

Remote Blood Pressure Monitoring Using a Wireless Sensor Network

William Walker, Todd Polk, Abhiman Hande, and Dinesh Bhatia
Department of Electrical Engineering
University of Texas at Dallas
email: {wpw021000, twp017000, abhiman.hande, dinesh}@utdallas.edu

Abstract-A system to remotely monitor a patient's blood pressure (BP) is described. The data is transferred to a central monitoring station using a wireless sensor network for display and storage. Crossbow MICAz motes were programmed to serve as the network nodes. One mote was interfaced to the BP monitor for data acquisition and the others were utilized to route the BP data to the monitoring station. A user-friendly graphical user interface was designed for monitoring current and past measurements for all patients being monitored. Test results indicated high accuracy in BP measurements. Power consumption by the BP monitor interfaced mote was minimized by forcing it into a low power sleep mode when not in use.

I. INTRODUCTION

Advances in wireless sensor network (WSN) technology and the overall miniaturization of their associated hardware are leading to several potential applications in the medical industry. In particular, the ability to remotely monitor patient vital signs in real time from a centralized location is a growing area of interest [1]. This interest in WSNs is fueled by the fact that wireless sensor nodes are cost effective, compact and can be energy efficient. Alternatives include WiFi and Bluetooth, which are focused on applications that normally require higher bandwidth. Wireless nodes using these two communication protocols are normally much more expensive and power hungry, and in the case of Bluetooth, allow a limited number of nodes to communicate at any given time. These issues make WiFi and Bluetooth nodes unsuitable for widespread remote monitoring of patient vital sign data. Additionally, the capability to do this without installing an expensive wired infrastructure is highly desirable. This paper describes a system using Crossbow 2.4GHz MICAz wireless sensor nodes [2], a commercial blood pressure monitor (BPM) and an internally developed Graphical User Interface (GUI) to design a prototype system that can monitor vital signs from a large number of patients simultaneously.

II. SYSTEM DESCRIPTION

A conceptual view of the system is shown in Figure 1. Each patient is connected to a remote monitoring system, which allows the medical staff to track the patient's vital signs. The vital sign readings are transmitted wirelessly from the patient through a fixed infrastructure of routing nodes to the base station. Depending on the patient's distance from the base station, messages can pass through multiple router nodes

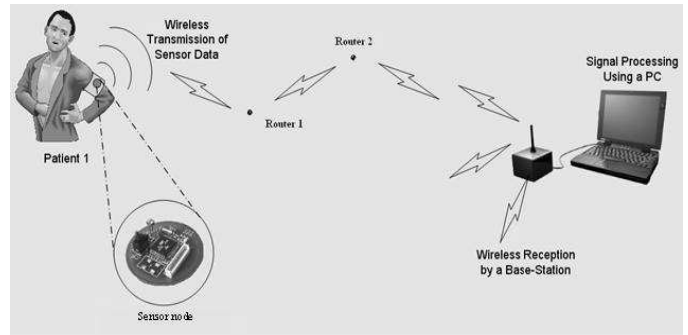


Fig. 1. Remote monitoring system

to reach the base station. The base station is connected to a host computer running a Java-based GUI to interpret and display the data.

There are three main areas of the system interface that we will discuss in detail below viz., sensor to BPM, sensor base station to host computer, and the human interface to the host computer (via the GUI). A brief description of the sensor network is also included.

A. Sensor to Blood Pressure Monitor Interface

A commercially available A&D UA-767PC BPM [3] is used to provide sensor readings for the system. The BPM takes simultaneous blood pressure and pulse rate measurements. It includes a serial port connection that facilitates bi-directional communication at 9600 kbps. A sensor node communicates with the BPM on this serial link to start the reading process and receive the patient's blood pressure and heart rate readings. Once the readings are received, the sensor node communicates with the network and transmits them to the base station.

The reading process is shown in the flowchart in Figure 2. To start the communication process with the BPM, the sensor node sends a start signal to the BPM to switch it to communication mode. Once the BPM is in communication mode, the sensor node sends a command to open the communication port. When the communication port is open, the BPM is ready to receive commands. A command to take a measurement is then issued. This causes the BPM to inflate the arm cuff and acquire the blood pressure and heart rate measurements. When the reading process has completed, the readings are sent to the sensor node. Limited processing is

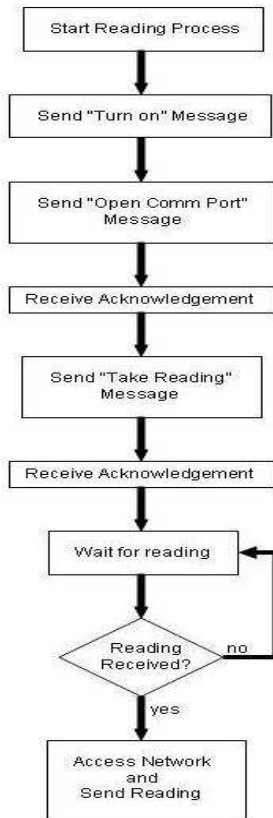


Fig. 2. BPM communication flowchart

performed by the sensor node on the reading data before transmitting it through the network to the base station¹.

