INTERACTING WITH SOCIAL NETWORKS TO IMPROVE HEALTHCARE BODY SENSOR NETWORKS

by

DAVID BAUSCHLICHER AND STEVEN BAUSCHLICHER

Advisor
HALA ELAARAG

A senior research proposal submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science
in the Department of Mathematics and Computer Science
in the College of Arts and Science
at Stetson University
DeLand, Florida

Fall Term
2010

ACKNOWLEDGEMENTS

Funding for this project was provided by a grant from the Dean's Senior Research Fund and Stetson University's Math and Computer Science department.

LIST OF FIGURES

ABSTRACT

Over the last decade, the demand for efficient healthcare monitoring has increased and forced the health and wellness industry to embrace modern technological advances. Body Sensor Networks, or BSNs, can remotely collect patient data and upload vital statistics to servers over the Internet. Advances in wireless technologies such as cellular devices and Bluetooth increase the mobility patients experience while wearing a body sensor network. When connected by the proper framework, BSNs can efficiently monitor and record data while minimizing the energy expenditure of nodes in the BSN. Social networking sites play a large role in the aggregation and sharing of data between many users. Connecting a BSN to a social network creates the unique ability to share health related data with other users through social interaction. In this paper, we propose to integrate social networks and BSNs to establish a community promoting well being and great social awareness.

1. INTRODUCTION

The health care industry has been rapidly expanding over the past few years. A particular area of research experiencing rapid discovery is the use of body sensor networks, or BSNs, to monitor patients. A BSN consists of sensors recording biological data which is then sent to the corresponding data coordinator. From there the data can be interpreted in various ways. At first, hospitals used BSNs to monitor patients on site, but soon technological advances allowed the patients to move into more native environments, such as their own homes, to be monitored. A wired connection between the sensors and data coordinator was originally used to facilitate BSN communication. With the advent of wireless protocols however, technologies such as Bluetooth and ZigBee (802.15.4) have eliminated the wires from the network and increased the mobility of the patient. Users can now wear body sensors and perform everyday tasks including exercising without the need to adjust their body sensors.

Mobile devices such as mobile phones and PDAs have seen significant development in the recent years as well. Modern day smart phones run complex operating systems such as Android and are more comparable to laptops than previous cell phones. Because of their inherit portability, these smart phones make an excellent candidate for the data controller portion of a BSN. These phones also come equipped with multiple sensors such as GPS, accelerometer, and light sensors, which further promote data aggregation and can reduce the need to buy additional sensors. Google's Android operating system is an excellent candidate for a BSN controller. The open source Android OS runs Java source code, and because of its portability and Bluetooth API, a single native controller can be made for the OS which could then be propagated to other devices in the future.

The final processing layer of BSN data can be seen as the most crucial part of the entire process. The raw data by itself is worthless unless it can be used to solve a problem or enhance the well being of the patient. The data can be used to track health patterns in patients, display vital health statistics, or even create recommendations for exercise. This paper explores submitting this sensor data into the realm of social networking. Social networking web sites such as Twitter and Facebook have gained massive popularity among users worldwide. User's send updates about their events, location, status, etc, only to share it with other users around the globe. If applied to the health industry, information such as health condition, weight, exercise levels, and heart rate, could be shared between users to create a positive and educational environment about wellness. This paper focuses on general healthcare BSN issues as well as distributing the BSN data through social media channels such as Facebook and Twitter to increase and challenge physical well being. The rest of the paper is organized as follows: section 2 provides background information about wireless and body sensor networks. Section 3 presents related work in the literature, while section 4 focuses on the SPINE framework. Section 5 describes the system design and section 6 shows the HatterHealthConnect system. Finally section 7 provides a conclusion of the paper.

2. Background

2.1 Wireless Sensor Networks

Mobility and freedom have always been a goal of network developers, and with the creation of wireless protocols and devices that goal is attainable. Wireless sensor networks (WSNs) are generally known for their limited resources such as energy, storage, and processing. However, their mobility allows for ad hoc deployment; creating new networks in previously unreachable areas. Other main differences from traditional networks include the data-driven design of WSNs and unique traffic patterns. WSNs are focused on data and its timely delivery, not just the connection of two end points in a network. Because WSN nodes are close together, often data is duplicated and not delivered in a timely manner. Throughput is not necessarily an accurate measurement of the network. The many-to-one traffic pattern also creates a need for protocols to be adjusted. With a single receiver, it is difficult to efficiently record all the senders' data [11]. WSNs are used in applications such as location awareness, ad-hoc networking, and patient monitoring. They set a standard for body sensor networks and can be used to test future protocols and techniques for BSNs as well.

Congestion is a common problem in WSNs because of the way the network must be set up. Wireless networks require the nodes to be listening at all times. Some processing is required for nodes to filter the correct messages from the traffic, and the larger the network, the larger the problem. To minimize congestion, standard routing protocols need to be altered to compensate for wireless issues. Protocols that focus on improving hop-by-hop reliability are more appropriate in a WSN because each node may be crucial for network sustainability [20]. Certain congestion preventing protocols such as congestion control and fairness (CCF) focus on each hop, and adjusts transmission rate depending on child nodes [22]. End-to-end protocols like TCP do not take into account individual nodes and are not as suitable for WSNs. When used for health-care, WSN congestion must remain minimal in case of emergencies. Alerts for patients experiencing cardiac arrest or other critical health conditions cannot afford to be lost in congestion [23].

