Design and development of a prototype for an inexpensive, multi-node mobile, Palm-PDA-based wireless clinical information system platform

Nicholas O'Donoughue
Villanova University
nicholas.odonoughue@villanova.edu
Sarvesh Kulkarni
Villanova University
sarvesh.kulkami@villanova.edu

Douglas Marzella
Villanova University
douglas.marzela@villanova.edu
Elliot B. Sloane
Villanova University
elliot.sloane@villanova.edu

ABSTRACT

In this paper, we describe the design, development, and testing of an inexpensive wireless intelligent patient monitoring system. The system uses commercially-available wireless data acquisition links that operate on non-IEEE-802.x frequencies to reduce privacy and security concerns. It uses the more readily secured and robust wired Ethernet LAN to move data through the hospital to a PC-based Ethernet server. This server stores the patient data in a relational database and also includes an intelligent agent to send "alerts" to a clinical user whenever patient data exceeds safe limits. Finally, there an IEEE-802.11x wireless network is used to allow the clinical user to view patient data and receive intelligent alerts in real-time on a Palm PDA wherever they are in the hospital. As hospitals move towards the Integrated Healthcare Enterprise (www.ihe.net), and federated Electronic Health Records and Regional Healthcare Information Organizations (senior goals of the US federal healthcare information technology strategy www.ANSI.org/HITSP), reliable, secure, and readily deployable mobile medical device networks will play an important role in automatic transfer of patient data to many parts of the healthcare enterprise.

Keywords

Wireless medical device networks, intelligent agents, medical devices, ubiquitous computing.

INTRODUCTION

This paper describes a Senior Electrical Engineering design project that built a working prototype of a wireless, mobile, Palm-PDA-based Clinical Information System. The major components of this system included the following:

1. A network of simulated miniature patient sensors that moves data to a central wireless hub. The radio frequency selected for this hub is different than that used by common IEEE 802.x networks to improve security and privacy, per HIPAA requirements (PrivacyRights.org, 2006) and to avoid conflict with potential in-hospital deployment of IEEE 802.x networks for other applications,
2. For reliability, privacy, and secure operation, all wireless links were connected to a wired Ethernet LAN network via the matched proprietary wireless hub.
3. A PC-based Ethernet server was used to create a software-driven Data Acquisition and Alert Module (DAAM), to record all sensor data in a relational database (MySQL) for record keeping. This server was designed to include an intelligent-agent-based and "alert" handling subsystem to notify the user if patient data exceeded specified thresholds. In some cases these "alerts" could represent life-critical alarms, but they could also represent trend- or exception-driven notices established by the hospital or the individual clinician.
4. A Palm-PDA-based intelligent agent to display physiologic data and "alert" information to the clinical user via an standard IEEE 802.11x which could be secured with WPA or a proprietary algorithm to comply with HIPAA privacy policies.

This paper focuses on the project components 3 and 4 above, as they represent a novel approach of organizing, evaluating, and communicating patient data and "alert" situations to a clinical user via ubiquitous 802.11x links.

The following primary objectives were considered vital to the success of our project, and had to be met for a successful execution of the project:

- To create a network of sensors, relays, and a base station, a Wireless Sensor Network (WSN), that is capable transferring simulated patient data patient via a proprietary wireless network available on any local wired Ethernet,
- To create the PC-based DAAM that receives patient data and stores that information in a MySQL relational database for user retrieval AND for intelligent alerting agent scanning to enhance patient safety,
- To implement an intelligent-agent subroutine in the DAAM that will send alerts to the ICUClient program whenever data is outside a specified safe range, and
- To create an ICUClient software program that will enable a Palm OS®[1] device to retrieve and display the previously mentioned sensor data.

DESIGN OVERVIEW

The design of our system consists of three main components, as seen below in Figure 1. The first subsystem, the WSN (red) is responsible for collecting and transmitting all relevant patient data. The second subsystem, the DAAM (blue), is responsible for recording and monitoring all received data, as well as managing connections with all of the Palm clients. The third subsystem, the *ICUClient* (green), is responsible for accessing the DAAM to retrieve patient data, and displaying that data for the doctors and nurses, hereafter referred to as *users*.