Example communication messages are shown in Figure 3. All communication with the BPM is in ASCII format. The communication format is described in [4]. For example, if the number 60 needs to be sent, two bytes, 0x36 and 0x30 are sent (which are 6 and 0 in ASCII). The “Turn on” message is just an arbitrary byte. It simply tells the BPM to wake up and be prepared to receive commands.

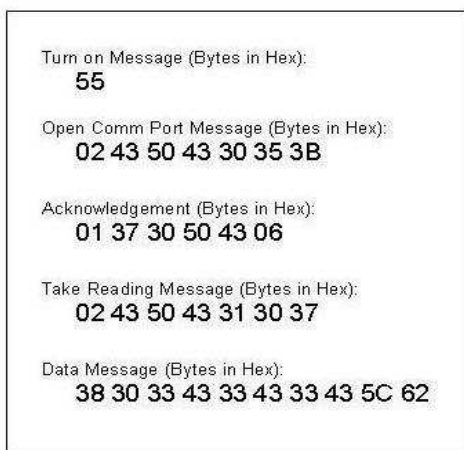


Fig. 3. BPM serial port example messages

The “Open Comm Port” message is structured as follows. **02** identifies the message as a command message. The next byte (**43**) is fixed and represents **C** in ASCII. The next two bytes (**50** and **43**) describe the device sending the message, which in this case is the sensor node and is **PC** in ASCII. The following two bytes (**30** and **35**) are the open communications command (**05**). The final byte (**3B**) is the least significant byte of the checksum of the message.

The “Acknowledgement” message contains a **01** in the first byte as an identifier signaling the message as a control message. The next two bytes again describe the device sending the message. Since this message is being sent by the BPM, it is **70** in ASCII. The following two bytes detail who is to receive the message, and in this example is **PC** which represents the sensor node. The final byte, **06**, identifies this control message as an acknowledgement.

The “Take Reading” message follows the same format as the “Open Comm Port” message. An ASCII **1** and **0** are sent as the command bytes. The first two bytes of the “Data” message are fixed and represent **80** in ASCII. The next two bytes represent the hex value of the systolic reading minus the diastolic reading in ASCII. From Figure 3, it can be seen that this value equals **3C**. The following two bytes are the ASCII representation for the hex value of the diastolic reading. This value also equals **3C** in ASCII. Therefore, in this case, the systolic reading of the patient is 120 and the diastolic reading is 60. The next two bytes are the ASCII representation of the hex value for the pulse rate, and in this example, it is 60 beats per minute. In the present prototype, the measurement process is started when the sensor node is turned on. In future implementations, multiple medical monitors, such as the BPM, a pulse oximeter and an electrocardiogram (ECG) will be integrated with a single sensor node and readings will be performed on a scheduled basis.

B. Sensor Base Station to Host Computer Interface

Java is used to implement the interface between the base station and the host computer. All communication between the base station and the PC is done through the UART. A separate thread from the main graphical user interface thread is used to maintain constant monitoring of the serial port. When a message is received, the message type is determined. If the message is a data message sent from a sensor node (i.e., a measurement from one of the medical monitors), the data is extracted and stored according to the ID of the sensor node that took the reading. If the message is a control message, then the information is passed directly to the program thread running the GUI. Control messages contain network information used to generate a network view for the user. A control message from a sensor node includes the ID of the sensor node and the ID of the router node used as the initial entry point into the network. A control message generated by a router node contains the ID of the originating router node itself and the ID of the router’s partner node. The information contained in the control message is used to generate a map of the network. In the current implementation, the map is a

¹ Currently, no encryption is employed. The data does have some inherent level of security due to the custom nature of the transmitted data packet.

simple tree depicting the network structure.

C. The GUI

The graphical user interface is a Java-based program running on the host computer. The main window of the GUI is shown in Figure 4. Here, the user can add a new patient to the monitoring network at any time by keying in the patient name and the ID of his assigned sensor node. If the patient name resembles other names in the database, a list will appear allowing the user to select the right patient from the database. Once entered in the system, all readings received by the PC from the patient's sensor node, will be stored in an object corresponding to the patient. All stored readings can be viewed by selecting the desired patient name in the list shown in Figure 4 and selecting the "View" option. This action will open the window in Figure 5, which allows the user to view various readings and edit patient information. The current implementation displays a text-only readout of the measurements; future versions will have a graphical readout. When a reading is received, its value is checked against the threshold limits seen in Figure 5 and if it is beyond these limits, a warning message window appears to notify the user of the newest reading. Readings are not accepted from nodes that are not registered to a patient and will also generate a warning message. In future implementations, the addition of a patient will trigger a database access.