2.2 Body Sensor Networks

A body sensor network, or BSN, can almost be seen as a subset or derivative of a WSN. Many of the same components and issues of WSNs are present in BSNs, and the similar techniques can be used to solve the problems. However, there are some distinctions between the two. BSNs' nodes consist of sensors able to read biological data, and these results often are forwarded to a lab or hospital to perform some medical evaluation. Sensor nodes can also consist of implantable devices. These implanted devices require a slightly different configuration because they only have one battery charge. Fixing them while implanted is currently not an option [1]. BSNs are often involved in critical systems as well; a patient's life or death may rely on the speed of the message delivery. Therefore, message delivery rate and congestion are important issues a BSN must focus on [2].

Most modern BSNs contain a variety of medical sensors along with a controller or base station. Many medical sensors such as a pulsioximeter, ECG, and heart rate monitor can be used to report data to the controller [7]. In order to work with an array of devices and reduce energy costs, most communicate through common protocols such as Bluetooth or ZigBee. The controller node must support the protocol as well, and often acts as an internet gateway too. Modern day cell phones are a natural fit for the controller node because they meet the protocol requirements and are portable [7]. Cell phones have additional connectivity tools such as SMS and 3G to increase network resources. Some BSNs rely solely on these technologies to communicate, and have even developed entire protocols around them [21]. The patient also has prior experience with his/her mobile device, so it is often an easy transition. Previous experiments have used similar technologies such as PDAs for the controller node [8].

Body sensor networks can greatly vary in size and purpose. A small BSN monitoring an athlete's vitals while exercising may only require a single patient. On the other hand, a large healthcare facility may have hundreds of patients, each with his/her sensors reporting to a large central database or program [10]. Depending on the case, a different network structure may be needed to maximize the efficiency of the BSN. However, the same basic BSN issues need to be addressed regardless of purpose or size. Energy efficiency is crucial for BSNs taking advantage of wireless sensors and devices, and can be improved by enhancing the network layers. The second issue BSNs face is security. Health data is extremely personal, and it is important for patients to know their data is secure and only accessible to the appropriate parties.

2.2.1 Energy Efficiency

With wireless sensors leading the forefront for body sensor networks, energy efficiency becomes a serious issue. All functions performed by the sensors, such as reading and transmitting the data, require energy. Because the sensors cannot yet be charged wirelessly from the data controller, they must power themselves via battery. This is very inconvenient because constant recording of data is necessary to obtain good results. Removing the sensors for charging could result in loss of vital data, or when monitoring critical patients, prevent the sending of an emergency alert. The problem becomes even more noticeable with sensors that can be swallowed. In that case, the sensor is only able to collect data for one charge and conserving energy is extremely important.

When examining the energy expenditure of sensors, it is clear the networking layers play an important role. While extracting data from the body uses a fair amount of energy, it is often already minimized in the sensor hardware. The network layers are much more software oriented and can be handled separately. This does not mean, however, no hardware adjustments should be made in sensors to reduce energy expenditure. IEEE 802.15.4 MAC does not meet the scalable BSN design the healthcare industry requires [1]. Because the local BSN does not need to reach other computers around the world, a simpler design reaching neighboring nodes is all that is needed. Distributed Queuing Medium Access Control (DQ MAC) is one proposed solution which meets the scalability of BSNs while meeting healthcare's power consumption requirements [1]. Another hardware solution, Distributed Queuing Body Area Network (DQBAN), seeks to enhance the MAC protocol by increasing hardware layer's reliability for BSNs [24].

The most important aspects of BSN networking are the message sending rate and successful delivery rate. The fewer transmissions required by the nodes, the lower the energy expenditure. Every message that does not successfully reach the host may have to be retransmitted, causing a greater loss of energy. Therefore, minimizing the transmissions is crucial to saving energy. These issues can be addressed at both the transport and application layers, and a combination of the two leads to the best results. When working with only a few nodes, congestion is not really an issue. Instead, the controller should attempt to request as little information as possible from the sensors. Several techniques have been developed to accomplish this, and the one this paper focuses on is a hop-by-hop backpressure mechanism [2]. This technique uses additive increase multiplicative decrease (AIMD) to adjust the rate at which sensors report data. When sensor data reaches a critical point, the rate of data sending increases and only slows down once the sensor reaches a suitable value. Learning Automata-Based Congestion Avoidance Scheme (LACAS) is one such implementation using hop-by-hop backpressure mechanism, although its approach is more focused on congestion awareness [2]. Congestion awareness can be handled similarly to message rate. When too many nodes send data to the controller, the controller becomes flooded and must drop packets. Those packets contain useful information and must be sent again using more energy and network bandwidth. The hop-by-hop backpressure mechanism can be used here as well, reducing the packet send rate of less critical sensors at any point in time. Because most BSNs deal with this many-to-one pattern, resource allocation for the sink node, or controller, is vital.

Another previous attempt to reduce energy consumption in BSNs was Energy Efficient LEACH, or EEL. LEACH stands for Low-Energy Adaptive Clustering Hierarchy, and is a protocol for reducing the number of messages sent to and from the controller node. As seen by the figure below, the sensors are grouped into corresponding clusters either by location or other routing protocols. A head node is then selected which communicates with the controller or base station directly [3].

