Mobile Health Monitoring Through Biotelemetry

Andrew D. Jurik[*] Jonathan F. Bolus[†] Alfred C. Weaver[*]
Benton H. Calhoun[†] Travis N. Blalock[†]

[*]Department of Computer Science, University of Virginia, Charlottesville, VA, USA
[†]Department of Electrical & Computer Engineering, University of Virginia, Charlottesville, VA, USA
{adj3t, jfb9e, acw, bhc2b, tnb2a} @virginia.edu

ABSTRACT

As the population ages and the risk of chronic disease increases, the cost of healthcare will rise. Technology for mobile telemetry could reduce cost and improve the efficiency of treatment. In order to achieve these goals, we first need to overcome several technical challenges, including sufficient system lifetime, high signal fidelity, and adequate security. In this paper we present the design, implementation, and evaluation of a Mobile Biotelemetric System (MBS) that addresses these remote medical monitoring challenges. MBS comprises a custom low-power sensor node that accurately collects and analyzes electrocardiogram (ECG) data, a client service with a multifaceted *policy engine* that evaluates the data, and a web portal interface for visualizing ECG data streams. MBS differs from other remote monitoring systems primarily in the policy engine's ability to provide flexible, robust, and precise system communication from end-to-end and to enable tradeoffs in metrics such as power and transmission frequency. We show that, given a representative set of ECG signals, policies can be set to make the operation of the hardware and software resilient against transient ECG conditions. Further, we incorporate state-of-the-art security practices to safeguard our data and foil common attacks.

1. INTRODUCTION

Chronic illness, particularly heart disease, is a problem that has a dramatic impact on the productivity of affected individuals and the cost of healthcare. As the risk of chronic illness increases with age and with projections predicting an increasingly elderly population, the costs of healthcare are going to rise. The prevention and effective management of chronic diseases may be the only lasting solution, and remote medical monitoring will be an integral part of that approach.

In order to address the challenge of chronic illness, it is natural to leverage technology. The confluence of small body sensors and standardized service interfaces over the Internet (e.g., Simple Object Access Protocol) provides a new platform for applications in a wide variety of domains, including healthcare, the military, emergency response, and consumer entertainment. The miniaturization and maturation of sensor technology could very well lead to the instrumentation of not only the world around us, but also our very selves. With a multitude of sensors that can deliver data on demand, the way we interact with technology will change dramatically—and could lead to early detection and adaptive, responsive control of diseases.

With personal information becoming more available, we must find how best to leverage the usefulness of physiological information while protecting it at the same time. Since body sensors detect clinically significant—and potentially personal—data, the applications to healthcare are especially compelling and challenging. The electrocardiogram, in particular, has been well established as having diagnostic relevance [3] and as a potential biometric [7]. The ECG, as shown in Figure 1, reveals the electrical activity of the chambers of the heart, reflecting the muscle's life-sustaining nature. Furthermore, other signals may be derived from the ECG including respiration and blood pressure (the former through analysis of baseline drift and the latter through the time between a heartbeat and a pulse at an extremity). We propose our Mobile Biotelemetric System (MBS) architecture to respond to physiological (ECG) data adaptively through a policy engine that contains several processing algorithms for providing and arbitrating information throughout the system.

Our thesis is that our hierarchical framework is an effective and flexible way to deliver and manage physiological data for medical monitoring purposes to involved stakeholders, including the wearer of the sensor and other concerned entities such as doctors or loved ones. We show that the system detects heartbeats with high accuracy and the policy engine, through appropriate policies, can be made robust against noise. We posit that allowing decisions at higher tiers (i.e., made by users or devices with more computational resources) to affect the operation of the sensors is beneficial, in that the system supports both local goals (e.g., low power consumption) and adjustments for global eventualities (e.g., emergency scenarios) that may override local decision-making. The policy engine, which resides in the mobile device software with interfaces to the sensor and to the web portal, monitors the connection to the sensor and interprets the data it receives to make decisions on whether—and how—the overall system should react if an exceptional event occurs.

BodyNets'09 April 1–3, 2009, Los Angeles, CA, USA
Copyright 2008 ICST 978-963-9799-41-7.

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

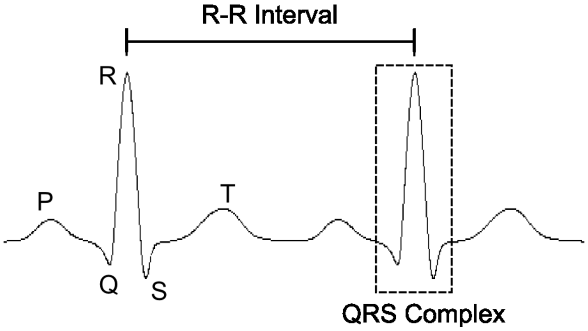


Figure 1. A typical ECG contains five common deflections whose distances between one another can vary in time and intensity based on many factors, including physiology and activity level.

Our main contributions lie in the emergent properties of the integration of diverse components for monitoring. We list them as follows:

- A policy engine for monitoring sensor connections and detecting and evaluating variations in heart rate
- A methodology for providing efficient tradeoffs based on the use of physiological information as an input to the policy engine
- A threat analysis of potential attacks and the technologies that MBS uses to cope with them
- An end-to-end system implementation and evaluation using our custom ECG sensor, two mobile device instances (PDA and laptop), and a web portal.