D. Network Description

The topology of our wireless network is that of a static routing infrastructure with mobile sensor nodes. Energy scavenging is employed at the router nodes so that connection to a permanent power supply is not needed. This also eliminates the need for batteries which need to be monitored and replaced. Scavenging energy from indoor light sources is an inefficient process and results in limited amounts of energy to power the router nodes [5]. To compensate for this, the router nodes operate in synchronized pairs and at a 50% duty cycle. This allows sufficient time during the off cycle for each node to scavenge the energy required for its next on cycle.

In the process of joining the network, each router node

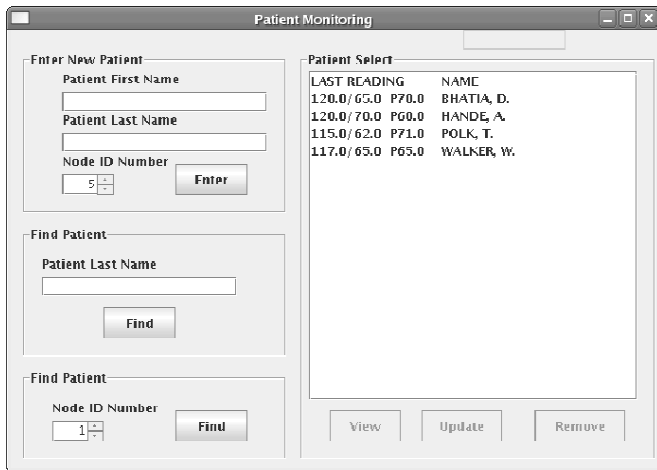


Fig. 4. GUI main window

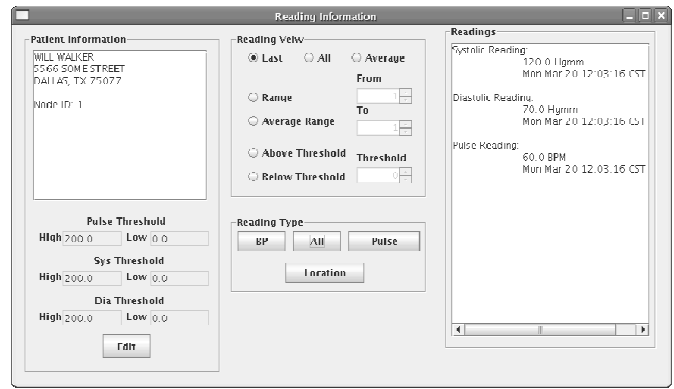


Fig. 5. GUI sub window

identifies its shortest path to the base station (i.e., the minimum number of hops). When this is complete it then searches for another node to pair with. The flowchart in Figure 6 displays the process the router nodes go through to establish themselves in the network.

Once the routing infrastructure is set up, a sensor node can easily access the network. The sensor node chooses the shortest path to the base station based on the information sent from nearby router nodes and sends the readings to the base station. The flowchart in Figure 7 details the process a sensor node goes through to send a reading.

Sensor nodes are designed to be mobile to allow patient movement. Thus, each subsequent reading requires the sensor node to reconnect to the network in potentially a different location. This pattern also allows the sensor node to sleep between readings, thereby extending battery life.

III. PRELIMINARY RESULTS

A prototype system containing all 3 interfaces described above has been built and tested. Utilizing a sensor node tied to the BPM we have successfully initiated a reading, gathered the data and can forward it through the network to the sensor base station. The measurements are then forwarded through the serial port to the host computer, and the GUI displays the data correctly.

We have also programmed multiple sensor nodes to represent multiple patients and interfaced them with the BPM. Figure 4 shows four patients with their own unique sensor node IDs. Measurements for each patient were successfully transmitted to the base station and then forwarded to the GUI for display. All readings were correctly maintained for each patient, and the display is similar to that shown in Figure 5. It can be seen from Figure 5 that the pulse rate and BP reading for patient Will Walker taken on March 20 at 12:03 pm was 60 and 120/70 respectively. These values were found to be exactly equal to those displayed on the BPM LCD display.