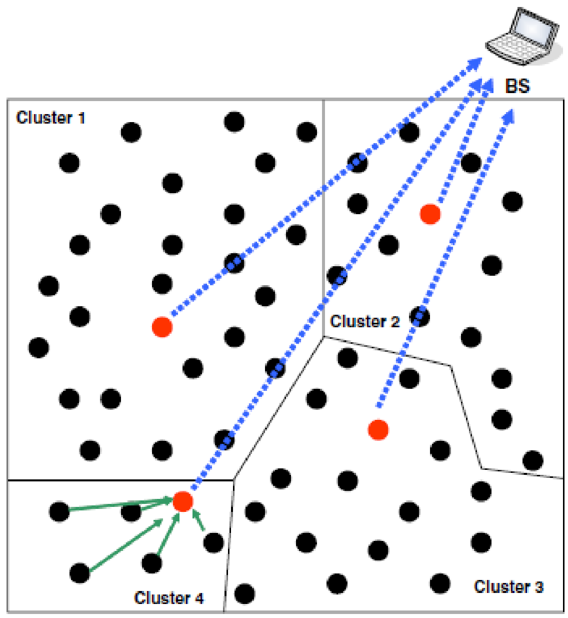

Figure 1: LEACH divides nodes into clusters [3]

EEL makes slight adjustments to LEACH to better suit wireless BSN energy consumption. First, the chosen cluster head is the one with the most energy remaining. Second, nodes are turned off if the sensing environment does not meet certain criteria. And finally, wirelessVegas is used for the network transport layer. Based on TCP Vegas, wirelessVegas uses high-speed data transfer to reduce transmission time and reduce the cost of transmission energy [3]. Used in scenarios where a large number of sensors connect to a single controller, the EEL protocol provides an attractive way to minimize sensor energy consumption.

2.2.2 Security

Security is an essential component of a BSN, because the physiological information gathered from the sensors is very private data. There are many separate security issues that arise between how the data is communicated and who is able to access certain data. The security requirements for a BSN should be data confidentiality, data integrity, and data authenticity. Data confidentiality means that only the correct user or node should have access to the data and that the data does not contain any personally identifiable material other than the health information. Data integrity is necessary to guarantee that the data received has not been modified or tampered with. Finally, data authenticity is required so that data gathered from a BSN is linked to the proper patient [15]. Consequently, if too many security features or inappropriate security features are implemented in a BSN, they may have the possibility of hindering it. This compromise that always exists between security and efficiency is extremely important in BSNs, because the BSNs must operate with minimal battery use and extra security features require more calculations and extra bandwidth. Also, emergency alerts would need to be sent out quickly and the security features should not slow down the messages to a large degree.

In case of an emergency, data must be accessible to the correct personnel when needed. For example, if a patient is found to have a heart attack, the information collected from the BSN must be available to whoever takes care of him when he arrives at the hospital. Because the patient may not be able to predict who will be viewing the information in advance and may also not be in condition to grant permission to the appropriate medical caretakers, the BSN should be able to grant access to some personal but not be freely available. Therefore, a BSN should be able to prevent unauthorized access, but also be able to grant permission when necessary to new users. One suggestion is to use biological information such as ECG data to generate a key to encrypt data that is stored on a centralized server. In order to access this identity-based encrypted data, a doctor must provide that same key which will only access a certain time slice of the data [12].

Communication between nodes must also be private and inaccessible by any device except for the central or gateway node. Several papers discuss the use of physiological data to generate keys for the security between the nodes. The keys generated should be time-variant, universally measurable, random, and distinct for each user. In [13], signals collected from an EKG are used to generate keys and encrypt the data passed between nodes. The authors in [14] also discuss the use of the physiological data itself for security in a BSN and implements a clustering scheme to prevent a malicious entity from spoofing nodes into believing it is the base station. Both papers mention that the security within the BSN should not be compromised if an unauthorized user gains physical access to a node or if two BSNs are in close proximity.

3. Related Work

In this emerging field of research, other similar projects have been created. Harvard's CodeBlue [16] is a project created to adapt wireless sensor networks for use in emergency medical situations. In this project, sensor networks are used to send real-time vital signs from a group of patients to emergency medical technicians. These sensor networks would allow for a rapid medical response to a mass casualty event and would allow health personal to locate those patients in most need. CodeBlue makes use of the Berkeley MICA mote which contains a microcontroller, local storage area, and a low-power radio. These motes run on the TinyOS operating system and have a battery life of approximately 5-6 days while running. The communication is handled over the IEEE 802.15.4 standard to also conserve power. A pulse oximeter that attaches to the mote has been developed to deliver heart rate and blood-oxygen level, and an ECG was in development during the publishing of the paper. CodeBlue is highly scalable and works in an ad hoc setting, a critical need for use in emergency situations.

A WSN has been proposed by members of the Computer Science Department at the University of Virginia [17] to provide support to the increasing elderly population. This WSN combines wearable sensors along with environmental sensors to provide medical monitoring and memory enhancement to the patient. The architecture is divided into five main components: a body network, an emplaced sensor network, a backbone, back-end databases, and human interfaces. The body network is composed only of unobtrusive wireless sensors, but the environmental sensors have the option of being completely wireless and requiring batteries for power or of being plugged into an outlet. Data gathered from the body and emplaced sensor network is sent across the backbone of the WSN to be stored in the back-end databases or displayed in one of the human interfaces. The sensors used in this implementation include a motion sensor, temperature sensor, breathing rate sensor, pulse-oximeter, and an EKG. These sensors communicate with the backbone using the Zigbee (802.15.4) wireless protocol. Real-time data can be viewed by a PDA that can connect to the backbone or by LCDs located on the motes.

Another wireless sensor network, discussed in [18], is designed to safely and continuously send physiological data from a patient to a local WLAN to be further transmitted. The patient wears an ECG sensor which wirelessly transmits data to one of the local relay nodes that are strategically placed throughout the entire building the patient is living in. This overcomes the deadspot problem that could occur if using a WLAN through walls. In order to avoid excess power consumption, the ECG sensor periodically takes samples, stores them in a buffer, and then goes back to sleep. Only once the specified buffer is full does it place the frame around the information and send it to the local node. The nodes have two different modes, self organizing mode and data transmission mode. When a node initially starts up, it stays in the self organizing mode until the surrounding nodes successfully add the node into their routing table and vice versa. Once the node is added to the system, it waits for data and then acknowledges the received packets and forwards the data towards the uplink node. An SMS message would be sent in the case of an emergency found in the ECG data.