2. RELATED WORK

Our MBS solution draws from many fields of research, including mobile computing, body area networks, wearable computing, and computer security, and we explore some of the efforts in these areas in this section. To put MBS into some context, it is not the first remote medical monitoring solution, nor is it the most comprehensive. Several companies offer proprietary systems, including CardioNet, Honeywell, Medtronic, and Biotronik, that can accommodate more sensors or have a larger monitoring network. Those systems, however, generally come with specialized equipment from end-to-end, whereas MBS is designed with open standards and protocols where possible.

Several researchers have developed ECG monitoring devices, primarily for healthcare and athletic-training reasons. Park et al. [15] present an ECG monitoring system with comparable power consumption to MBS but do not discuss any signal processing or the implications of how their system is meant to be used. Anliker et al. [1] create a wristwatch system, AMON, with multiple sensors, including an ECG sensor, that is meant to provide a multi-faceted sensor profile. They acknowledge, however, that the ECG "provides poor or no results" due to the signal's derivation from the wrist where the signal-to-noise ratio is low. Lucani et. al [10] develop an ECG monitoring device for telemedicine applications but with different design goals from MBS (for example, in their system, power and form factor are not large concerns as two AA batteries are used on the sensor). The authors additionally do not discuss the user interface of the system. Lorincz et al. [9] establish a software framework, CodeBlue, that operates as an information plane in a disaster scenario, allowing devices to discover each other and report events. The design of MBS does not focus on triage situations, but rather on the utility of a particular mobile device coupled with its sensor for long-term monitoring. Gyselinckx et al. [6] describe their Human++ research program for health monitoring applications in body area networks. They have developed a flexible substrate with a bandage-sized form factor that has the potential to make the sensor lighter and more comfortable to wear. Innovations in wearable computing and "smart" clothes [4,11,18] have increased the usability and convenience of body sensors but usually do so at the cost of accuracy.

Other researchers have used the ECG as an offline (i.e., post-data-collection) identification mechanism based on feature extraction. Since MBS's ECG signal is derived from chest leads due to our design goals of small size and high signal fidelity on the sensor, feature extraction is likely not a viable option at this stage. Signals from electrodes placed close to the heart are subject to waveform changes when the electrodes are displaced by even as little as 10 mm [3]. Biel et al. [2] and Israel et al. [7] use the standard 12-lead ECG to characterize the signals and identify individuals after data collection (i.e., not in real time). Wübbeler et al. [19] found that a three Einthoven (limb) lead system achieved an equal error rate of 2.8% for verification and an accuracy of 98.1% for identification. MBS's goal is not identification of an individual (authentication on the mobile device is done through other means), but rather to gain knowledge about the ECG.

Kumar et al. [8] have researched the use of biometrics in securing shell login sessions and Poon et al. [16] developed a means to secure body area sensor networks based on R-R (also called interbeat or peak-to-peak) intervals. Neither has demonstrated the feasibility of the techniques in practice, as the former requires a resource-rich Linux workstation with continuous fingerprint and face detection technology and the latter adopts R-R intervals as a biometric, which are not unique and depend largely on transient factors such as activity level. MBS implements security of mobile device data largely through the policy engine, which can be customized to address anomalous events by changing the mobile device state.

3. POLICY ENGINE

The policy engine functions as the core of the functionality mechanism on the mobile device. It enhances the robustness of the system against pervasive problems such as noise, radio irregularity, unstable electrode contacts, and motion artifacts. The policies represent a mapping from events to actions. We separate the policies into two subsets: those based on the quality of the signal and those that pertain directly to the physiological data received. For example, the set of events in our implementation are {Disconnection, Signal timeout, Low heart rate, Sensor removed, Low battery on sensor} and the set of actions include {Change Mobile Device State, Send Message to Portal, Send Message to Sensor, Ignore}.

These events and actions are not comprehensive, and a single event can be mapped to multiple actions. While in our implementation we do not specifically focus on sensors other than ECG sensors, we can foresee the idea of policy-

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

driven operation to other body sensors. Accelerometers, for example, could trigger a policy action when acceleration along a particular axis, modulo noise, exceeds some threshold. A blood glucose meter may enact a policy action when connected or disconnected. Sensors in general share many overlapping characteristics that could be used as input to a policy engine regardless of the particular sensor instance invoked.

Policies are defined in a variety of ways. There is a user interface that can modify policies at run-time with an administrator account, otherwise policies are configured per user such that they get compiled into the object file statically. This strategy allows sufficient flexibility without dealing with the risk of lost or modified external policy files that would be read in while the program is running.

In terms of implementation, each policy is inherited from an abstract policy class, which contains fields for the user interface form, an enabled flag, and a mobile device state. A factory method makes the events, and a subroutine processes the events to determine if an action is warranted (that is, if a callback hasn't already precipitated an asynchronous action). For example, the *Signal Timeout* event relies on a timer with a callback that performs an action if time expires. Adding or modifying a policy requires a new inherited class with associated implementations of inherited methods, a modification to the user interface, the factory method, and the centralized processing subroutine.

3.1 Operational Modes

One key crosscutting concern is the management of resources across the system to minimize latency and power consumption while maximizing signal fidelity. In emergency scenarios, sacrificing system resources in favor of saturating bandwidth and providing full signal information is a reasonable tradeoff. Although it may be possible for the sensor to infer such scenarios, flexibility is gained and false positives are reduced in allowing the user to override the flow of information dynamically.