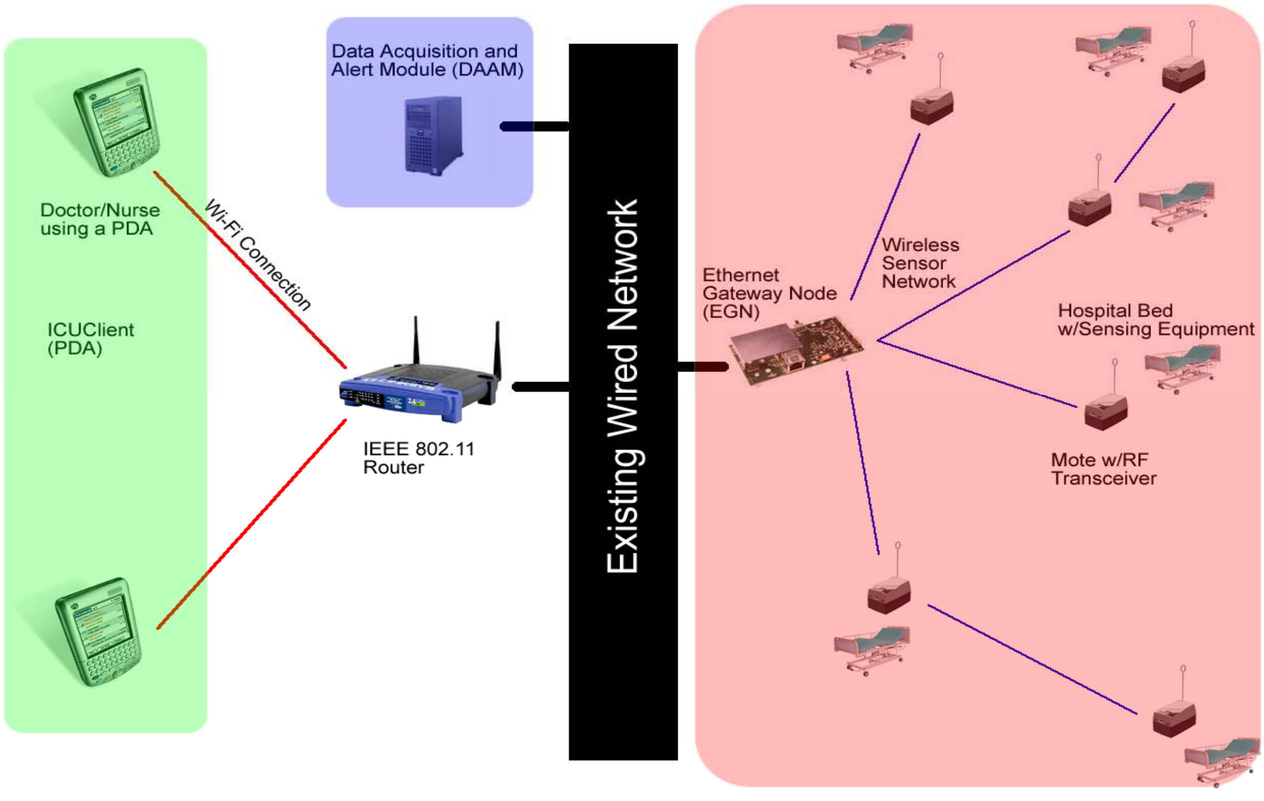

Figure 1 Overall system design (Based on images at www.xbow.com, www.linksys.com, www.palm.com, and www.dell.com)

[1] Palm OS® is a registered trademark of Palm, Inc.

Wireless Sensor Network (WSN)

The WSN consists of Crossbow *Mica2* and *Mica2Dot* motes (figure 2, noting that the MICA2DOT sensors are about the size of a US quarter.). The motes were purchased from Crossbow, Inc. <www.xbow.com>. The motes were configured with the vendor-supplied Surge Reliable Route routing algorithm, which runs on the TinyOS®[2] platform that is available from Crossbow, Inc. The base station of the WSN, the Crossbow MIB 600 (see bottom of figure 2(c)), receives all of the data transmissions from the motes and makes them available over the attached Ethernet port. A "node" is hereby described as any mote in the WSN. In our proposed solution, patients would be issued a unique mote, and that mote would be tied to their personal records through a database in the DAAM. The mote itself would be situated near, or attached to the patient's hospital bed and would be able to transmit information to the DAAM server over multi-hop wireless links. These multi-hop wireless links are formed by the motes operating in a co-operative fashion in order to transport each-others data traffic. The WSN is also self-healing and reconfigurable. Consequently, if any of the hospital beds (with the motes attached to them) are moved around, they can reconfigure themselves into a WSN provided they stay in communication range with one another.

Figure 2 Wireless sensor network, including base station that provides the receiver-to-Ethernet link, at bottom (www.xbow.com)

Data Acquisition and Alert Module (DAAM)

The second subsystem is the DAAM (figure 3). The main component in the DAAM is the *Logger* module. It opens a socket to the MIB600 in the WSN and stores all the data into a MySQL®[3] database. The *Logger* is also responsible for monitoring the incoming data for what we have described as "emergency conditions." Whenever an incoming data point is flagged as an emergency, a message is sent via a method call to the *PalmServer*, which is a running process on the same machine.