DexterNet [4], a heterogeneous body sensor network, is an open-source project that makes use of the open-source library called Signal Processing In Node Environment (SPINE). DexterNet has a three layer architecture that includes a body sensor layer (BSL), a personal network layer (PNL), and global network layer (GNL). The BSL contains two different types of custom sensors. The first is a motion sensor that contains an accelerometer and a gyroscope. The second type is a biological sensor that acts as an electrical impedance pneumography (EIP), and ECG, and an accelerometer. The PNL consists of a Nokia N800 tablet that communicates and collects data from the different sensors within the BSL. This communication is accomplished over IEEE 802.15.4 and makes use of the SPINE API on the node and base station side. Finally, the Nokia N800 forms a GNL by sending data across the Internet through a Bluetooth, Wifi, or other broadband connection. These Internet servers then use the data collected for higher level applications. Some server-side applications already created provide features such as displaying a graphical animation of a user's current position, creating a database of movement information to improve human movement recognition algorithms, and monitoring pollution on a patient's walk through the city [4].

Researchers at the University of California, Berkeley have begun working on an API for the Android OS that provides specific functionality for BSN development. The project, titled WAVE [5], has several core components including sensor interaction, database interaction, and data processing functions. The sensor interaction is handled by the SPINE framework, allowing WAVE to focus less on the low level communication between sensor nodes and the mobile platform. Because the Android OS has a java based development kit, SPINE should work well with it as long as the sensors can also make use of it. Also, because the Android OS is becoming increasingly popular on multiple models of phones, the SPINE framework can be ported over once and used for multiple applications on multiple devices. After data is collected from the nodes in WAVE, database interaction is handled using REST architecture so a user can easily insert and store information from the mobile platform into a remote database. This REST architecture basically refers to a stateless, client-server model similar to how the World Wide Web works. Finally, the data processing functions allow the user to access algorithms that are frequently used within BSN applications. These functions include action recognition, energy expenditure calculating, and GPS location tracking.

Using the WAVE API, a few applications have already been created to monitor certain health aspects. One such application is CalFit [5]. CalFit uses the native sensors on specific Android phone models to calculate energy expenditures based on the SPINE Kcal algorithm. It then logs the user's data into a database where all the users of this application can compare their caloric expenditures. The database also ranks users and allows for the creation of teams to sponsor encouragement and competition.

Work has also been done to use artificial neural networks, Bayesian networks, and Hidden Markov Models to develop context aware sensing in BSNs. People are very sensitive to external context changes such as a change in the person's activity or temperature of the environment and these situations need to be analyzed appropriately to draw the correct conclusions about a person's health status. Difficulties in accomplishing this include noise introduced by the sensors, the need for context sensing to detect transitions in context as opposed to a single snapshot in time, and the problem that as the number of inputs (sensors) increases, the learning rate slows down. Advanced computing techniques such as neural networks and Bayesian networks would be advantageous because each of the individual sensors could learn without supervision and do not require prior knowledge of the context [19].

4. SPINE

The Signal Processing In Node Environment (SPINE) [6] is an open source software framework that was created to assist in the implementation of Body Sensor Networks. The SPINE framework provides APIs that abstract much of the low level functionality required when working with sensing nodes that read in raw data and that must communicate that data. This allows for rapid prototyping and the ability to locate and address problematic issues quickly. The SPINE framework has two components: one is located on individual sensor nodes and the other on the coordinator node, or central node, that collects data from the individual sensor nodes. This distributed architecture allows for computations to be done on both the node and coordinator side, saving power and network usage.

The sensor node component is developed in the TinyOS environment and addresses the issues of communication, sensing, and processing. The communication portion deals specifically with accessing the radio and organizing the packets that will be sent to and from the coordinating node. The radio can be operated with a CDMA-CS schema or a TDMA schema if a large amount of data from different nodes needs to be sent during a similar time frame. The package manager on the node side handles the separation of the packets into smaller pieces that can be reassembled on the server-side. The sensing logic block controls the data sampling from the sensor and storage on the local node. This block is where the sensors are registered to SPINE and where the parameters for each sensor are set. The data collected from each sensor is stored in a pool of first-in first-out buffers to be processed later. The final block on the node side, the processing block, takes care of all the necessary computations to be handled on the node before sending the data to the coordinator. These functions all must implement a generic interface so the coordinating node will not need to know the implementation details. The functions can be implemented as a Feature Engine or as an Alarm Engine. A Feature Engine function is able to run a simple periodic calculation on raw data before sending the data to the coordinating node. These features include calculations such as a maximum, mean, or standard deviation of a select group of data. The SPINE Alarm Engine is used when data only needs to be sent to the coordinator if it meets specific conditions. This engine provides event-driven messages by sending data when it does not respect a specific threshold. These four different thresholds are customizable and can trigger an alarm if data is above a value, below a value, in-between a range of values, or outside a certain range of values. Each of the nodes' processing functions are set up and started from the server-side.

The sever side component of the SPINE framework is developed in Java SE and manages the organization and operating of the BSN. Because this component is a lightweight Java API, any platform that can run Java code is able to act as the coordinator node. Also, the framework logic is not dependent on the network it is communicating over and can run independently of the underlying protocol stack. Another strength of this framework is that the majority of development and programming is done on the server side. Indeed, the establishing, managing, and data collection from nodes is all handled from the Java code written on the coordinator node. In order to use the server side of SPINE, a Java application simply needs to implement the appropriate SPINE interface and then it can access the entire API [6].