In our ECG setting, both the sensor and the MBS client service operate in one of two distinct modes: *heartbeat* or *waveform* mode. They operate in tandem, such that messages are exchanged when transitioning between modes. The heartbeat mode, which transmits the preceding peak-to-peak time when a QRS complex is detected, has a higher computation cost but a lower communication cost. The ECG signal is applied to a variation of the Pan-Tompkins algorithm [13] in order to detect heartbeats. The waveform mode, however, sends the sampled waveform directly to the mobile device without any substantive processing. This mode reduces computation, but increases the required amount of communication.

3.2 Policy Events

Policy events intend to capture situations that could affect the interpretation of the sensor data. The *Disconnection* event occurs when the connection is forcibly closed, which may be caused by the sensor being turned off or the mobile device going out of range of the sensor. The *Signal timeout* event occurs when a connection is still established, but a data packet has not been sent within a configurable timeout period. This umbrella event is used to guard against failure of the radio (or any component on the sensor before the data gets sent from the microcontroller to the radio).

The *Low heart rate* and *Sensor removed* events are detected based on the contents of the data packets. The *Low heart rate* event, in particular, is derived directly from the QRS complex detection algorithm. Within the source code, the *Low heart rate* event is parameterized by a window of R-R intervals such that transient results and long-term trends may be traded off. We analyze the effects of this window in Section 5.3. For example, if the QRS detector misses a beat or the Bluetooth connection is weak, a larger window may obfuscate the effect. This approach has the disadvantage that the policy engine is less responsive. The *Sensor removed* event is detected when the signal flatlines high or low for a period of time as a result of noise. The *Low battery on sensor* event is detected when the voltage monitor in the processor detects a voltage below 2.65 V, a 12% drop from the normal supply voltage of 3.0 V. While more events can be recognized, we restricted them to this subset as they have proved the most useful in practice. The design of the software is such that instances of events inherit from an abstract base class and a centralized routine polls event functions each time a sample is received. Alternatively, events may have callback functions that occur based on timers.

The events are periodically monitored based on their occurrence rate. For example, *Signal timeout* has a running timer that gets reset every time data is received. The *Disconnect* event is incorporated into (and dependent upon) the wireless protocol. The *Low heart rate* event and *Sensor removed* event analysis is triggered every time the signal is delivered to the MBS service. The *Low battery* event is communicated in the form of a message sent from the sensor (which has a voltage detector on-chip).

3.3 Policy Actions

Policy actions explicitly permit the tradeoffs between goals. While many policy actions may be implemented, we believe the most useful are changing the device state and sending messages to other tiers of the system. These actions working in concert enable bidirectional communication and allow remote control of sensor-level metrics and different data to appear on the mobile device's and web portal's user interfaces.

Changing the mobile device's state may be as simple as displaying a notification to the user of a particular event or as complex as locking out the mobile device. The policy engine can be customized to offer file protection in a mobile device. In a military application, for example, if the presence of ECG data is used to verify proximity to a mobile device and suddenly data is no longer being received, an appropriate action may be to lock the mobile device for further reauthentication. The ability to change events, actions, and the mapping between events and actions makes our system flexible. The policy engine is the invariant link between an array of applications in multiple domains.

3.4 Sources of Noise

Compensating for the inconsistencies in the ECG signal presents perhaps the greatest single challenge to interpreting the MBS policies correctly. Sources of noise are encountered at every stage of data acquisition until the data is digitized. Power line (60-Hz) interference, muscular contractions, electrode movement, and analog-to-digital converter noise all perturb the ECG signal. If an electrode is removed (intentionally or unintentionally) the ECG signal becomes

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

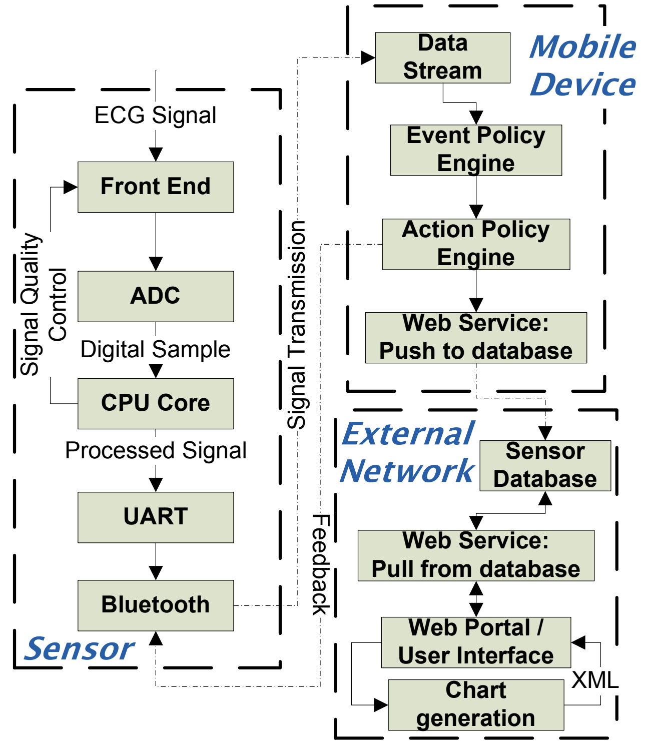


Figure 2. The sensor produces data that flows to the mobile device and external network while control commands flow back to the sensor.

indecipherable.

