents the zero-mean Gaussian process noise. The measurement equation is in Eq. (3). Similarly, h is the non-linear function that relates $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$. Assume that $\mathbf{v}_k$ is the measurement noise. Since individual noise values, $\mathbf{w}_k$ and $\mathbf{v}_k$, are not known at each time step, instead the estimated state and measurement can be replaced with equation Eqs. (4) and (5), respectively. In both, $\hat{\mathbf{x}}_k$ is *a posteriori* estimate of the state from a previous time step k.

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{2}$$

$$z_k = h(x_k, v_k) \tag{3}$$

$$\widetilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{4}$$

$$\widetilde{z}_k = h(\hat{x}_k, 0) \tag{5}$$

New linearizing equations are needed to estimate a process with non-linear relationships. The governing equations seen in Eqs. (6) and (7) linearize about Eqs. (4) and (5), respectively. The equations for the matrices A, W, H, and V, seen in these equations can be seen in Eq. (8) through Eq. (11), respectively. For simplicity, the time subscript k is not used in the notation for these matrices even though they are in fact different at each time step.

$$x_k \approx \widetilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1} \tag{6}$$

$$z_k \approx \widetilde{z}_k + H(x_k - \widetilde{x}_k) + Vv_k \tag{7}$$

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \tag{8}$$

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \tag{9}$$

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\widetilde{x}_k, 0) \tag{10}$$

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\widetilde{x}_k, 0) \tag{11}$$

Given all this information, a new prediction error and measurement residual can be defined. Recall that the measurement residual reflects the discrepancy between the predicted measurement, $\mathbf{z}_k$, and the actual measurement, $\widetilde{\mathbf{z}}_k$. These can be seen in Eqs (12) and (13), respectively.

$$\widetilde{e}_{x_k} \equiv x_k - \widetilde{x}_k \tag{12}$$

$$\widetilde{e}_{z_k} \equiv z_k - \widetilde{z}_k \tag{13}$$

Unfortunately, in practice one does not have access to $\mathbf{x}_k$ in Eq. (12), since it is the actual state vector attempting to be estimated. One does have access to $\mathbf{z}_k$ since it is the actual measurement that one is using to make an estimate. By utilizing Eqs. (12) and (13), approximate linear governing equations for an error process can be written. These can be seen in Eqs. (14) and (15). It should be noted ε_k and η_k, representing new independent random variables having zero mean and covariance matrices $\mathbf{WQ}_k\mathbf{W}^T$ and $\mathbf{VR}_k\mathbf{V}^T$. Recall that Q and R are the process noise covariance and measurement noise covariance matrices, respectively. Additionally, $\widetilde{\mathbf{e}}_{x_k}$ has a covariance matrix of $\mathbf{E}\left\lfloor \widetilde{\mathbf{e}}_{x_k}\widetilde{\mathbf{e}}_{x_k}^T \right\rfloor$.

$$\widetilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \tag{14}$$

$$\widetilde{e}_{z_k} \approx H\widetilde{e}_{x_k} + \eta_k \tag{15}$$

It can be seen that Eqs. (14) and (15) are linear, and resemble the original discrete Kalman filter cases. This suggests that there should be a better way to estimate the error prediction of Eq (15). This new estimate of $\widetilde{\mathbf{e}}_{x_k}$, called $\hat{\mathbf{e}}$, can be found by using Eq (13) and a second hypothetical Kalman filter. The Kalman filter equation used for this estimate is:

$$\hat{e}_k = K_k\widetilde{e}_{z_k} \tag{16}$$

Once Eq. (16) is attained, it can be used in Eq. (17) to obtain an $aposterioristate$ estimate for the original non-linear process. By substituting Eq. (16) into Eq. (17), the equation in Eq. (18) is obtained. Finally, by substituting Eq. (13) into Eq. (18), the equation which will be used for the measurement update in the extended Kalman filter is achieved (see Eq. (19)). Note that $\widetilde{\mathbf{x}}_k$ and $\widetilde{\mathbf{z}}_k$ come from Eqs. (2) and (3), with the appropriate substitution for the measurement error covariance.

$$\hat{x}_k = \widetilde{x}_k + \hat{e}_k \tag{17}$$

$$\hat{x}_k = \widetilde{x}_k + K_k\widetilde{e}_{z_k} \tag{18}$$

$$\hat{x}_k = \widetilde{x}_k + K_k(z_k - \widetilde{z}_k) \tag{19}$$

C. ECG Modeling for Particle Filtering / EKF computation

Due to the non-linearity of ECG data, we need to determine the equation which governs an ECG and apply it to the EKF. Produced by an ECG, the signal is constructed by measuring electrical potentials between various points of the body using a galvanometer. Fig. 13 shows an example of a normal ECG trace, consisting of a P wave, a QRS complex and a T wave.

McSharry *et al.* have proposed a synthetic ECG generator, which is based on a nonlinear dynamic model [29]. This model has several parameters, P, Q, R, S, and T, which come from the ECG, and makes it adaptable to many normal and abnormal signals. The dynamic model consists of a three dimensional state equation, which generates a trajectory with the coordinate (x,y,z). These equations may be seen in Eq. (20) through Eq. (22). The variables α, $\Delta\theta_i$, and θ are given in Eq. (23) through Eq. (25). Note that Eq. (25) is the four quadrant arctangent of the real parts of the elements of x and y, with the bounds given in Eq. (26). The variable ω is the angular velocity of the trajectory as it moves around the limit cycle. The baseline wander of the ECG signal has been modeled with z_0.

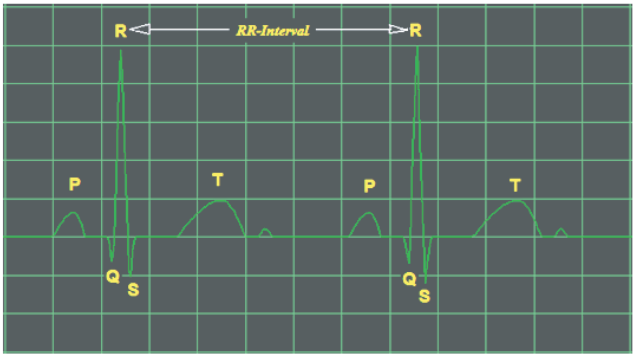


Fig. 13. Example of a Normal ECG Trace

$$x^{'} = \alpha x - \omega y \tag{20}$$

$$y^{'} = \alpha y + \omega x. \tag{21}$$

$$z^{'} = - \sum_{i \in (P,Q,R,S,T))} \alpha_i \Delta\theta_i exp\left(-\frac{\Delta\theta_i^2}{2b_i^2}\right) - (z - z_0) \tag{22}$$

$$\alpha = 1 - \sqrt{x^2 + y^2} \tag{23}$$

$$\Delta\theta_i = (\theta - \theta_i)\, mod\, (2\pi) \tag{24}$$

$$\theta = \arctan 2\,(y, x) \tag{25}$$

$$-\pi \leq \arctan\,(y, x) \leq \pi \tag{26}$$

The three dimensional trajectory which is generated consists of a circular limit cycle which is pushed up and down when it approaches one of the P, Q, R, S, and T points. In fact, each of the components of the ECG waveform has been modeled with a Gaussian function, which is located at a specific angle. This can be seen in Eq. (20) through Eq. (22) by neglecting the baseline wander term, $z - z_0$, and integrating the $z^{'}$ equation. The projection of the three dimensional trajectory on the z axis gives a synthetic ECG signal.