5. SYSTEM DESIGN

To further the functionality and diversity of body sensor networks, we propose HatterHealthConnect. HatterHealthConnect is a health monitoring system that gathers physiological information to be integrated with social networks to promote healthy living and further peer connectivity. Live data such as heart rate, calories burned, and workout duration will all be calculated and will be sharable through these networking portals. HatterHealthConnect is designed to be easy to use, highly portable, and unobtrusive. Figure 2 depicts the overall design of HatterHealthConnect.

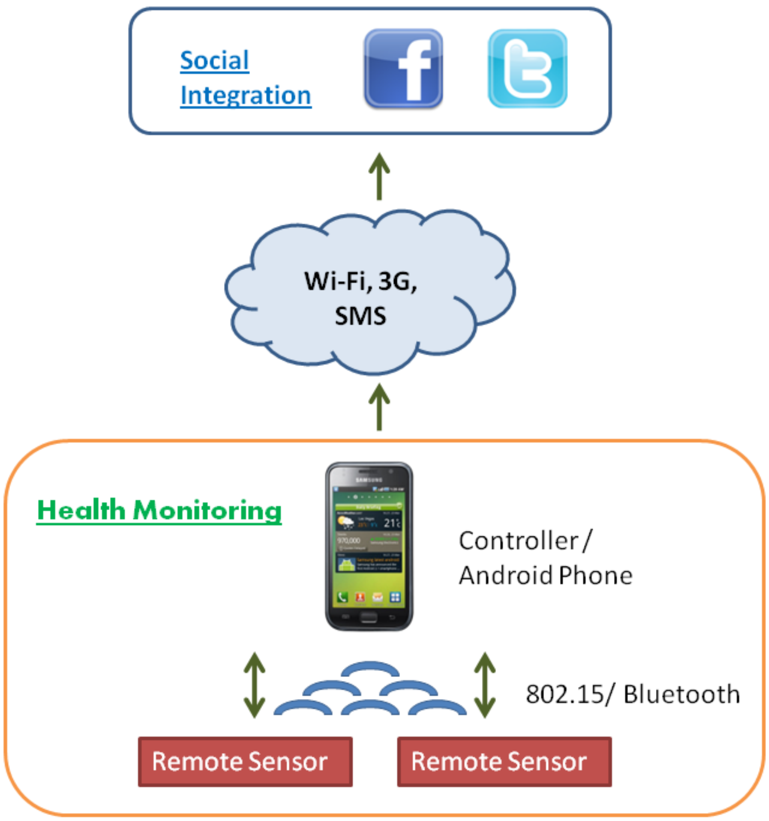

Figure 2: HatterHealthConnect Design

The monitoring and gathering of physiological data will be handled by a wireless body sensor network comprised of multiple nodes. The sensor nodes will include an Intel Shimmer Wireless Sensor Unit and attached ECG sensor and a HxM Bluetooth Heart Monitor. The Intel Shimmer is a lightweight unit that contains a 3-axis accelerometer, tilt sensor, Bluetooth and 802.15.4 radio, and connectors for expansion equipment. It is run by a MSP430 processor that allows for low power consumption and longer battery use. The software on the wireless Shimmer node is run on the open-source TinyOS environment and is therefore completely configurable. Attached to the Shimmer node will be a 3 Lead micro-power ECG expansion module that can record electrical impulses within the heart. The second sensing node used will be a Zephyr HxM Bluetooth Heart Monitor, which uses a conductive fabric to gather heart rate data. The monitor also contains an internal accelerometer to calculate speed and distance using an impact detection algorithm. It is attachable to a user by a chest strap and can communicate wirelessly via Bluetooth. Finally, the coordinating or central node will be a Samsung Captivate mobile device. This phone runs Android OS 2.2 on a 1 GHZ processor and includes a GPS and accelerometer.

Figure 3(a): The Intel Shimmer internals, Figure 3(b): the HxM heart monitor

The BSN will use Bluetooth as its form of intra-node communication so the user can wear or carry the appropriate nodes wherever he goes. The Samsung Captivate will periodically collect data from the sensor nodes and store that information locally. Because the Intel Shimmer sensor is run on TinyOS, communication and control of the sensor will be accomplished using the SPINE framework. The HxM Bluetooth Monitor comes with a developer's kit that simplifies the sharing of heart data. Data collected on the phone will be analyzed and processed further if necessary and the results can then be uploaded to the social networking sites Facebook and Twitter over Wifi, 3G, or SMS as shown in Figure 4.