We approach this problem through several different tactics. First, including an additional reference electrode significantly attenuates power line noise. Second, the QRS detection algorithm uses cascaded low-pass and high-pass filters to preserve the frequency content of the ECG while reducing noise. Third, when an electrode is removed, the signal flatlines high or low (depending on which electrode) and when the policy engine encounters this situation, it can perform the action that the policy engine specifies.

4. SYSTEM DESIGN AND IMPLEMENTATION

MBS adopts a classical three-tier architecture in which the sensors, mobile devices, and the outlying network cooperate to adjust the flow of information throughout the system as shown in Figure 2. Tier one consists of a physiological sensor (initially an electrocardiograph), microcontroller, and radio (initially Bluetooth) with the form factor of a bandage. The sensor prototype collects and processes ECG data and transmits the processed information over the wireless channel either continuously or periodically. Tier two is the mobile device (e.g., PDA or laptop), which supports a number of programmable policies through the policy engine that can map events to actions. Tier three is a web portal that, together with associated web services and a database, allows authorized viewers to view the sensor information remotely either in real-time or post-collection. In this section we describe our approach and an overview of the system.

4.1 Sensor Design

The sensor prototype with discrete components has the form factor of a bandage and is attached to the user (patient) via three disposable electrodes that connect to the sensor via snaps. Two of the electrodes are used to make a differential measurement of the cardiac signal, and a third is used to hold the user at a fixed potential relative to the prototype ground. This provides a large input impedance and high common-mode rejection, which reduces many forms of interference. The electrodes can be placed anywhere on the chest, preferably in positions similar to the standard precordial leads.

A two-stage amplifier topology was chosen, the first stage consisting of an AD623 instrumentation amplifier with adjustable DC offset, and the second stage consisting of a single-ended amplifier with adjustable gain. A Texas Instruments MSP430 microcontroller with an integrated 12-bit analog-to-digital converter (ADC) is used to digitize the signal. Feedback from the MSP430 is used to adjust the DC offset and gain, so that the cardiac signal occupies the maximum dynamic range of the ADC. The prototype is powered from a 430-mAh polymer lithium-ion battery, which is regulated with an LTC4080 integrated charger and DC-DC buck converter.

4.2 MBS Client Service

We organize the MBS client service on the mobile device into three main operational blocks: the wireless radio, the policy engine, and the external network interface. We assume that the mobile device is normally powered up and able to receive data from the sensor, which is generally a fair assumption for mobile devices with small form factors. The MBS client service can operate in the foreground (with a user interface displaying pertinent information) or in the background. Figure 3 depicts the sensor and mobile device working together to show the ECG.

The wireless radio interface exports a stream of data such that the rest of the client service does not need to account for the details of the wireless radio used. We implement the client service to accommodate Bluetooth radios and the TCP/IP suite of protocols commonly associated with the IEEE 802.11 standard. Certain events may stem from the properties of the radio, including whether the radio is connected and whether the radio has received data within a specified amount of time. The policy engine incorporates these properties as events.

It is important to note that the policy engine's placement on the mobile device is a key aspect of the hierarchical framework. The mobile device has more computational resources than the sensor and is in proximity to the user so decisions can be incorporated quickly into the policy engine. The centralized location of the policy engine simplifies the architecture by being the mediator between the tiers.

The external network interface in the client service allows data to be exported as either an ECG signal or a heart rate signal. In order to prevent re-implementation of a heartbeat detection algorithm at the web portal and to leverage computation that has already occurred, the client service automatically piggybacks the heart rate value onto the ECG signal when unicasting data. The ECG signal is sent in 2076-byte blocks to minimize the number of transmissions over the wireless channel. The security of the wireless channels between the MBS client and the other tiers is addressed by

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

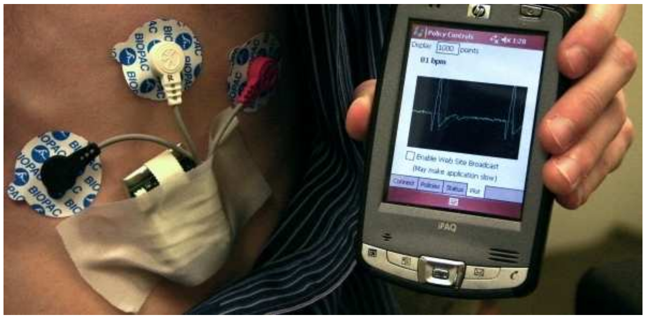


Figure 3. The sensor sends a wireless signal to the mobile device.

enabling encryption and authentication, preventing unauthenticated users from gaining access to the service. If wireless Internet access is unavailable, a logging option enables data to be stored on the mobile device for subsequent upload to the web server when wireless access returns.