The MySQL database (figure 3) consists of three tables. The first table in the database, "nodes," is a list of all nodes in the network, which contains pertinent information about each node, including type, and time of last transmission. The second table, called "packetReceived," is a list of each transmission that was received from the sensor network, as well as all the data carried in those transmissions. The third table, "safeRange," is a list of the accepted safe ranges for each data set.

The next module in the DAAM is the *PalmServer* module. The *PalmServer* is responsible for listening for connections from users via a socket interface. As each connection request comes in, the *PalmServer* reallocates a new, random, port for that connection and hands it off to a new instance of the *PalmServerThread* module. The *PalmServer* is also responsible for keeping track of all the active *PalmServerThread*s. This way, whenever an "alert" is generated by the *Logger*, corresponding to a set of data that is outside pre-defined safe ranges, the *PalmServer* can effectively dispatch this alert to all the concerned *users*. Currently, the *PalmServer* alerts every user, but the inclusion of control and decision logic into this process to comply with a hospital's HIPAA specifications would be very straightforward.

Continuing in the data flow, we discuss next the *PalmServerThread* module. This module is instantiated by the *PalmServer* module with every incoming connection. It handles communications with the ICUClient, queries the MySQL database in response to requests for data, and formats the results appropriately before transmission. This module is used to send information to the user about what nodes are "active" in the sensor network, and to provide a "snapshot" of the most recent data transmission received from a selected node. If the user wishes to receive streaming information on a specific data set (such as pulse or body temperature in a final system, and light or acoustic levels in our concept system), then the

[2] TinyOS® is a registered trademark of UC Berkeley

[3] MySQL® is a registered trademark of MySQL AB

PalmServerThread hands this task off to a new instance of the *UpdateThread*. This is done so that the *PalmServerThread* can listen for messages from the user without any lag in response time.