IV. Network Congestion Aware Medical Transmission

A. Medical Data Compression through Feature Coefficients

The above section simply transmits original ECG waves without any compression. Its advantage is to reduce software complexity and CPU processing workload. Furthermore, the loss recovery algorithms in the receiver side does not need to be 100% reliable since the consecutive ECG can still be understood even with some data errors. However, the above scheme brings significant network overhead. The wireless bandwidth can be easily consumed by the ECG/EEG streams and thus network congestion (called "hot spots") can occur in some middle sensors. To reduce the network congestion and also save wireless transceiver energy consumption, we propose to use accurate feature extraction to compress the generated ECG wave data. Our feature finding strategy is based on *multi-scale wavelet* analysis [48]. It can use localized singularity detection to zoom in non-smooth ECG variations.

The application of wavelet decomposition can provide feature coefficients for ECG classification. This initial heart beat pattern classification can at least tell us whether the patient has any abnormal heart beats. Therefore, our PSoc interface (see Section II-A) can filter *normal* sensor data and only transmit *abnormal* data. This further saves network bandwidth. To improve the ECG classification accuracy in terms of identifying different types of abnormal heart beats, we have investigated the theory of *Support Vector Machine* (SVM), which has been proved to be able to minimize the probability of misclassifying yet-to-be-seen patterns [31], [32]. The basic procedure of SVM algorithm is as follows. Considering the problem of separating the set of training vectors belonging to two separate classes, we have

$$\begin{aligned} S \;\; = \;\; & \{(x, y) \,|\, \{(x_1, y_1), (x_2, y_2) \cdots, (x_L, y_L)\}\,; \\ & x \in R^n, y \in (-1, 1)\} \end{aligned} \tag{27}$$

To apply the above SVM theory, we first extract some dominant features from ECG data to serve as the SVM classification vectors. Wavelets analysis is well known for its feature extraction efficiency for non-periodic signals. The Wavelet Transform of a function f is a convolution product of the time series with the scaled and translated kernel, and is given by:

$$W_{S,x_0} = \int_{-\infty}^{+\infty} \frac{1}{s} \bullet \Psi\left(\frac{x - x_0}{s}\right) \bullet f\,(x)\,dx \tag{28}$$

, where s is a scale parameter and is a space parameter.

To find out the "features" (i.e., the singularity points) of the above wavelet function, here we introduce the concept of *Local Holder Exponent* (LHE) [33]. The LHE of a function f(•) at the point $\mathbf{x}_0$, denoted as $h(\mathbf{x}_0)$, is defined as the largest exponent such that there exists a polynomial $\mathbf{P}_n\,(x)$ of order n satisfying the following condition for a in a neighborhood of $\mathbf{x}_0$:

$$|f\,(x) - P_n\,(x - x_0)| \leq C \bullet |x - x_0|^k \tag{29}$$

To reduce the regular wavelet analysis redundancy and calculation complexity, WTMM [33] proposes to change the "continuous" sum over space (see Eq (2)) to a "discrete" sum over the local Maxima of $|W_{s,x_0}\,(f)|$. Although WTMM provides efficient estimation for "Global" scaling of ECG time series, it has been shown that the "Local" scaling analysis could provide more relevant information on feature extraction [34]. Through the *Struzik Multiplicative Cascade Model* [34], we can estimate the LHE at singularity $\mathbf{x}_0$ as the slope [48].

Even though the wavelet analysis and LHE can provide us a series of ECG features, it is necessary to increase the accuracy of the induction algorithms through the reduction of parameters. We used *Wrapper* approach in [31] to conduct a search in the wavelet space. Our *Wrapper algorithm* includes a "State" that is a vector of LHE, an initial state (set to empty), a heuristic evaluation through 5-fold cross-validation (repeated multiple times with a small penalty for every ECG feature), and a Hill-climbing search algorithm.

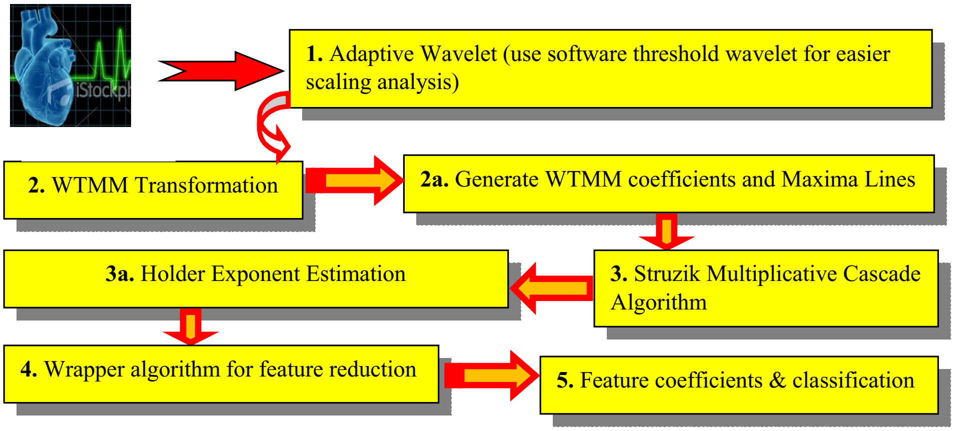

iStockph
1. Adaptive Wavelet (use software threshold wavelet for easier scaling analysis)
2. WTMM Transformation
2a. Generate WTMM coefficients and Maxima Lines
3. Struzik Multiplicative Cascade Algorithm
3a. Holder Exponent Estimation
4. Wrapper algorithm for feature reduction
5. Feature coefficients & classification

Fig. 14. ECG Data Series Feature Extraction Software components

Regarding to heart beat Arrhythmia, we have chosen the following four types: (1) Paced rhythm, (2) Atrial Fibrillation, (3) Nodal (A-V junctional) rhythm, and (4) Ventricular fibrillation. For each of the five rhythms (i.e., Normal (NSR), Paced, A-Fib, Nordal, and V-Fib), we have used the following procedure (see Fig. 14 [48]) to extract the WTMM LHEs that will be used for the input vectors of SVM model. Note that Step 3a (in Fig. 14) does not directly use the "single-value" Holder exponents since we have used statistical analysis based on large amount of MIT-BIH arrhythmia record flows [35]. Thus we have calculated the Probability Densities of different LHEs and then fitted those densities into a Gaussian model.