4.3 External Network and Web Portal

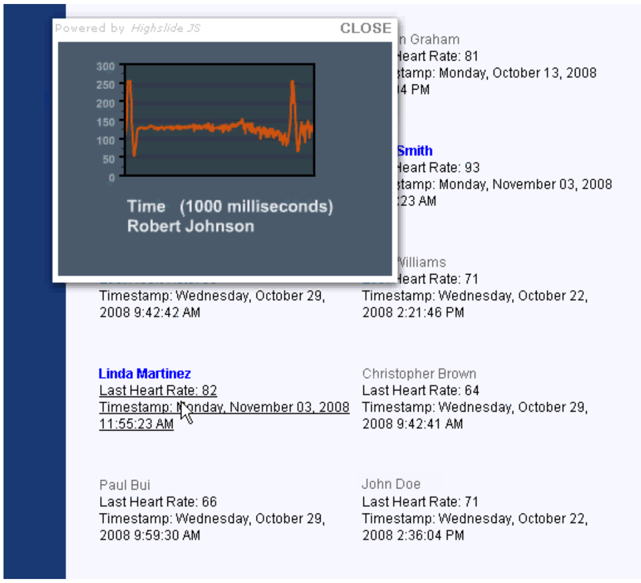
The communication between the MBS client and an external database is structured around a service-oriented architecture. We created two principal web services. The first pushes the data from the mobile device to a remote MySQL database and the second pulls the data from the database to the web portal. While we do not claim that our web portal is "secure," we take significant measures in adopting security best practices (e.g., requiring SSL, authenticating and authorizing web portal accounts, validating user input, encrypting the database connection string in a `web.config` file, using a limited-access database account to access and update the data, etc.).

The heart rate and ECG signals and any actions of the policy engine (via messages) are the primary sources of information for the web services. Heart rate can be determined regardless of the particular mode of operation that the mobile device has requested and therefore is available regardless of the mode of operation. The actions of the policy engine, while directly impacting the operation of the mobile device and sensor, also serve to inform a remote monitoring center by highlighting alarms or events of particular interest.

The web portal interface design, shown in Figure 4, is guided by the visual information seeking mantra, "overview first, zoom and filter, then details-on-demand" [17]. The heart rate of multiple users can be displayed at one time, providing the overview. A user's name can then be clicked to obtain more information. In order to dynamically view ECG data we leverage the Highslide JS library for point-and-click ease of viewing the ECG and the XML/SWF Charts library to produce dynamic chart content.

While viewing data in real-time for at least ten users is an important feature of the system, it also complicates the viewing of archival data for specific users. To reconcile this we developed another user interface that allows more fine-grained control of metadata specification. For example, the user, type of signal, connection time, and a signal window length may be provided for the web portal to render a chart.

The implementation of the web portal's sensor stream visualization was originally stateful in that the web server retained information for each connection and each sensor stream. As the number of connections and number of users scaled, however, the complexity of managing the information became overbearing. As a result we transitioned to stateless web services and made several optimizations. For example, since the chart is defined by an XML document we embed the specification for the chart in client-side code and return only the data portion of the chart from the server.



Time (1000 milliseconds)
Robert Johnson

n Graham
Heart Rate: 81
stamp: Monday, October 13, 2008
4 PM

Smith
Heart Rate: 93
stamp: Monday, November 03, 2008
:23 AM

Timestamp: Wednesday, October 29, 2008 9:42:42 AM

Williams
Heart Rate: 71
Timestamp: Wednesday, October 22, 2008 2:21:46 PM

Linda Martinez
Last Heart Rate: 82
Timestamp: Monday, November 03, 2008 11:55:23 AM

Christopher Brown
Last Heart Rate: 64
Timestamp: Wednesday, October 29, 2008 9:42:41 AM

Paul Bui
Last Heart Rate: 66
Timestamp: Wednesday, October 29, 2008 9:59:30 AM

John Doe
Last Heart Rate: 71
Timestamp: Wednesday, October 22, 2008 2:36:04 PM

Figure 4. The user interface permits an overview of heart rates and a view of the full ECG signal on demand.

5. EVALUATION

In this section we evaluate MBS in terms of power, performance, and accuracy of QRS detection. The three aforementioned metrics were chosen because the effectiveness of the policies follows largely from them.

The usability of MBS, and wearable computers in general, while critical to technology acceptance, is more difficult to quantify. The placement of the electrodes takes a few minutes and the connection setup takes approximately one minute. Together those steps represent the minimal amount of required user intervention.

5.1 Power and Performance

Low power consumption is most critical for the sensor's operation but is also relevant for the mobile device. With the 430-mAh battery on the sensor, our implementation can operate for up to twelve hours when heart rate information alone is sent. When a continuous waveform is sent, the sensor consumes 94 mW of power on average, with 87 mW used to power the Bluetooth radio. The transmit power of the radio has a direct effect upon the distance—that is, between the sensor and the mobile device—at which the policy engine will detect a disconnection. Class 1 Bluetooth devices, such as the RN-41 Bluetooth module that we adopt, have a range of approximately 100 m. As a result, the lifetime reduces to approximately two hours. This problem further motivates the need for a policy engine, and we recommend that the ECG waveform be sent for short, bounded periods of time and not indefinitely to maximize system lifetime. Future work to develop MBS as an extremely low power integrated circuit together with low-power radio options is expected to

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

extend its operational runtime.

The MBS client service has approximately 4,100 source lines of C# code and has a memory footprint of 103 KB. The mobile device used in our implementation is an HP iPAQ hx2490b with 64 MB of RAM. The sensor software is 435 source lines of C and 4,308 bytes once compiled. When both receiving heart rate data and unicasting via 802.11, the mobile device's lifetime is approximately four and a half hours with an average CPU utilization of 32.1%. Thus the mobile device represents a system-wide power bottleneck. These metrics may be improved by more closely managing the 802.11 radio such that information gets buffered for several seconds rather than sent whenever a heartbeat is detected or every 2076 bytes in the case of an ECG signal being sent. When running in waveform mode, the mobile device CPU is almost completely dedicated to the task. After profiling the application in waveform mode, we determined that drawing the waveform on the screen requires 42% of the time (1663 seconds of the 3920-seconds of running time). The utilization could be significantly lowered by reducing the amount of graphical user interface feedback.