IV. FUTURE WORK

The current implementation, as discussed above, is the first version of our system. Future enhancements to the system include:

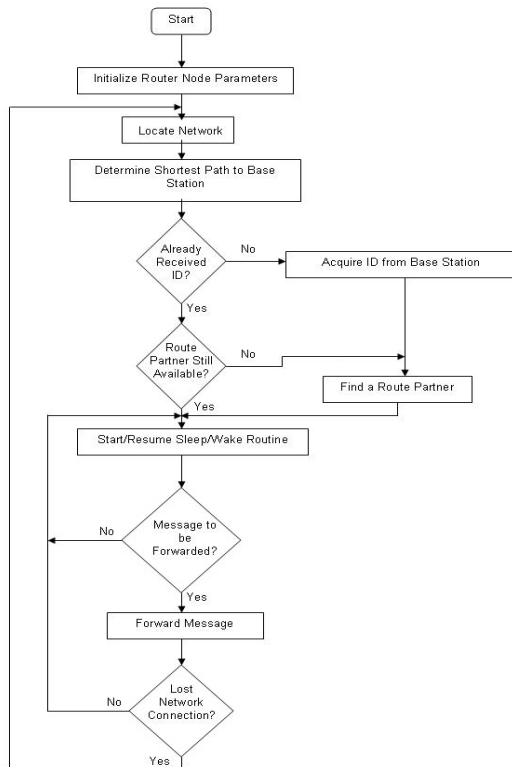


Fig. 6. Router node flowchart

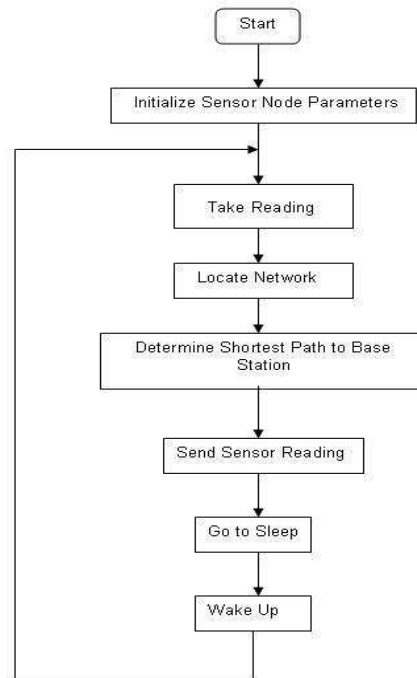


Fig. 7. Sensor node flowchart

- A graphical display of the incoming data to replace the current text display.
- An alarm generation capability to alert the care provider of a reading outside the given limits. This alert will be automatically sent to a PDA or similar device.
- Interfacing of additional medical instruments, including a pulse oximeter and a portable ECG.
- The ability for the care provider to view stored readings remotely from a PDA or computer.
- A basic location algorithm, based on the fixed router infrastructure, to track patient movement.
- A data encryption routine that can effectively run on each sensor node to further protect the transmitted data (and meet HIPAA requirements).
- Increased integration with existing hospital systems and databases.

V. CONCLUSIONS

A prototype BP monitoring system using WSNs has been designed, developed and tested. The system allows health personnel to monitor a patient's BP and heart rate vital signs from a remote location without requiring the physician to be physically present to take the measurements. The system concept can be used for routing vital sign information to a central location within the hospital premises as well as in applications that require monitoring from within a patient's

home. Preliminary tests showed that the BP and heart rate measurements for four patients were fairly accurate and equaled the measurements displayed on the monitor LCD display. Further research (as indicated in section IV) will focus towards integrating smaller OEM medical sensor boards to the Crossbow sensor nodes to miniaturize their size and make the system more practical.

ACKNOWLEDGMENTS

This research was supported by an internal grant from the University of Texas at Dallas (UTD) Erik Jonsson School of Engineering and Computer Science, and the work was conducted at UTD's Wireless Sensor Development Group (WSDG) laboratory. The authors would like to thank the support received from UTD and acknowledge the advice and assistance provided by Dr. Padmakar Kulkarni from University of Texas Southwestern Medical Center.

REFERENCES

- [1] K. Lorincz, et. al, "Sensor Networks for Emergency Response: Challenges and Opportunities", *Pervasive Computing, IEEE*, vol.3,no.4, pp. 16- 23, Oct.-Dec. 2004.
- [2] A & D Medical, Inc., "UA-767BT Wireless Blood Pressure Monitor", 2005, <http://www.lifeforceonline.com/products/telemonitoring.cfm>.
- [3] Crossbow Technology Inc., "MPR/MIB User's Manual", Rev. B, Document 7430-0021-06, 2005, http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf.
- [4] "UA-767PC RS-232 Command Sets and DataFormat Specification v2.1", A & D Medical, Inc., March 2004.
- [5] J.F. Randall and J. Jacot, "The Performance and Modelling of 8 Photovoltaic Materials under Variable Light Intensity and Spectra", World Renewable Energy Conference VII Proceedings, Cologne, Germany, 2002.