B. Wireless Loss Recovery for Compressed Transmission

Here we revisit wireless loss recovery issue from another perspective. Although we could use *Particle Filtering / EKF* to achieve good loss compensation, we do not need to achieve 100% loss recovery for continuous ECG waves. However, in the compressed data transmission through wavelet feature extraction, any wavelet coefficient is critical to destination signal reconstruction. In other words, we need a 100% wireless loss recovery rate. We have used two strategies to perform 100% loss recovery: one is *hop-to-hop, localized* retransmission and the other is *end-to-end, code-distribution* based reconstruction.

1) Hop-to-hop Localized Loss Recovery: Data Retransmission (D-R) is a typical approach since it can resend the damaged data packet. Most D-R schemes are based on Internet TCP that uses end-to-end retransmission, and all routers between the two ends do not check / recover packet loss. Although simple and publicly available, TCP cannot achieve optimal MSN error recovery performance due to the following reason: the wireless transmission errors can accumulate exponentially over each hop. To simply illustrate this, assume that the packet error rate of a wireless hop is $\mathbf{p}$, and then the chances of exchanging a message successfully across a single hop is $(\mathbf{1}-\mathbf{p})$. The probability that a message is successfully received across $\mathbf{n}$ hops decrease quickly to $(\mathbf{1}-\mathbf{p})^{n}$.

Inspired by [22] but with 100% recovery consideration, we have designed a localized loss recovery scheme as shown in Fig. 15. First of all, the sensors use a self-organized topology management scheme to get to know which "ripple" it belongs to. The same ripple means the same number of hops to a sink (base station). In Fig. 15, after B sends out a packet to **A** (the closest *sensor* to **B**), **B** sets up a timer T_1. If T_1 expires and **B** has still not received a positive acknowledgement packet from **A**, **B** will retransmit the packet to **A**. If **B** attempts more than **3** retransmissions without success, it regards **A** as unreachable (possibly due to an unreliable wireless link), and it then finds another good path as follows. **B** checks the reachability of **A**'s neighbors **H** and **E**. If either of them is reachable, **B** will transmit the packet to one of them. If both **H** and **E** are unreachable, it goes back one *ripple*, i.e., to **K**. **K** discovers **B**'s neighbors and tries to deliver the packet. Eventually the packet gets through without any loss.

2) End-to-end Loss Recovery through Multi-path Erasure Codes: Using erasure code [39], [40], we can reconstruct m original messages with any m out of n code words. Its basic principle is shown in Fig. 16, where the original bits could be added some redundant bits, and even partial received bits could be used to reconstruct original message. If n is sufficiently large compared to the loss rate, we can achieve high reliability without retransmission [41]. A particular erasure code algorithm called *Reed-Solomon (R-S)* code is used. For R-S encoding process, there exists encoding function $C(X)$ where X is a vector of m messages. Then $C(X)$ will produce a vector of n code words ($n > m$). A linear code has a property of $C(X) + C(Y) = C(X + Y)$. A linear code can be represented with a matrix A, and code word vector for message vector X is simply AX. Encoding is matrix-vector multiplication. Decoding is finding X such that $AX = Z$ for a received code word vector Z. This is to find the solution of linear equation $AX = Z$. This code is very useful, since encoding and decoding are computationally inexpensive. This is especially attractive in resource-constrained sensor networks.

Matrix A should have m linearly independent rows so that linear equation has unique solution, and in turn unique message vector. *Vandermonde Matrix* is typically used [39], [41]. For n by m *Vandermonde matrix* ($n > m$), any set of m rows forms nonsingular matrix. That is to say, whatever set with m rows we may choose, rows in the set are linearly independent. It has the following format:

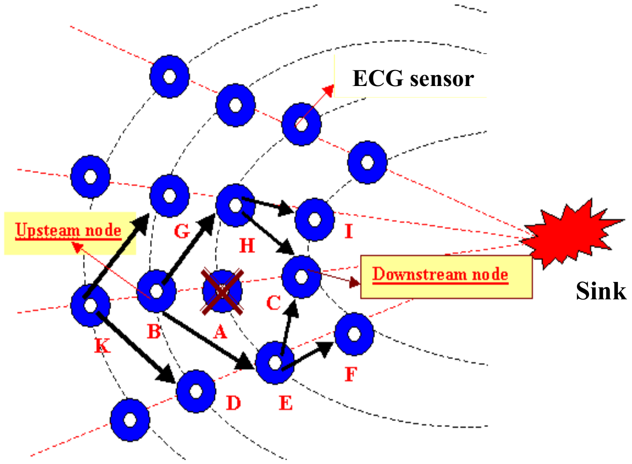


Fig. 15. Ripple-to-ripple loss recovery

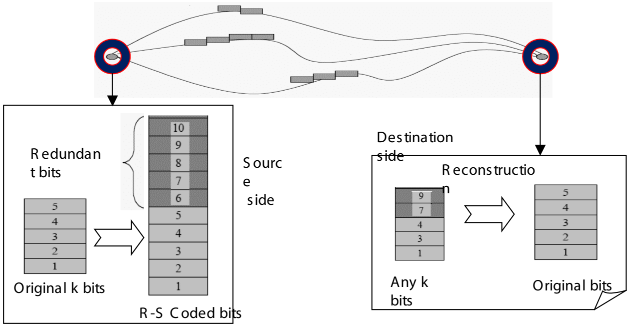
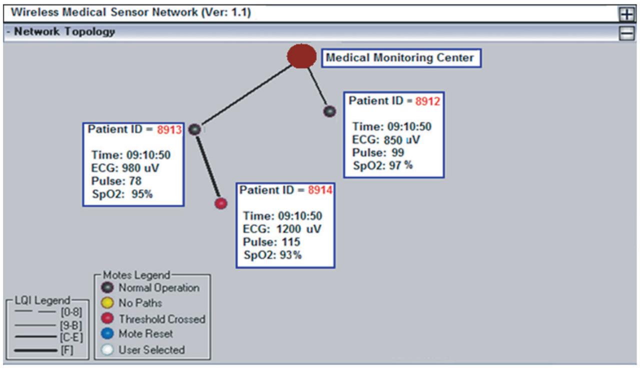


Fig. 16. Multi-path receiving to reconstruct original data via erasure codes

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{m-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{m-1} \\ 1 & x_n & x_n^2 & \cdots & x_n^{m-1} \end{bmatrix} \tag{30}$$

Erasure codes implementation for sensor networks is presented as follows. First, we need to encode the application layer messages (that hold EEG/ECG data) into code words. To reduce communication overhead, we need to put multiple independent code words into a packet. Second, we need to use the concept of extension field [41], where a field is any set of elements which satisfies the field axioms for both addition and multiplication and is commutative division algebra. Field axioms include commutativity, associativity, distributivity, identity, and inverse. An archaic name for a field is rational domain. A field with a finite number of members is known as a finite field or *Galois* field. Extension field is a Galois field whose elements are integers in $[0, p^r - 1]$. For efficient use of bits in packets, we use extension field with base 2. Third, we need an operation table in the sensor's memory for the following reasons. Operations on extension field are not simply addition and multiplication combined with modulo operation. They are polynomial operations with modulo. Rather than performing complex computation on the fly, a table is used to lookup result of operation. For multiplication and division operation, exponent and log values are computed and stored as tables.