5.2 QRS Detection

In order to detect QRS complexes, we analyzed all 48 recordings in the annotated MIT-BIH (Massachusetts Institute of Technology – Beth Israel Hospital) Arrhythmia database [5]. These half-hour ECG signal recordings contain cardiologist annotations that indicate when beats occur and their type (normal, premature ventricular contraction, etc.). While we recognize that the MBS front-end apparatus does not produce these signals, we submit that the database's ECG signals are noisier than those produced by our system, and thus represent a challenging data set, for two reasons. First, several of the signals from the database contain arrhythmias and other cardiac conditions that make QRS detection a more difficult problem. Second, the ECG signals produced by the MBS sensor are dynamically corrected for baseline wander before the QRS processing begins and the common mode signals in both electrodes are removed. Therefore we contend that the results of the QRS detection on the MIT-BIH signals are comparable or worse than if performed on signals obtained from a population tested with the MBS apparatus (which would not have the annotations associated with them to allow comparison with a known standard).

In order to prepare the MIT-BIH signals for input to the MBS client, they were upsampled from 360 samples per second to 1000 samples per second and shifted to unsigned 16-bit values. As the signals pass through the MBS client a log is produced that lists the R-R intervals in units of samples between QRS complexes. We executed the `bxb` program [5] with default parameters (including a 150-ms match window) to compare the cardiologist annotations with the R-R interval lists produced by the MBS client.

Our QRS detection algorithm operates on data sampled at 1000 samples per second. First each data point is filtered through a low-pass filter, high-pass filter, and first difference filter. Then the absolute value of the filtered data is included in a moving window integrator filter with a window length of 80 milliseconds. Once the data is filtered, peaks are determined and scrutinized with respect to an adaptive threshold based on recent QRS peak values and the values of noise peaks (including smaller deflections in the ECG, P-waves and T-waves).

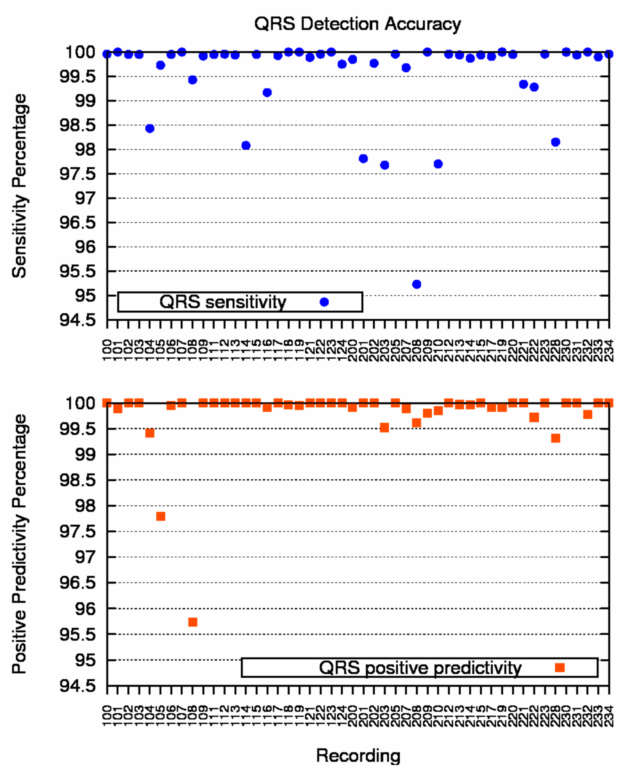


Figure 5. The average QRS sensitivity and QRS positive predictivity are over 99.5% for a total data set of over 109,000 beats. The signal recordings are not numbered consecutively in the MIT-BIH database.

Figure 5 illustrates the results of QRS detection for the 48 recordings. The sensitivity and positive predictivity were computed as given by

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Positive\ Predictivity = \frac{TP}{TP + FP}$$

where TP represents true positives, FN represents false negatives (missed beats), and FP represents false positives (beats that should not have been classified as beats). We achieve an average of 99.54% QRS sensitivity (the proportion of QRS complexes correctly identified as such) and 99.79% QRS positive predictivity (the probability that a QRS detection is correct) over all recordings. The average root-mean-square error of the RR intervals is 60.54 milliseconds, well below the tolerance window of 150 milliseconds. There are certain problematic recordings, including recording 108 and recording 208, that possess unusual ECG characteristics. For example, in recording 208 around the 23-minute mark, a great deal of noise is encountered in the first lead (but not in the second) preventing any discernible QRS complex when using the first lead alone. After the noise subsides the algorithm takes time to adapt the thresholds, causing several false negatives and, hence, a lower QRS sensitivity. In recording 108, the QRS complexes are particularly small relative to the P-waves and T-waves that occur at 0:20 and 28:30 into the recording. This results in incor-

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

Table 1. Varying Moving Average Window. More than 99.97% of the R-R intervals are between 273 ms and 2000 ms (220 bpm and 30 bpm, respectively) for up to 5 seconds when varying the number of beats over which to calculate the heart rate.

Window	Grace Period (seconds)			
Length	1	5	10	15
1 beat	0.9982535	0.9997869	0.9999649	1
2 beats	0.9998409	0.9999649	1	1
4 beats	0.9999265	0.9999883	1	1
8 beats	0.9999712	1	1	1

rect double-detection, resulting in several false positives and a lower positive predictivity.