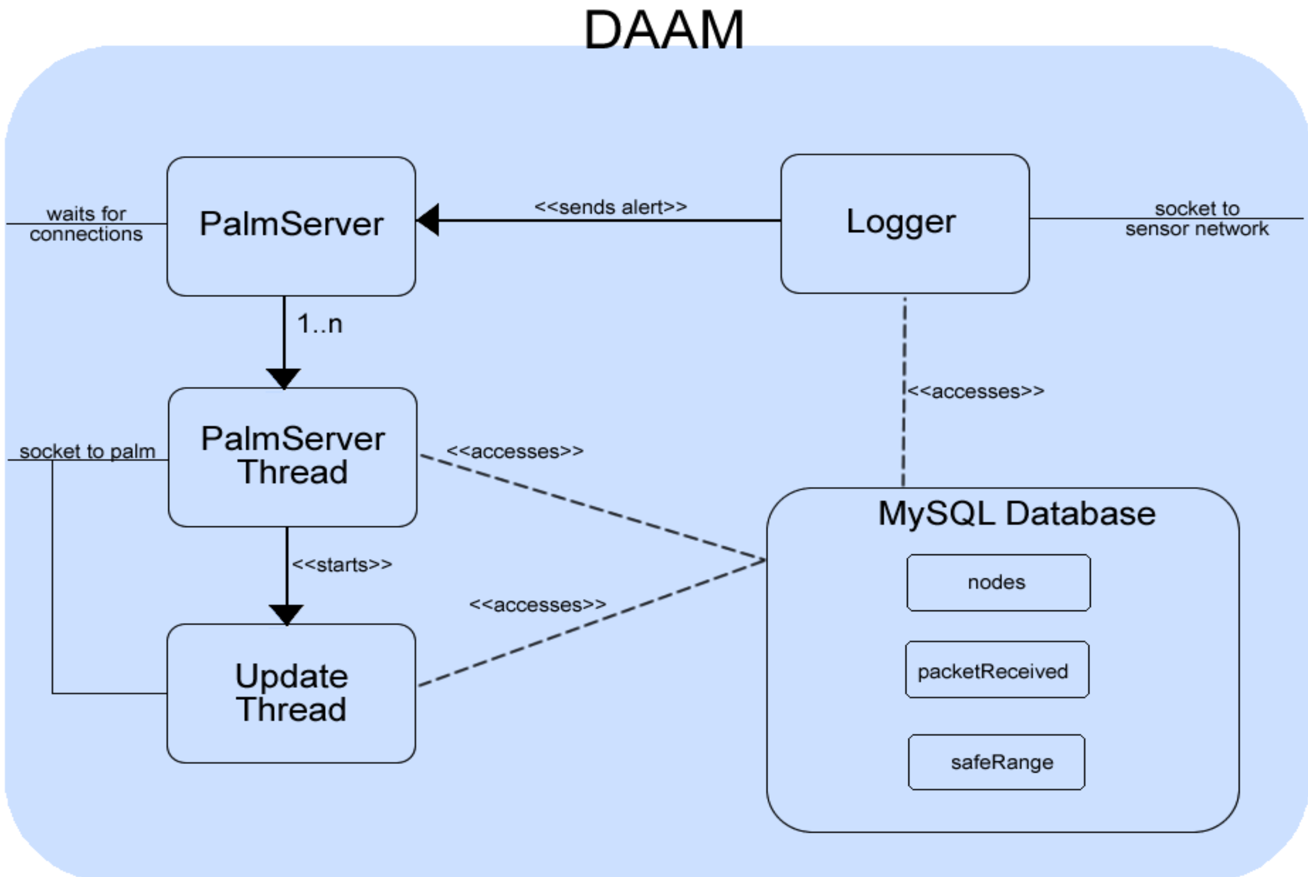

Figure 3 DAAM Overview, including MySQL relational database tables

The final module in the DAAM is the *UpdateThread* module. This module is straightforward. Upon instantiation, it receives the identification number of a node and a sensor attached to that node, as well as the socket handle for outgoing communications to the user. The *UpdateThread* queries the MySQL database for all transmissions received from that node within recent history, and then transmits the data points, in order, to the user.

ICUClient

The final component, the program which runs on a Palm OS handheld, is the *ICUClient* (Figure 4). All logic control for this program lies in the *ICUClientMIDlet* class, with all socket functionality handled by the *Communicator* class. The five 'screen' classes comprise the visual interface, and the *Graph* class handles the actual plotting of received data streams.

This application has been designed to robustly respond to all messages from the server and allow only appropriate selection on the part of the user, based on the current state.