Patient ID = 8913

Time: 09:10:50
ECG: 980 uV
Pulse: 78
SpO2: 95%

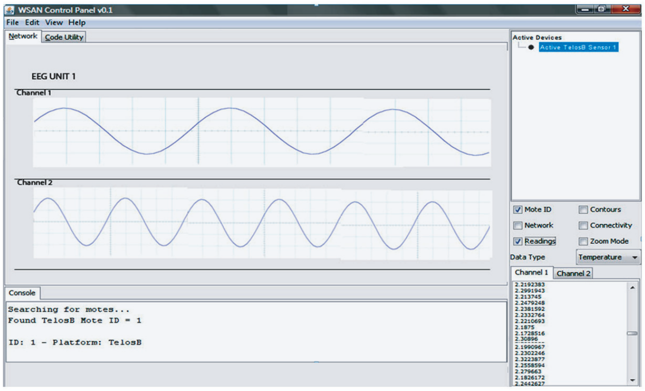
Patient ID = 8912

Time: 09:10:50
ECG: 850 uV
Pulse: 99
SpO2: 97 %

Patient ID = 8914

Time: 09:10:50
ECG: 1200 uV
Pulse: 115
SpO2: 93%

Fig. 17. MSN software GUI (Graphical User Interface)



☑ Mote ID ☐ Contours
☐ Network ☐ Connectivity
☑ Readings ☐ Zoom Mode
Data Type: Temperature

```
Searching for motes...
Found TelosB Mote ID = 1

ID: 1 - Platform: TelosB
```

```
2.2192383
2.2991943
2.213745
2.2479248
2.2381592
2.2332764
2.2210693
2.1875
2.1728516
2.30896
2.1990967
2.2302246
2.3223877
2.2558594
2.279663
2.1826172
2.2442627
```

Fig. 18. EEG signal visualization in the doctor's machine

V. Experimental Results

A. System GUI (Graphical User Interface)

All patients' sensors have the capability of remote wakeup/sleep control. The monitoring center can pull out any sensor's data and visualize them in a friendly GUI, shown in Fig. 17. We have also embedded CodeBlue sensor localization technology [9] in our software. This is GPS-free system with around 3 meters of localization errors.

Fig. 18 shows the GUI for EEG signal capturing. Each EEG channel's data is displayed separately. The EEG signals could be changed easily if the patient blinks the eyes due to electrical signal interference from eyes and scalp. Therefore, our software allows remote "zoom-in" for a specific patient to observe the EEG values with higher resolution over a long term. Fig. 18 shows a button called "Zoom mode" in the right side of the GUI.

Our ECG analysis software can automatically highlight all TRS peaks for doctor's better heart attack decision. It is based on function's localized singularity analysis. Fig. 19 illustrates this effect.

An important feature of our MSN software is that we are able to control the ECG sensors' performance parameters (such as ECG detection threshold) through the remote command transmission from the server to any ECG sensor. We can set up the ECG server (i.e., the MSN workstation) control parameters to change the sensors' detection frequency, i.e., how many ECG values which we should collect in each second. As we know, a higher detection frequency can bring

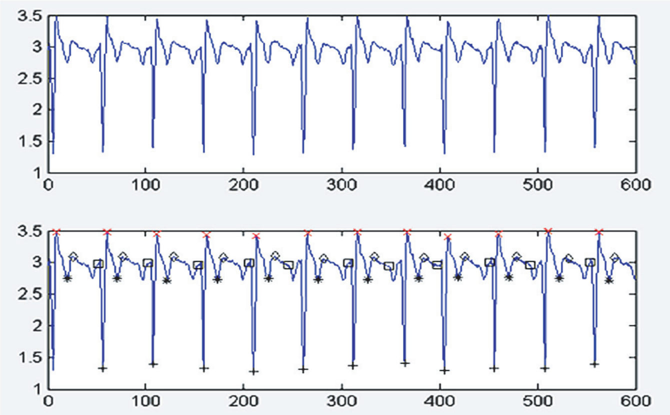


Fig. 19. ECG Data Display (Top) original stream with automatic ECG wave peak finding (Bottom)

higher ECG signal quality; however, it also causes the higher power consumption in each sensor, and more memory storage overhead in each RF board. A good balance is needed. Here we collect ECG values every 0.01 seconds, which is good enough to capture each change of heart beats.

To demonstrate the native impacts of wireless transmission errors on MSN performance, we did not use any error recovery scheme but collected the packet loss rate at different sending rates (5 packets/sec $\sim$ 40 packets/sec) and different mobility speeds (walking - 3 mph, running - 10 mph, driving - 30 mph), network scale = 12 sensors. As shown in Fig. 20(a), an increasing traffic load could cause up to a 19% loss rate. When two neighboring sensors try to send out data simultaneously, a wireless access conflict occurs. Wireless conflicts can cause data errors. On the other hand, the increasing mobility could also lead to higher wireless error rate since mobility could make an established route useless, and during the establishment of a new route, many packets can get lost.

B. Wireless Loss

Besides the aforementioned data error reasons (i.e., MAC-layer wireless conflicts and patients' mobility), other factors that cause packet transmission errors and losses could be stated as follows: (1) *Wireless signal energy loss during propagation is one reason*. It is well-known that the received wireless signal strength decreases when distance from the sender increases. When the received signal is too weak, it is difficult for the receiver to detect the original values. Thus, typically, it just simply discards the received data from its buffer, and this can cause packet loss. (2) *Radio reflection / diffraction / scattering is another reason*. They can damage the wireless signals. An erroneous packet will be discarded by any receiver, and this causes packet loss.

C. Energy Consumption

The top concern for sensor networks is energy consumption since it determines the batteries' lifetime. Our energy measurement includes two places as follows:

(1) PSoc chip with source EEG/ECG compression algorithms through wavelets and pre-transmission signal processing calculations. Fig. 4 has shown the architecture of PSoc interface that handles medical signal pre-processing before the wireless transceiver sends them out. Such a local calculation consumes certain energy. Cypress MicroSystems [13] has made a Power calculator available to the PSoC users, and this helps developers to estimate the power consumption in the chip. Based on the Power calculator results, we can estimate the current and power consumption for our thresholding and cardiac monitoring system.