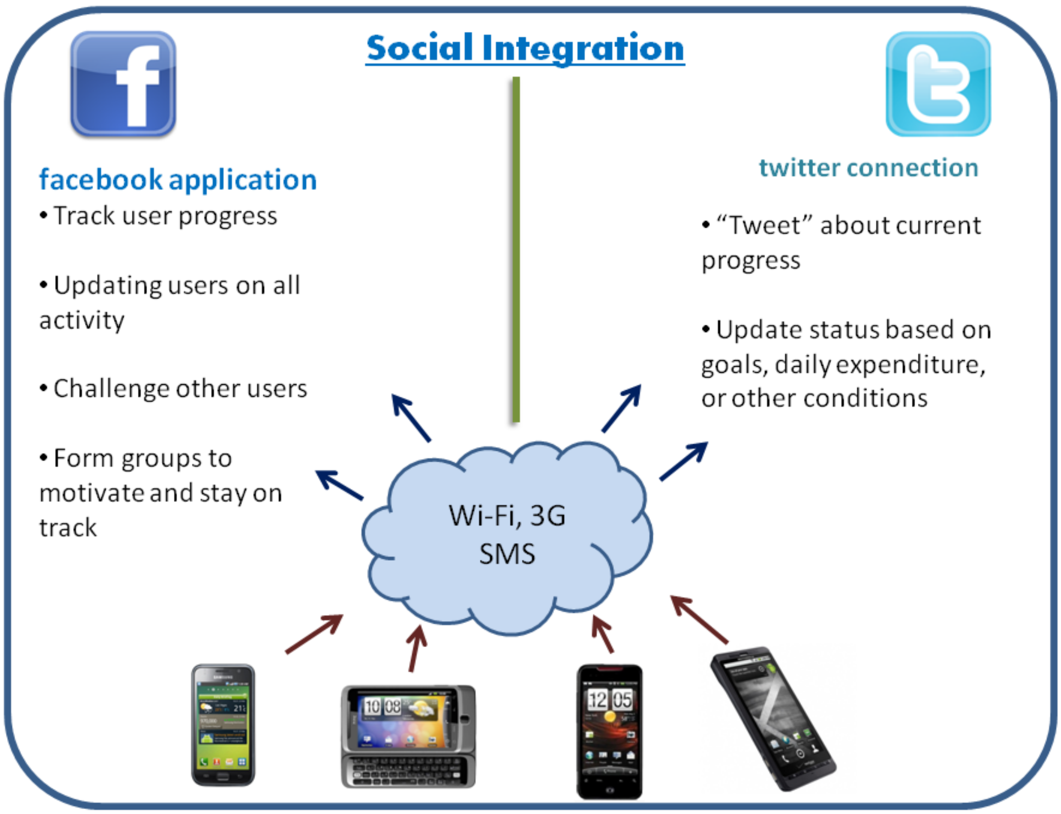

Figure 4: Methods of Social Interaction

Through Facebook, users will be able to interact with other people using HatterHealthConnect to share results, give advice, and compete against other users. This social interaction aims at promoting a healthier lifestyle by using encouragement and competition. Users will be able to reach a much larger resource of advice and support as they share their own data and also comment on others' data. Also, by frequently uploading data, a user will be able to track his progress compared to his peers' and will be positively pressured into continuing his own exercise activities or improving on them. Similarly, users will be able to share their results via Twitter, so those who follow someone's posts who is using HatterHealthConnect may be inspired to live a healthier lifestyle.

6. HatterHealthConnect

The controller node of the BSN will be a Samsung Captivate Galaxy S Android phone, and the two body sensors used to monitor users will be the INTEL Shimmer Mote and ECG Sensor along with the HxM Bluetooth Heart Monitor.

6.1 Android Development

As mentioned previously, a Samsung Captivate will be the base/controller node of the BSN. Because the phone is running Android, Java will be the primary language used for development. Freely available from Google is the Android SDK and Android Development Tools (ADT) for the Eclipse IDE. The Android SDK comes with a variety of tools for developing and publishing applications onto android devices. The emulator as seen in Figure 5 can be used for development without a device, and it is often easier to test the applications on the emulator before publishing it on the device.

Figure 5: Android emulator used for application development and testing

Eclipse is a fairly standard suite for developing Java applications and connects easily with the Android SDK using the ADT. From Eclipse, new programs can be created and published directly onto emulators and devices. Settings for the Android projects can be configured through the GUI without using the command-line, and files can be retrieved or pushed to the device as well. Seen in Figure 6, a snapshot of Eclipse using the ADT and emulator.