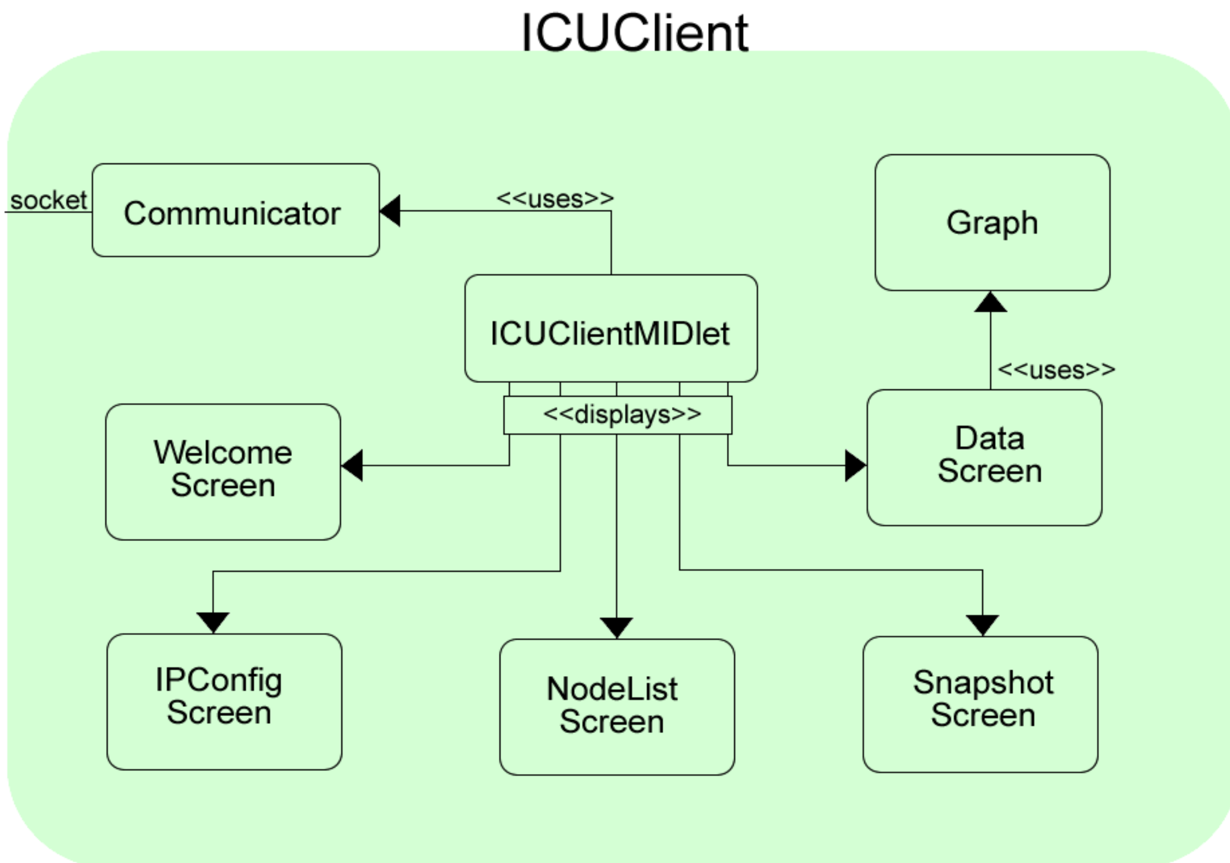

Figure 4 ICUClient overview

The finished application allows the user to browse all active nodes, view 'snapshots' of the most recent transmission received from that node, and view streaming data from a specific node and sensor graphically. While the user is browsing, the system is always ready to receive and immediately display an alert if one is sent by the DAAM.

TESTING

This wireless patient monitoring network was tested in a large, multi-room commercial building that exceeds the footprint of many ICU areas. The DAAM server was located in a different building, to simulate a multi-building hospital setting. The Palm Tungsten PDA was operated via an IEEE 802.11 network co-located with the DAAM server.

All components operated as designed, demonstrating successful completion of the project's objectives.

DISCUSSION AND CONCLUSIONS

We successfully completed all originally stated primary objectives, and we have created and successfully tested a prototype of this system. We were able to develop our system using Java (J2SE for the DAAM and J2ME for the ICUClient.) and the total retail hardware cost of this 8-node prototype is less than $5,000. This system should be quite affordable to maintain as well as manufacture in large quantities.

There are numerous remaining hurdles to turn this into a commercial medical device, including, but not limited to, the following:

- Physical interfacing of existing hospital sensing equipment with the radio nodes of the ad-hoc sensor network and testing to ensure all physiologic data integrity requirements are met.
- Addressing risks of patient data loss if/when patients move out of wireless range or other system failures occur.

- Configuration of the MySQL database used to store all information that is gathered from the patients, and compliance with HIPAA regulations.
- Enhancement of the DAAM to allow selective clinical user access and alerting, to comply with HIPAA requirements.
- Configuring the IEEE 802.11x for WPA or other acceptable security methods for HIPAA compliance.
- Compliance of the system to FDA Good Manufacturing Practices.

The successful completion of this design project and prototype means several things. First, it is possible to implement an inexpensive, multi-node, wireless system whereby patient data stored in a central server can be accessed remotely from a handheld device and displayed graphically in real-time. At this point, at least a successful prototype and framework have been created to allow potential implementation of a market-ready system.

REFERENCES

1. Figure 1, Mote and Gateway Images from <www.xbow.com>.
2. Figure 1, Router Image from <www.linksys.com>.
3. Figure 1, PDA Image from <www.palm.com>.
4. Figure 1, Server Image from <www.dell.com>.
5. PalmSource, Inc. "Palm OS Developer Suite 1.1" <www.Palm OS.com/dev/tools/dev_suite.html>.
6. PalmSource, Inc. Code Samples. "SockNotifyTestApp.zip" <http://kb.palmsource.com/cgi-bin/palmsource.cfg/php/enduser/fattach_get.php?p_sid=Tfy_g*Ch&p_tbl=9&p_id=643&p_created=1077642523>.
7. Stevens, W. Richard, Unix Network Programming, Prentice Hall: New York, 1990.
8. Thorn, Jeff, "Deciphering TinyOS Serial Packets," Octave Tech Brief #5-01, 10 March 2005, <http://www.octavetech.com/pubs/TB5-01%20Deciphering%20TinyOS%20Serial%20Packets.pdf>.
9. "Version 2.0 of MIDP Specification," Sun Microsystems, 2002, <j2medev.com/api/midp/overview-tree.html >.
10. Mahmoud, Qusay, "MIDP for Palm OS 1.0: Developing Java Applications for Palm OS Devices," Sun Developer Network, January 2002, <developers.sun.com/techtopics/mobility/midp/articles/palm/index.html>.
11. HIPAA Basics: Medical Privacy in the Electronic Age, December 2005, Privacy Rights Clearinghouse, 19 February 2006, <http://www.privacyrights.org/fs/fs8a-hipaa.htm>.
12. IEEE 754: Standard Floating-Point Arithmetic, 11 Feb. 2006, IEEE Standards Group 754, 22 Feb. 2006, <http://grouper.ieee.org/groups/754>.