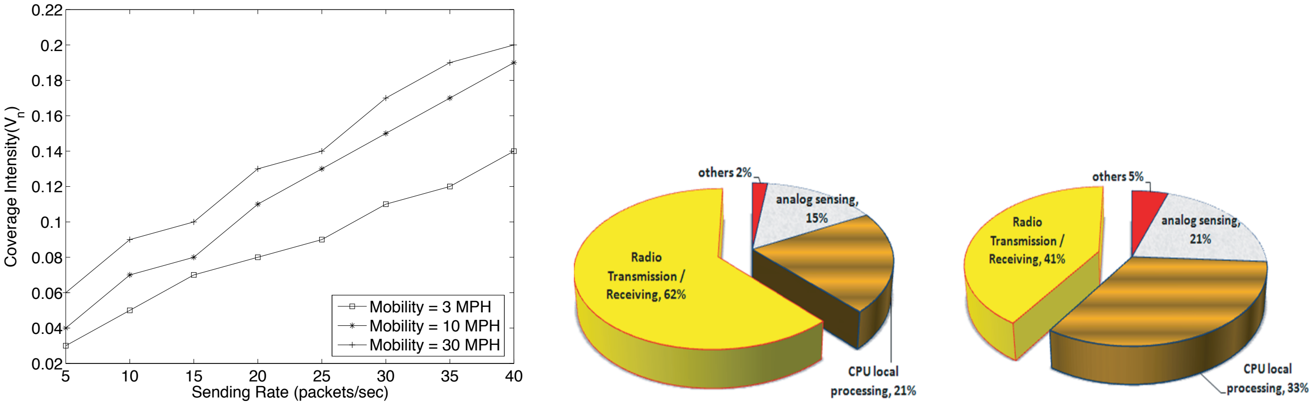
(2) The RF mote with CPU / memory and wireless transceiver. It performs all sensor network wireless communication protocols such as multi-hop relay, erasure codes multi-path maintenance, sensor localization, etc. Because it is difficult to *directly* measure energy consumption of sensors hardware, we have thus resorted to an accurate simulator called PowerTossim [14] where hardware peripherals (such as the radio, EEPROM, LEDs, and so forth) are instrumented to obtain a trace of each device's activity during the simulation run. To scale up to large numbers of sensor nodes, PowerTOSSIM runs applications as a native executable and does not directly simulate each node's CPU at the instruction level. PowerTOSSIM employs a code-transformation technique to estimate the number of CPU cycles executed by each node. Finally, the trace of each node's activity is fed into a detailed model of hardware energy consumption, yielding per-node energy consumption data. This energy model can be readily modified for different hardware platforms.

1) The energy saving performance of compressed ECG transmission: We are interested in the energy saving effect of compressed signal transmission through *multi-scale wavelet feature* extraction (Section IV-A). (1) First, we investigated the raw ECG stream case without compression. Fig. 20(b) is our energy measurement result in relative percentage. It reflects the general case in most sensor networks: most energy is consumed in RF communications instead of in local CPU processing due to the expensive antenna issuing power and many hops of data relay. (2) Then we introduced wavelet compression into source data and transmit only the decomposition coefficients. Although the 100% error/loss recovery (through network single-hop retransmission and erasure code multi-path reconstruction) could consume more energy than non-compression case where the network doesn't participate in the loss recovery (the destination does the particle filtering-based prediction), the total wireless transmission energy is largely reduced through data compression. As shown in Fig. 20(c), we can see that the percentage of different energy components has shifted quite a lot from wireless transmissions to local CPU (PSoc) processing.

D. Wavelet-based feature extraction and classification

Our wavelet analysis has satisfactory signal reconstruction performance (after using 100% error recovery, see Section IV-B). Fig. 21 is our ECG signal reconstruction result. They have very good matching.

To validate our LHE / WTMM -based feature extraction and classification (Section IV-A), we have used the following ECG data sets from heart attack patients: (1) 10 Normal Sinus Rhythms (NSR) were recorded from real ECG sensors; (2) Other Arrhythmia came from PhysioNet [35], which provides



(a) Packet Loss Rate (b) Before multi-scale wavelet processing (c) After multi-scale processing

Fig. 20. (a) Packet Loss Rate during different sending rates and patient mobility speeds, (b) Before multi-scale wavelet processing, and (c) After multi-scale processing

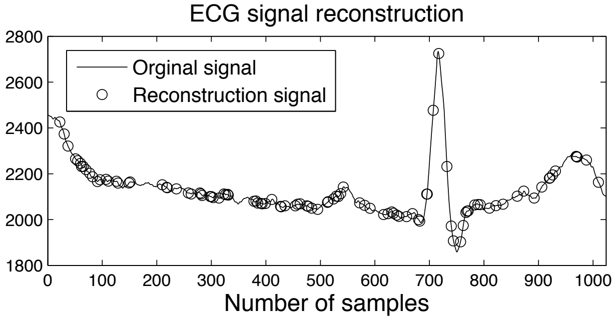


Fig. 21. ECG reconstruction after receiving wavelet coefficients

a set of databases that group records of one or more digitized ECG signals, as well as a set of their corresponding beat and rhythm annotations. Especially, we have used (a) PhysioNet MIT-BIH Noise Stress Test Database that contains typical noises in ambulatory ECG recordings, and (b) PhysioNet MIT-BIH Arrhythmia Database, which is used to study the different types of arrhythmias. As we mentioned in Section IV-A, the purpose of ECG feature extraction is not only for data compression. More importantly, the local sensor can coarsely determine whether the ECG signals indicate abnormal heart beat patterns. Our SVM-based ECG classification results are shown in Fig. 22(a). In that diagram, we have also compared our classification performance to two of the best ECG classification algorithms, i.e., Bayesian Classifier [37] and Decision Tree [36]. Although the accuracy for NSR is similar between ours and others, the accuracy to identify arrhythmia is higher in our scheme. More importantly, our algorithm can use WTMM /Wrapper to efficiently extract multiple features from a "large-scale" ECG database within a reasonable small calculation time.

E. EEG (brain signal) compressed transmission

Besides ECG (heart beat signals), we have also investigated the compression effects of EEG brain signals. Fig. 23 (upper) shows compressive measurements of one channel EEG signals with 16 sensing elements at random sampling rate, resulting in a compression ratio of 1024/120, and Fig. 23 (lower) reconstructed EEG signals from the compressively sampled data. Fig. 24 is the 2D scalp map projection of the first 6 independent components of the compressively sampled data after interpolation.