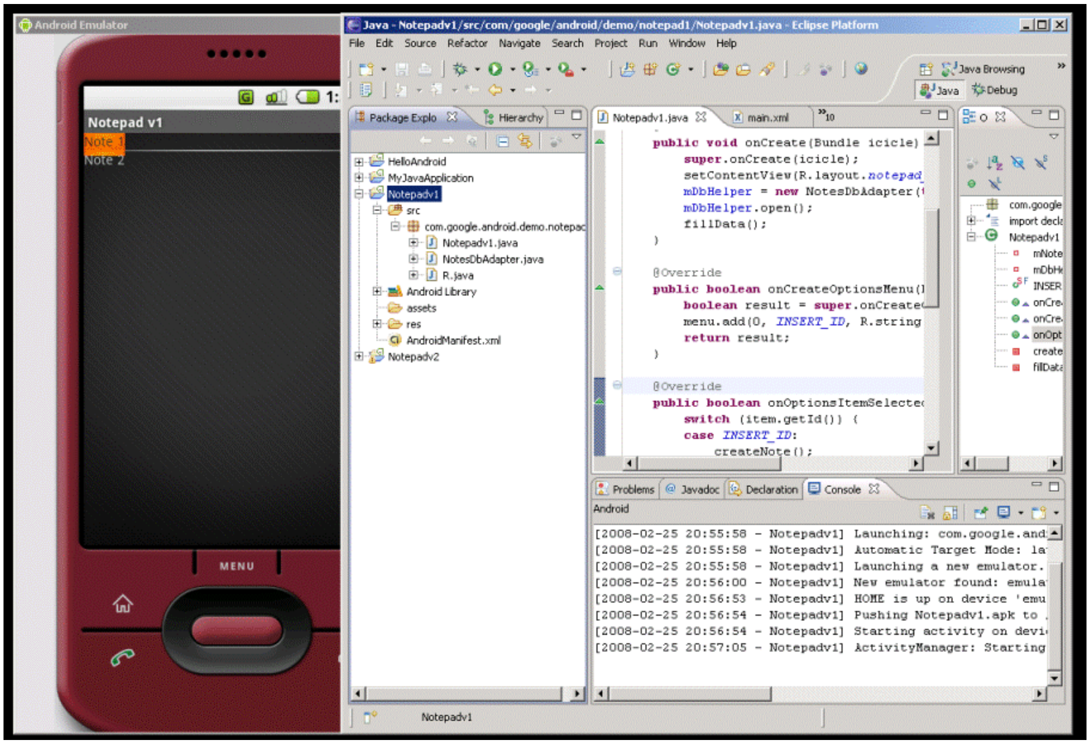

Figure 6: Eclipse IDE with ADT Plug-in

Java development for Android is essentially the same as Java on other platforms. The main differences for Android development are the addition of new sensor functions along with handling the screen's display. Luckily, like most other modern GUI development schemes, Android can create designs on the screen from Extensible Markup Language (XML). The view for the application can be designed in XML then accessed through the program to edit what the user sees.

The Java SDK for Android gives the application privileges to access sensor data. Along with the remote sensors attached, the application will be able to capture data such as GPS coordinates, accelerometer values, and screen orientation. Android 2.0 introduced the new Bluetooth 2.1 functionality, allowing an application to connect with other devices and push objects back and forth. Hence, bluetooth will be the primary technique used to connect and collect data from remote sensors. The SDK also allows the application to use the networking tools available on the phone such as Wi-Fi, 3G, and SMS. These can be used to remotely update databases or send messages to other devices. Because Android is open-source, it is an excellent platform to develop the controller node on. The server portion of the SPINE API is also written in Java, so it should be feasible to port it over to Android. If ported over to Android, most of the sensor connectivity and energy efficiency techniques SPINE offers will be used to maximize the application effectiveness.

The Android application will provide an interactive interface in addition to being the controller for the BSN. Users will be able to monitor their own progress and view past data. While the data is being uploaded for social networks, the user will also have the opportunity to post an update about the results in his online statuses. Alterations can be made to monitoring habits, and the user can update his view depending on which data he would rather see. The familiar Android interface should make the user more comfortable and reduce the learning curve of the program.

6.2 Facebook and Twitter Development

In order to increase visibility, social networking sites often provide developer-friendly APIs for accessing and updating user data. Facebook and Twitter are no exception, and they support APIs for a variety of platforms and languages. These APIs often include multiple ways of accessing the site data and have security measures in place to restrict whose data is visible. Additionally, many open-source projects have been initiated to wrap these APIs and increase the ease of site interaction.

Facebook development is more complex than Twitter because of all the different functionality Facebook provides. User's status updates are only a small portion of the site. Users can upload photos, videos, and notes as well as play games. They also have the option to sign up for and use applications by third party developers. These applications can access third party information and can use premade tools by Facebook for enhancement.

HatterHealthConnect will include a Facebook application and it will be the primary method of exposing content to users. Data will be uploaded from a phone to the database backend which is responsible for conglomerating all users' data. The Facebook application will then pull data from the database and display it to the users. Users will be able to see their current progress as well as others connected with them. Modifications to goals and future plans will also be editable within the application, and users without a BSN will be able to edit their information directly. Because updating a status should be at the discretion of the user, that option will be controlled directly from the phone.

Facebook has released an open source API for Android, allowing a user to interact directly with his/her Facebook account. When the API libraries are added to a project, an activity can be created to login and request account information. The application, if prompted, can then update a user's status based on the information received from the healthcare monitoring portion. Interacting directly through the Facebook Android API for status updates should increase awareness, because it will require little work to update one's status and many users should use it. It will be much easier for users to activate the updates on their phones than to go through the Facebook web application and update it there.

Twitter integration will be similar to the status update portion of the Facebook integration. Twitter is a microblogging site that lets users “tweet” updates about themselves, and remotely interacting with the site is fairly simple. All Twitter API calls are done through HTTP. Updates for a user can be sent via POST and the API accepts multiple formats such as XML, JSON, RSS, and ATOM. Open source wrapper APIs have also been developed for many platforms including Android, which will simplify the development process. A HatterHealthConnect user will have the option to tweet about his/her progress or current status, very similar to how one would update his/her Facebook status. Along with Facebook, Twitter should provide another channel for promoting health and fitness by creating exercise awareness for peers.

7. CONCLUSION

Body sensor networks will continue to play an important role in the development of healthcare applications as the need for lightweight and remote patient monitoring continues to grow. While body sensor networks are a fairly new topic, they correlate well with wireless sensor networks, which have seen a fair share of research. Many principles and design strategies have been tweaked and modified from WSN's to better serve BSN's compact, mobile nature. Because BSN's must be mobile and deal with important private healthcare information, low energy consumption and security are a requirement. These issues are especially important for wireless BSN's, because each node must individually power itself and wireless communication is easier to intrude upon. Many WSN's and BSN's have already been developed to overcome these challenges to allow the sharing and monitoring of health information remotely. In this paper we have presented

HatterHealthConnect, a network that combines the mobility and monitoring of a BSN with the interaction capabilities of social networks. With the advent of social networking and the increased ability to share data, people are able to communicate like never before. We believe that creating a tool to connect health/sensor data to a social media channel will help promote physical wellness by creating peer groups that can help motivate and encourage each others. HatterHealthConnect includes a reliable BSN that is easy to operate, as it can run on any mobile phone the uses Android OS 2.1 or greater and is completely wireless. Most users will already be comfortable using social networks as well, so it will not be difficult for them to access and interact with data on these sites. Promoting physical wellness is an important goal to focus on and self motivation is not always enough to encourage it. HatterHealthConnect is designed to meet that goal effectively by easily connecting users to a global network of those who want to pursue a healthier lifestyle.