5.3 Policy Robustness

For each recording, we took the results of our QRS detection algorithm at a sampling rate of 1000 Hz and calculated the instantaneous heart rates based on the estimated R-R intervals. We found that the instantaneous heart rates for the ECGs with arrhythmia display great variety (from less than 10 bpm to more than 305 bpm). These large variations are primarily due to arrhythmia, but noise can also produce the same effect. Based on this information, we seek to determine acceptable threshold levels for the policy engine to guard against being affected by arrhythmia or noise.

There are several ways to adapt the policy engine's functionality to be less sensitive to noise. The first is to use a low-pass (averaging) or median filter on the heart rate signal to reduce the effect of large swings and transient noise. The second solution is to adjust the threshold levels and the threshold times. A third solution is combining both techniques, which enables a large amount of flexibility in determining policies.

In order to gauge the dynamic operation of our policy engine, we examined the length of time that heart rates were reported for each of the recordings and derived a histogram of the number of seconds outside given thresholds. We found that on the MIT-BIH database with the cardiologist annotations, a lower threshold of 10 bpm and an upper threshold of 187 bpm was a sufficient window for all R-R intervals to be detected within one second. Note that although the distribution of heart rates goes above 200 bpm, because the R-R interval time and heart rates are inversely related, we observe that the higher the heart rate the less time it takes for the algorithm to "correct" itself (usually less than 0.5 seconds).

Table 2. Varying Sampling Rate. More than 99.97% of the R-R intervals are between 273 ms and 2000 ms (220 bpm and 30 bpm, respectively) for up to 5 seconds when varying the sampling rate of the ECG signal.

Sampling	Grace Period (seconds)			
Rate	1	5	10	15
100 Hz	0.9981245	0.9997557	0.9999650	1
250 Hz	0.9981967	0.999777	0.9999649	1
500 Hz	0.9982272	0.9997329	0.9999650	1
1000 Hz	0.9982535	0.9997869	0.9999649	1

Tables 1 and 2 show four different moving average window lengths and sampling rates, respectively, and the percentage of R-R intervals that fall within set thresholds of one, five, ten, and fifteen seconds. We set the lower threshold at 30 bpm and the upper threshold to 220 bpm to be indicative of rates beyond which a heart rate would be considered anomalous. As the threshold window becomes smaller, fewer R-R intervals fall within the thresholds in a given grace period. Raising the lower threshold to 40 bpm and lowering the upper threshold to 100 bpm still resulted in 98.02% of R-R intervals falling within the threshold window within a 1 second grace period. We note that sampling rate does not have a large effect upon the operation of the policy engine. Regardless of sampling rate, the number of intervals that fall within the threshold are about the same for each of the sampling rates. The length of the moving average window, however, has a more significant effect. While the vast majority of R-R intervals falls outside the thresholds for less than 1 second, the larger the window the more quickly the R-R intervals converge. This is because the larger window smooths out heart rate signals, tending to cancel out the effects of extreme values. We conclude that the policy engine mitigates the potential deleterious effects of rapidly changing ECGs on the operation of the system.

5.4 Threat Analysis

Fundamentally our system intends to bring convenience and service rather than invading privacy. We acknowledge that no system can be made perfectly secure. To justify our system design we now discuss (1) the security mechanisms we leverage in our implementation and (2) a threat model of the system. In our threat analysis, we proceed by first characterizing the system, then identifying assets and access points, and finally enumerating potential threats [12].

There are three main challenges in achieving the proper balance between usability and security within MBS. First, the entire system must have a reasonable lifetime. The sensor nodes are severely resource-constrained and energy use is at a premium. Second, the performance of the mobile device must not be dramatically affected by the presence of the MBS service. A process that dominates the CPU's cycles or drains the mobile device's battery is not tolerable. Third, the potential faults and transient errors must be well understood and mitigated such that the number of false event triggers is kept at an acceptably low level.

An adversary may have several objectives in attacking a sensor-based telemedicine system comparable to ours. First, obtaining ECG data could be a primary goal, either to replay it later to simulate a user's presence or to obtain a perspective on the wearer's state of being. Second, because the telemedical data is transported over the Internet, there is an incentive to eavesdrop or make copies of the data (e.g., a man-in-the-middle attack) or break into the data repository to view medical information through a brute force password attack, spoofing, or SQL injection. The benefit of a heavily networked architecture is that it permits a great deal of flexibility and intercommunication, but also calls for stiff measures to repel against an increased number of attack vectors. The web service and web portal interfaces provide an access point to a potential attacker. Furthermore, denial of service attacks through, for example, jamming the communication links or sending spurious requests to the sensor or mobile device to drain their batteries, represents a significant class of potential threats. In our system, denial of service will simply cause a *Disconnection* or *Timeout* event and sleep deprivation torture is difficult because once the Bluetooth

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835

radio is connected, it does not respond to service inquiries. We note that in an emergency scenario, a denial of service could be life threatening.

The foremost challenge for establishing the authenticity of the ECG waveform is establishing the origin of the signal. Since the validation process on the mobile device is not tightly coupled with the sensor, we must be able to have some amount of confidence that the signal is authentic. In other words, we must guard against an adversary replaying a previously recorded reading with an attack tool. The first aspect of an attack must be proximity—the attacker must be within range of the radio. The requirement of proximity can also limit more subtle attacks based on traffic analysis and surveillance. The second aspect of an attack is the strength of the bond between sensor and mobile device. MBS uses encryption and a PIN code to assist in protecting against replays through mutual authentication. In this case readings that are played from untrusted components are ignored.