F. RFID performance

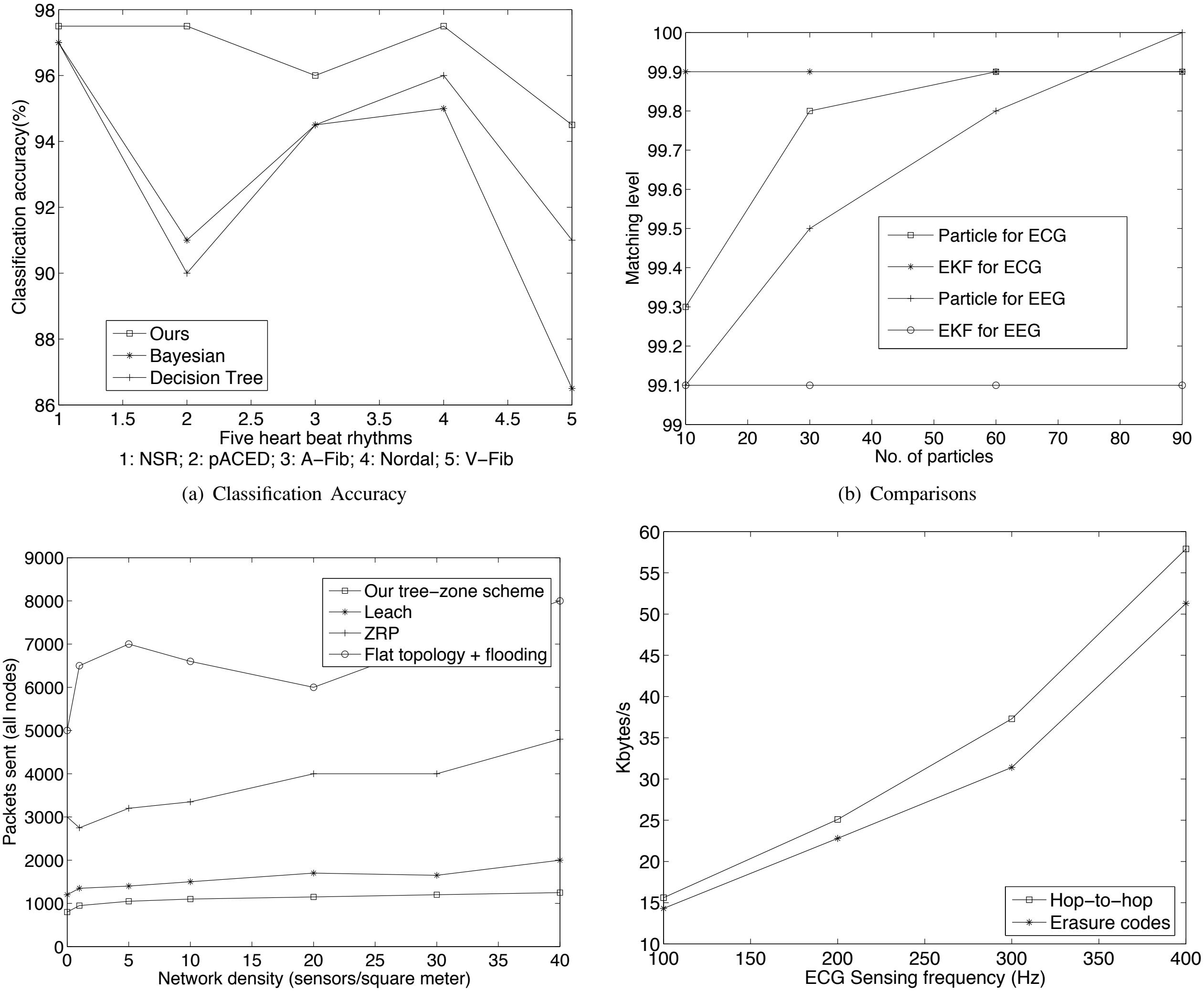
Section II-E has discussed our designed RFID principle. It plays an important role in patient's medicine management. We have built medical database with the collected RFID data such as patient's medicine information. We have also built the GUI to automatically display warning message if there is overdose case or the patient takes wrong medicine. Please refer to [47] on our results on RFID performance.

G. Extended Kalman Filter (EKF) for loss prediction

We have utilized the accomplishments of Sameni *et al.*, [44] who were able to linearize the ECG model and apply the EKF to it in Matlab. Their usage was noninvasive fetal ECG extraction. Their scheme can be applied in a completely different area, wireless data recovery. The actual ECG data came from the PhysioBank database of physiological signals [35]. A sample of one of these normal ECG signals can be seen in Fig. 25 (the heavier curve). In this experiment, we assumed that the lost rate is 10%. Fig. 25 bottom curve is a received ECG stream with lost data. We have marked all missed data. Fig. 25 upper curve is our EKF-recovered ECG data. We have increased the repaired ECG stream's magnitude for certain level to clearly compare those two streams. As we can see, the repaired stream keeps all ECG features and can be used for further pattern recognition to detect abnormal heart beats.

H. Particle filtering for loss recovery

To measure the matching level between the repaired wave and original sent one, we have used the function **xcorr** (i.e. the Matlab cross-correlation function). It measures similarity (in the range of 0∼100%) between two given signals, as a function of the relative time-shift (we used 0 in our case) between them.



(a) Classification Accuracy

(b) Comparisons

(c) Communication Overhead

(d) Network Traffic

Fig. 22. (a) Normal/Arrhythmia Classification Accuracy, (b) Comparisons between Particle Filter and EKF, (c) Hop-to-hop Error Recovery: Communication Overhead, and (d) Network Traffic Load for Two Loss Recovery Schemes

To compare the Particle Filter and EKF, we have changed the number of "particles" (i.e., system state trajectories) to see the matching level between those two signals. For ECG (heart beat), the two loss recovery schemes have almost the same performance after we use particles of more than 60 (see Fig. 22(b) top two curves). However, for EEG (brain signals), we have found that EKF has worse performance than Particle filter (see Fig. 22(b) lower two curves). One possible reason could be due to the following fact: it is understood that particle filter has stronger capability to predict and correct signals with non-linearity and non-Gaussian noise, while EKF does not have stable prediction performance when signal non-linearity is too high. While ECG has typically good regularity due to its direct human pulse measurement, the EEG measurement can be more difficult and have less linearity due to the eye blinking and head movement. On the other hand, our PSoc interface (see Section II-A) has already filtered most of the noise in ECG/EEG signals, but it cannot distinguish between scalp radiation and eye-movement bio-electric signals. Thus EEG has higher variations than ECG stream.

We have also found that the implementation of Particle filter is more complex when considering more particles. The corresponding PSoc memory overhead is larger than EKF "states" maintenance. But thanks to the fast increasing memory density in the last decade, the storage issue is not as critical as CPU calculation overhead.

I. Complete loss recovery (via wavelet): Communication overhead

1) Hop-to-hop loss recovery: In Section IV-B, we discussed 100% loss recovery through network-assisted localized retransmission. But its efficiency depends on our proposed ripple-based topology management structure. A crucial metric for the complexity of sensor network topology control protocols is the amount of control packets transmitted between individual nodes. We compare our ripple-based (also called tree-zone) routing architecture to other three topology control schemes that are based on other ad hoc routing protocols including flat topology, LEACH [43], and ZRP [42], and calculate the number of control packets transmitted in the