7. REFERENCES

[1] Otal, B., L. Alonso, Ch. Verikoukis, "Towards Energy Saving Wireless Body Sensor Networks in Health Care Systems," Communications Workshops (ICC), 2008 IEEE International Conference on, Proc. of Communications Workshops, 2008, ICC Workshops '08. IEEE International Conference, Capetown, pp.1-5, 2010.

[2] Misra, S., Tiwari, V., Obaidat, M., “LACAS: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks,” IEEE J. Sel. Areas Commun., 27(4): 466-479, 2009.

[3] Gervasi, Osvaldo, Marina Gavrilova, Jungrae Kim, Ki-young Jang, Hyunseung Choo, and Won Kim, "Energy Efficient LEACH with TCP for Wireless Sensor Networks," Computational Science and Its Applications, ICCSA 2007 4706 (2007), pp. 275-85. Springer Berlin / Heidelberg. Web.

[4] Kuryloski, P., Giani, A., Giannantonio, R., Gilani, K., Gravina, R., Seppa, V., Seto, E., Shia, V., Wang, C., Yan, P., Yang, A.Y., Hyttinen, J., Sastry, S., Wicker, S., and Bajcsy, R. 2009. DexterNet: An Open Platform for Heterogeneous Body Sensor Networks and its Applications. Body Sensor Networks 2009 (BSN 2009 Berkeley, CA June 3-5, 2009)

[5] Seto, E., Martin, E., Yang, A., Yan, P., Gravina, R., Lin, I., Wang, C., Roy, M., Shia,

V., and Bajcsy, R. 2010. Opportunistic strategies for lightweight signal processing for body sensor networks. In Proceedings of the 3rd international Conference on Pervasive Technologies Related To Assistive Environments (Samos, Greece, June 23 - 25, 2010). F. Makedon, I. Maglogiannis, and S. Kapidakis, Eds. PETRA '10. ACM, New York, NY, 1-6.

[6] The SPINE Team. The spine manual version 1.2. Technical report, Telecom Italia Lab, 2008.

[7.] M. J. Moron, J. R. Luque, A. A. Botella, E. J. Cuberos, E. Casilari and A. Diaz-Estrella, "A smart phone-based personal area network for remote monitoring of biosignals," International Federation for Medical and Biological Engineering Proceedings, 2007.

[8.] B. Lo, S. Thiemjarus, R. King, and G.Z. Yang. Body sensor network - a wireless sensor platform for pervasive healthcare monitoring. In Adjunct Proceedings of the 3rd International Conference on Pervasive Computing, May 2005.

[10] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, “System architecture of a wireless body area sensor network for ubiquitous health monitoring,” Journal of Mobile Multimedia, vol. 1, no. 4, pp. 307–326, 2006.

[11] Munishwar, Vikram P., Samer S. Tilak, and Nael B. Abu-Ghazaleh, Congestion and Flow Control in Wireless Sensor Networks. Guide to Wireless Sensor Networks, Dordrecht: Springer, 2009.

[12] C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," Proc. 1st ACM conference on Wireless Network Security, pp. 148–153, 2008.

[13] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "EKG-based key agreement in Body Sensor Networks," IEEE Conference on Computer Communications Workshops (INFOCOM), pp. 1-6, 2008.

[14] K. K. Venkatasubramanian and S. K. S. Gupta, "Security for pervasive health monitoring sensor applications," Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP '06 Bangalore, India), pp. 197 – 202, December 18, 2006.

[15] W. Leister, T. Fretland and I. Balasingham, "Use of MPEG-21 for Security and Authentication in Biomedical Sensor Networks," 3rd International Conference on Systems and Networks Communications (ICSNC), Sliema, Malta, pp. 151 – 156, October 26, 2008.

[16] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc

sensor network infrastructure for emergency medical care," Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, pp. 12-14, June 6, 2004.

[17] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, and J. Stankovic, "An advanced wireless sensor network for health monitoring," Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), Apr. 2006.

[18] Chien-Chih Lai, Ren-Guey Lee, Chun-Chieh Hsiao , Hsin-Sheng Liu , Chun-Chang Chen, "A H-QoS-demand personalized home physiological monitoring system over a wireless multi-hop relay network for mobile home healthcare applications," Journal of Network and Computer Applications, v.32 n.6, p.1229-1241, November, 2009.

[19] B. Korel, S. Koo, "A Survery on Context-Aware Sensing for Body Sensor Networks, Wireless Sensor Network," Volume 2, p. 571-583, 2010.

[20] P. R. Pereira, A. Grilo, F. Rocha, M. S. Nunes, A. Casaca, C. Chaudet, P. Almstrom and M. Johansson, "End-to-end reliability in Wireless Sensor Networks: survey and research challenges," EuroFGI Workshop on IP QoS and Traffic Control, Lisbon, December, 2007.

[21] Al-Ali, A.R., Al-Rousan, M., Ozkul, T., “Implementation of experimental communication protocol for health monitoring of patients,” Computer Standards & Interfaces, Vol. 28, Issue 5, pp. 523-530, June 2006.

[22] Fang, W., Chen, J., Shu, L., Chu, T., Qian, D., “Congestion avoidance, detection and alleviation in wireless sensor networks,” J. Zhejiang Univ. 2010.

[23] Varshney, U., “Pervasive healthcare and wireless health monitoring,” Mob. Netw. Appl., 12(2-3), pp. 113-127, 2007.

[24] B. Otal, L. Alonso, and C. Verikoukis, “Highly reliable energy saving mac for wireless body sensor networks in healthcare systems,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, pp. 553–565, 2009.