6. SYSTEM EVOLUTION AND DISCUSSION

The prospects of MBS, and particularly the policy engine, go much further than just remote monitoring, as applications to healthcare and security will be more fully explored in several directions. With respect to hardware innovation, we seek ultra-low-power operation of the sensor by leveraging sub-threshold logic circuit design techniques. Eventually we intend to generate power through energy scavenging, which has been shown to be a promising approach to make sensor operation less dependent on other power sources [14].

While the ECG signal provides useful information for analyzing an individual's state of health, a richer view can be gained with multiple sensors, including sensors for the body and sensors for the environment. We envision a hardware platform in which sensors may be dynamically switched in and out with accompanying software that keeps everything in order. With multiple signals, sophisticated algorithms for processing should allow correlations to be made, with adverse events being validated by other signals before being reported to reduce errors. The presence of multiple signals would make the overall system more reliable and robust.

Although we do not currently implement a protocol for the web portal to issue directives directly to a particular sensor, the ramifications of such a protocol raise several issues related to security and federated trust. We leave these issues to future work.

7. CONCLUSIONS

We have presented MBS, a novel approach to managing the operation of remote medical monitoring. Our policy engine can accommodate an array of systems and applications with minimal changes. We have shown that our QRS detection algorithm has sufficient predictivity to be used as a security policy. The use of averaging windows (both for R-R intervals and within the QRS detection algorithm) is critical to ensure continuity in spite of local anomalous events and noise. While flexibility and configurability are important attributes of the system, intelligent defaults lead to a convenient off-the-shelf implementation. Ease of use will be the intangible factor that bridges technology to the patients that could make good use of it.

8. ACKNOWLEDGMENTS

The work described herein is sponsored in part by the National Science Foundation through the Wireless Internet Center for Advanced Technology (WICAT) and by an NDSEG Fellowship.

References

[1] U. Anliker, et al. AMON: a wearable multiparameter medical monitoring and alert system. *IEEE Trans. on Information Technology in Biomedicine*, 8(4):415–427, Dec. 2004.

[2] L. Biel, et al. ECG analysis: a new approach in human identification. *IEEE Trans. on Instrumentation and Measurement*, 50(3):808–812, June 2001.

[3] G. D. Clifford, F. Azuaje, and P. E. McSharry. *Advanced Methods and Tools for ECG Data Analysis*. Artech House, Norwood, MA, USA, 2006.

[4] R. K. Ganti, et al. Satire: a software architecture for smart attire. In *Int'l Conf. on Mobile Systems, Applications and Services*, pages 110–123, Uppsala, Sweden, June 2006.

[5] A. L. Goldberger, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, June 2000.

[6] B. Gyselinckx, et al. Human++: Emerging technology for body area networks. In *IFIP Int'l Conf. on Very Large Scale Integration & System-on-Chip*, pages 175–180, Nice, France, Oct. 2006.

[7] S. A. Israel, et al. ECG to identify individuals. *Pattern Recognition*, 38(1):133–142, May 2004.

[8] S. Kumar, et al. Using continuous biometric verification to protect interactive login sessions. In *Computer Security Applications Conference*, pages 441–450, Tucson, AZ, USA, Dec. 2005.

[9] K. Lorincz, et al. Sensor networks for emergency response: challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, Oct. 2004.

[10] D. Lucani, et al. A portable ECG monitoring device with Bluetooth and Holter capabilities for telemedicine applications. In *Int'l Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 5244–5247, New York City, NY, USA, Aug. 2006.

[11] T. Martin, E. Jovanov, and D. Raskovic. Issues in wearable computing for medical monitoring applications: a case study of a wearable ECG monitoring device. In *Int'l Symp. on Wearable Computers*, pages 43–49, Atlanta, GA, USA, Oct. 2000.

[12] S. Myagmar, A. J. Lee, and W. Yurcik. Threat modeling as a basis for security requirements. In *Symp. on Requirements Engineering for Information Security*, Paris, France, Aug. 2005.

[13] J. Pan and W. J. Tompkins. A real-time QRS detection algorithm. *IEEE Trans. on Biomedical Engineering*, BME-32(3):230–236, Mar. 1985.

[14] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, 4(1):18–27, Jan. 2005.

[15] C. Park, et al. An ultra-wearable, wireless, low power ECG monitoring system. In *IEEE Biomedical Circuits and Systems Conference*, London, United Kingdom, Nov. 2006.

[16] C. C. Poon, Y.-T. Zhang, and S.-D. Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine*, 44(4):73–81, Apr. 2006.

[17] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symp. on Visual Languages*, pages 336–343, Boulder, CO, USA, Sept. 1996.

[18] M. Steffen, A. Aleksandrowicz, and S. Leonhardt. Mobile noncontact monitoring of heart and lung activity. *IEEE Trans. on Biomedical Circuits and Systems*, 1(4):250–257, Dec. 2007.

[19] G. Wübbeler, et al. Verification of humans using the electrocardiogram. *Pattern Recognition Letters*, 28(10):1172–1175, July 2007.

Digital Object Identifier: 10.4108/ICST.BODYNETS2009.5835
http://dx.doi.org/10.4108/ICST.BODYNETS2009.5835