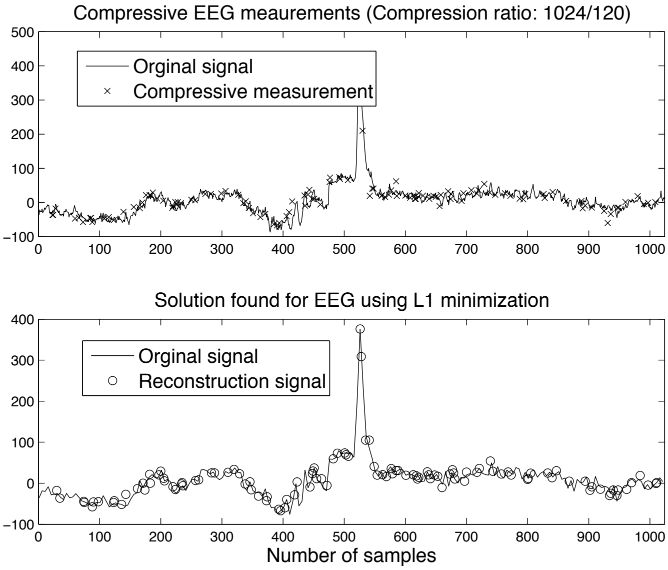


Fig. 23. Results of the compressively sampled EEG signals

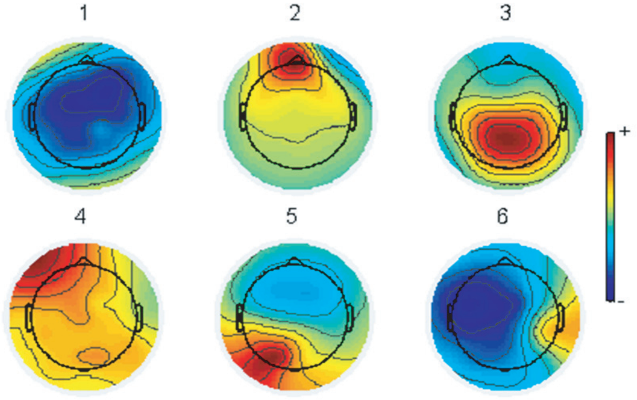


Fig. 24. 2D scalp map projection

routing architecture. As shown in Fig. 22(c), both LEACH [43] and our scheme have a satisfactory control packets communication overhead compared to others. LEACH [43] adopts a cluster-based routing scheme. However, it asks each cluster head to directly communicate to the sink, which may not be practical since a long distance of wireless communication consumes too much energy. Our routing scheme only requires each sensor communicates with the next-ripple, which limits communication in a small area and thus saves much energy.

2) Compared to Erasure codes: We have also compared the above hop-to-hop approach (Section IV-B)) to erasure codes approach (section IV-B). We have noticed the following communication overhead sources: (1) Hop-to-hop approach needs to maintain the tree-ripple topology in order to quickly find out next-ripple sensor to forward data. In addition, the local retransmission could take certain attempts to get through. The patients' mobility causes the difficulty of building topology. (2) Erasure codes need to add redundant bytes to each packet. The multi-path transmission adds more communication traffic for the same ECG/EEG data. However, the erasure codes do not need middle nodes' retransmission and the reconstruction occurs only in the destination side. Fig. 22(d) is the result of

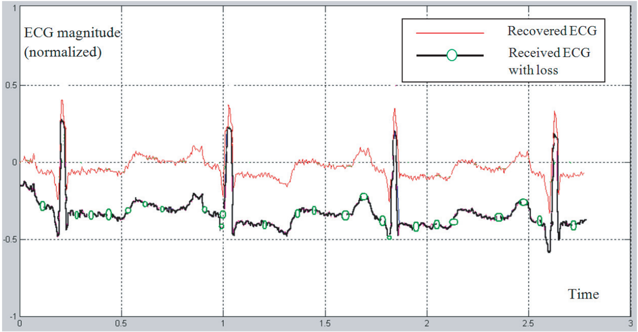


Fig. 25. EKF-based loss compensation

the following scenario: 20 patients, 4 hops maximum from the furthest patient to the sink. Patient mobility speed is walking case (1m/s). The packet error rate is 15%. It indicates the slightly lower network traffic overhead in erasure codes case. This implies the importance of avoiding middle-in-the-network retransmissions.

VI. Conclusions

We have presented our mobile tele-healthcare research results in this paper. The following conclusions can be drawn from our above discussions. (1) the tiny sensors can achieve long-distance bio-signal monitoring through self-healing multi-hop communication architecture. It is necessary to use certain interface (we used PSoc) to pre-process different medical signals (from stream speed and priority perspective) before assembling into RF transmission packets. (2) Our Psoc interface has used wavelet-based signal decomposition to obtain the signal feature parameters (coefficients). Those features can also be used for normal/abnormal signal classification and normal signal will not be transmitted. (3) From wireless loss recovery viewpoint, if no 100% loss recovery is required (such as in uncompressed ECG stream transmission case), a destination-only particle filter or EKF's loss prediction / compensation could work efficiently from our above discussions. But for compressed transmission, each coefficient is important from signal reconstruction viewpoint. TCP cannot be used although they can achieve 100% recovery. Therefore, we proposed a ripple-based local recovery and erasure-codes-based approach to guarantee all data's safe arrival. (4) The sensor network based tele-healthcare system could be integrated with RFID technology to achieve patient medicine-taking monitoring. We have built reprogrammable RFID reader for future security and communication protocol enhancement purpose.

References

[1] National Coalition on Health Care: http://www.nchc.org/facts/cost.shtml, accessed in June 2005

[2] P. Palatini, "Need for a Revision of the Normal Limits of Resting Heart Rate," *Hypertension*, vol. 33 (2), pp.622-5, Feb 1999.

[3] G. R. Schwartz, *Principles and Practice of Emergency Medicine*. King of Prussia, PA: Rittenhouse Book Distributors, 1999.

[4] R. E. Behrman, *Nelson Textbook of Pediatrics*. Philadelphia, PA: W.B. Saunders Company, 2000.

[5] A. V. Chobanian et al., "The Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure," *Hypertension*, vol.42, pp.1206-52, Dec 2003.

[6] E. Jovanov, A. Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, “Stress Monitoring Using a Distributed Wireless Intelligent Sensor System,” *IEEE Engineering in Medicine and Biology Magazine*, Vol. 22, No.3, May/Jun. 2003, pp. 49-55.

[7] E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen, “A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation,” *Journal of NeuroEngineering and Rehabilitation*, 2:6, pp. 18-19. Mar. 1, 2005.

[8] J. Welch, F. Guilak, and S. D. Baker, “A Wireless ECG Smart Sensor for Broad Application in Life Threatening Event Detection,” in *Proc. of EMBS ’04*, pp. 3447-3449.

[9] D. Malan, T. R. F. Fulford-Jones, M. Welsh, and S. Moulton, “CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care,” in *Proc. of WAMES 2004*, pp. 12-14.

[10] M. J. Mathie and B. G. Celler, “A System For Monitoring Posture And Physical Activity Using Accelerometers,” in *Proc. of EMBS ’01*, pp. 3654- 3657.

[11] M. J. Mathie and B. G. Celler, “Moteiv: Wireless Sensor Networks,” Available at: http://www.moteiv.com, Accessed: June 2005.

[12] C. Otto, J. P. Gober, R. W. McMurtrey, A. Milenkovic, and Emil Jovanov, “An Implementation of Hierarchical Signal Processing on a Wireless Sensor in TinyOS Environment,” *Proc. of the 43rd ACM Southeastern Conference*, Mar. 2005, Vol. 2, pp. 49-53.

[13] Cypress CY8C27443 Datasheet, in http://www.cypress.com

[14] PowerTossim software: http://www.eecs.harvard.edu/ shnayder/ptossim/.

[15] Crossbow Inc. http://www.xbow.com.

[16] Chipcon Inc. http://www.chipcon.com

[17] Texas Instruments: http://www.ti.com

[18] Ember Inc., http://www.ember.com

[19] Zigbee wireless communication standard: http://www.zigbee.org

[20] M. Welsh, B. Chen, *et al.*, CodeBlue: Wireless Sensor Networks for Medical Care, Harvard University. 2006.

[21] $SkyeRead^{TM}$ M1-mini is reader: see the following site for more details: http:// www.skyetek.com/

[22] F. Hu, “Modeling and Allocation of Connections in Next-generation Wireless Networks,” PhD Thesis, Clarkson University. Jun. 2002.

[23] I. F. Akyildiz, W.J. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, Aug. 2002, pp. 102-114.

[24] A. Sinha and A. Chandrakasan, “Dynamic Power Management in Wireless Sensor Networks,” *IEEE Design & Test of Computers*, Vol. 18, No. 2, Mar.-Apr. 2001.

[25] Sensor node, From Wikipedia, the free encyclopedia, vistited in Nov 2008, http://www.wikipedia.org/

[26] M. J. Coates, ”Distributed Particle Filtering for Sensor Networks,” in *Proc. of IPSN*, Apr. 2004.

[27] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Series: Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[28] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Available from http://www.cs.unc.edu/ welch/kalman/kalmanIntro.html.

[29] P.E. McSharry, G.D. Clifford, L. Tarassenko, L. A. Smith, “A dynamical model for generating synthetic electrocardiogram signals,” *IEEE Transactions on Biomedical Engineering,* Vol. 50, No. 3, Mar. 2003, pp. 289 - 294.

[30] Wavelet Tutorial, visited in Nov of 2008: http://users.rowan.edu/ polikar/WAVELETS/WTtutorial.html.

[31] A. J. Joshi, (PhD Progress Report), “Data Mining of Biomedical Signals,” Dept. of CSE, Indian Institute of Technology, Aug. 27, 2005.

[32] C. Burges, “A tutorial on support vector machines for pattern recognition,” Data Mining and Knowledge Discovery, Vol. 2, No. 2, pp.121-167, 1998.

[33] Wavelet singularity analysis: see http://hesperia.gsfc.nasa.gov. Visited in Nov of 2008.

[34] Z. R. Struzik, “Determining local singularity strengths and their spectra with the wavelet transform,” *Fractals*, 8(2), 2000.

[35] The Research Resource for Complex Physiologic Signals (2004, Sept. 8). PhysioNet [Online]. Available: http://www.physionet.org.

[36] R. Le Blanc, “Quantitative analysis of cardiac arrhythmias,” CRC: Critical Review Biomed. Eng. vol. 14, no. 1, pp. 1-43, 1986.

[37] T. M. Mitchell, Machine Learning, 1997.

[38] C. Wan, A. T. Campbell, and L. Krishnamurthy, “PSFQ: a reliable transport protocol for wireless sensor networks,” In *Proc. of ACM WSNA ’02*, pp. 1-11.

[39] S. Kim, “Efficient Erasure Code for Wireless Sensor Networks”, available from: (visited in Nov of 2008), http://www.cs.berkeley.edu/

[40] J. Blomer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, “An xor-based erasure-resilient coding scheme,” ICSI Technical Report tr-95-048.

[41] S. Parikh, “On the use of erasure codes in unreliable data network,” MS thesis, Dept of EECS, MIT, 2001.

[42] Z. Haas and M. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, *IETF Internet draft for the Manet group*, June 1999.

[43] W. R. Heinzelman, “Application-specific protocol architectures for wireless networks,” PhD Thesis, MIT, Jun. 2000.

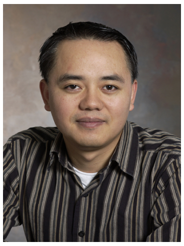
[44] R. Sameni, M. B. Shamsollahi, C. Jutten, M. Babaie-Zade, “Filtering noisy ECG signals using the extended Kalman filter based on a modified dynamic ECG model,” *Computers in Cardiology*, 2005,pp.1017 - 1020.

[45] TinyOS and NesC: http://www.tinyos.net.

[46] F. Hu, M. Jiang, L. Celentano, and Y. Xiao, “Robust medical ad hoc sensor networks (MASN) with wavelet-based ECG data mining,” (Elsevier) Ad Hoc Networks, Vol. 6, No. 7, pp. 986-1012, Sep. 2008.

[47] F. Hu, L. Celentano, and Y. Xiao, “Error-Resistant RFID-Assisted Wireless Sensor Networks for Cardiac Tele-healthcare,” Wireless Communications and Mobile Computing (WCMC) Journal, Vol. 9, No. 1, pp. 85-101, Jan. 2009.

[48] F. Hu, M. Jiang, Ma. Wagner, and D. Dong “Privacy-Preserving Telecardiology Sensor Networks: Towards A Low-cost, Portable Wireless Hardware / Software Co-design ”, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 11, No. 6, Nov. 2007, pp. 617-627.



Fei Hu is currently an associate professor in the Dept. of Electrical and Computer Engineering at the Univ. of Alabama, Tuscaloosa, AL, USA. His research interests are wireless networks, wireless security and their applications in Bio-Medicine. His research has been supported by NSF, Cisco, Sprint, and other sources. He obtained his first Ph.D. degree at Shanghai Tongji Univ., China in Signal Processing (in 1999), and second Ph.D. degree at Clarkson Univ. (New York State) in the field of Electrical and Computer Engineering (in 2002). He has published over 100 journal/conference papers and book (chapters). He is also the editor for over 5 international journals.

Yang Xiao (SM’04) is currently with Dept. of Computer Science at The Univ. of Alabama. He currently serves as Editor-in-Chief for *International Journal of Security and Networks* (IJSN), *International Journal of Sensor Networks* (IJSNet), and *International Journal of Telemedicine and Applications* (IJTA). He serves as an associate editor for several journals, e.g., *IEEE Transactions on Vehicular Technology*. His research areas are security, telemedicine, sensor networks, and wireless networks. He has published more than 300 papers in major journals, refereed conference proceedings, book chapters related to these research areas.

Qi Hao (M’06) is an assistant professor of Dept. of Electrical and Computer Engineering at The Univ. of Alabama. He earned his Ph.D. from Duke Univ. for the work in design and implementation of wireless distributed infrared sensors in 2006. His post-doctoral training in Visualization and Virtual Environment Center at The Univ. of Kentucky was focused on 3-D computer vision for human tracking and identification. His current research interests include compressive sensing, intelligent wireless sensor networks, and biomedical image processing.