AYUSHMAN: A WIRELESS SENSOR NETWORK BASED HEALTH MONITORING INFRASTRUCTURE AND TESTBED*

K. Venkatasubramanian, G. Deng, T. Mukherjee, J. Quintero, V. Annamalai and S. K. S. Gupta
Dept. of Computer Science and Engineering
Arizona State University
Tempe, AZ, 85281, USA
{kkv, guofeng, tridib, jquin, valliappan, sandeep}@asu.edu

Abstract

With a rapidly aging population, the health-care community will soon face a severe medical personnel shortage. It is imperative that automated health monitoring technologies be developed to help meet this shortage. In this direction, we are developing Ayushman, a health monitoring infrastructure and testbed. The vision behind its development is two-fold: first, to develop a wireless sensor-based automated health monitoring system that can be used in diverse situations, from home-based care, to disaster situations, without much customization; second, to provide a testbed for implementing and testing communication protocols and systems. Ayushman provides a collection of services which enables it to perform this dual role. It possess a hierarchical cluster topology which provides a fault-tolerant and reliable system by ensuring that each tier in the hierarchy is self-contained and can survive on its own in case of network partition. Ayushman is being implemented using off-the-shelf and diverse hardware and software components, which presents many challenges in system integration and operational reliability. This is an ongoing project at the IMPACT lab at Arizona State University[1], and in this paper, we present our system's architecture and some of our experiences in the development of its initial prototype.

Keywords: health monitoring, wireless sensor networks, testbed, dependability

1. Introduction

The health-care system in developed countries will increasingly come under pressure as the average age of their populations increases and the number of el-

*This work is supported by NSF grant ANI-0086020

derly people swells. This will most likely lead to dire shortages of health-care personnel, and, if left unattended, could result in a drop in the quality of medical care and a substantial increase in health-care costs. However, this trend can be checked by the introduction of automated remote health monitoring technologies that will monitor a persons health and alert appropriate health-care personnel in case of emergencies, thereby improving the chances of a patient's successful recovery.

Recent years have seen large-scale innovation and development in the field of small, low-powered, low cost sensing devices. These devices have the ability to communicate over wireless channels and can also perform data computation to a limited extent. Such sensors, owing to their form factor and processing capabilities, can play an important role in automated health monitoring systems, as they can be placed on a patient's body to collect continuous and real-time medical data (blood glucose, EKG). The presence of such a sensor or group of sensors will be totally non-intrusive compared with today's technology which ties a patient down to a room or limits her mobility. A patient wearing portable medical sensors can go about her work as usual while the sensors are collecting and analyzing her medical information.

In this paper, we at the IMPACT lab at Arizona State University present our experiences with the development of a health-monitoring infrastructure and test-bed called **Ayushman**. Our *motivation* for developing Ayushman is two fold: first, to develop a dependable, secure and scalable automated health monitoring system that can be used in diverse scenarios, from home based monitoring to disaster relief, with minimal customization; second, to provide a realistic environment (test-bed) for testing communication protocols and systems. We are developing Ayushman using completely off-the-shelf components and technologies, and making these diverse components and technologies work together is one of the major challenges we face in Ayushman's development.

Though many research projects have been pursuing the goal of remote health monitoring using sensor networks like [4][7][8][9]and [10], none of them share the Ayushman's vision of having a dual set of functionalities of a testbed and a fully functional health monitoring infrastructure.

The paper is organized as follows: Section 2 presents the challenges in developing such an infrastructure. Section 3 presents the Ayushman architecture, followed by Section 4 which recounts our experiences building our infrastructure. Finally Section 5 presents our conclusions.

2. Challenges

Building the infrastructure we envision presents many challenges. To be able to develop an effective medical monitoring system and a realistic testbed, we need Ayushman to possess the following capabilities:

- *Dependability :* Dependability is defined as the ability of a system to be able to avoid service failures that are more frequent and severe than acceptable [6]. Any health monitoring system, in order to be accepted by users, has to be dependable; that is, it must be able to maintain a high level of reliability (communication/ hardware/software), availability and fault-tolerance. Our vision for Ayushman is to be able to install it in multiple scenarios as a health monitoring infrastructure, which requires it to be flexible enough to provide the required level of dependability based on each scenario. Moreover, as a testbed, Ayushman's dependability should be adjustable in nature, to allow developers to test their protocols in different environmental conditions.
- *Long-term Data Collection :* Ayushman should provide continuous data collection facilities. By continuous we mean that the system should have the capability to collect data at any time, in response to an event or a query, and for a sizable length of time.
- *Energy Efficiency:* Health monitoring sensors are both Ayushman's greatest asset and its weakest link. A sensor, being small in size, can be completely non-intrusive to a patient, protecting her privacy and allowing her to be mobile. However, it also has limited battery power, and therefore has to be energy efficient.
- *Real-time Information Gathering :* Ayushman has to collect information from patients in real-time. This is necessary for two reasons: one, to alert a medical professional immediately in case the health of a patient deteriorates, and two, to allow a medical professional to view the patient's current state of health at any time.
- *Complete Information :* The data collected from patient has to be complete to allow a medical professional to accurately diagnosis a patient's condition. This requirement is different for each type of medical data (EKG,blood glucose) and heavily influences the design of the system, including parameters such as the data sampling rate, the required bandwidth, memory use etc.
- *Security :* Wireless communication is highly susceptible to eavesdropping and spoofing. However, the sensitive nature of health information requires most communication to be secure, at least in terms of information integrity, confidentiality, and the authentication of communicating parties. Peoples health and lives are at stake and the importance of security cannot be overstated.
- *Performance Analysis and System Controlling Tools :* Ayushman, as a test-bed, needs to provide developers with the ability to evaluate their

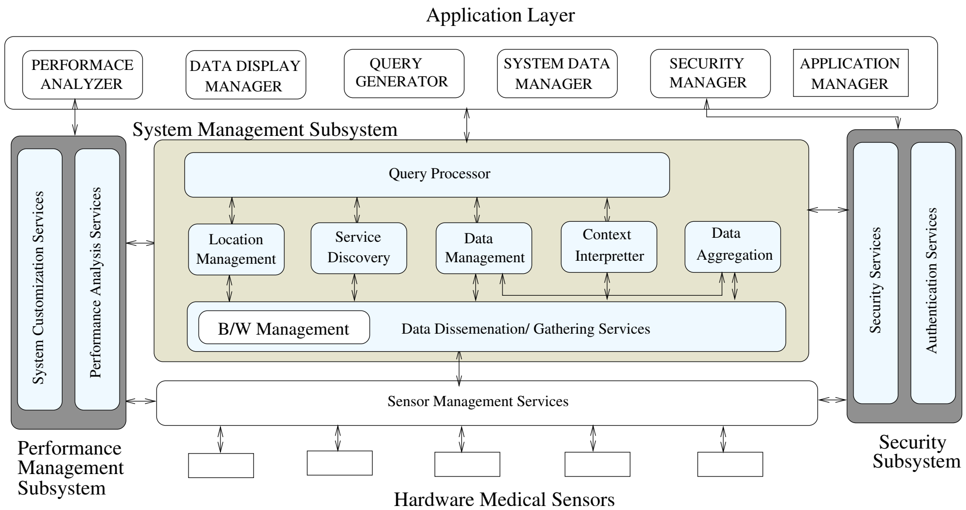

Figure 1. **Ayushman: Architecture**

protocol's performance in different scenarios. It should therefore provide tools to finely control the whole system and also to monitor the performance of each component of the system.

3. Architecture

The essential functions of Ayushman have been implemented as a set of services as shown in Figure 1. In this section, we present in some detail the different services provided by the Ayushman architecture.

- At the lowest layer are the hardware medical sensors, which measure medical and other context information (location, time, physical state, work state) from patients. The information from the sensors is passed on to a *Sensor Management Services* layer, which provides an abstraction of the hardware to the upper layer services. The upper layer can control the working of the sensors through a standardized interface provided by the Sensor Management Services layer.

- The Sensor Management Services layer forwards the sensor data to a **System Management Subsystem**, which provides services for effective patient health monitoring. Some of the services provided include:

  – *Data Dissemination and Gathering Service :* This service provides effective communication primitives for gathering and disseminating data within the infrastructure, and it often uses multicasting, flooding and unicast primitives. It also implements a *Bandwidth*

Manager, which manages the bandwidth resource between various components of the infrastructure.

  - *Location Management Service :* It is used to provide location information of patients, doctors and equipment within a medical facility based on sensor data.
  - *Context Interpreter Service :* It is used to interpret patient's context information sent by the sensors.
  - *Data Management Service:* Ayushman is designed to collect both medical and administrative data from patients. In order to manage the data it collects, it provides two functions: buffer management, and storage management. The former is used mostly for synchronization between different parts of the system after a recent disconnection, while the latter is used mostly for long-term storage of medical data.
  - *Service Discovery Service :* A medical facility provides many services, and it is essential for an automated medical monitoring system to be able to discover them, register them, and present information about them when queried. Ayushman's service discovery allows applications to search for available medical services.
  - *Data Aggregation Service :* A system like Ayushman will collect lots of health-related data for each patient, and it is essential to be able to aggregate, analyze, and extract information from such data. The result of th data analysis and aggregated values are sent to the data management service for appropriate storage.
  - *Query Processing Service :* This service gets queries from the application layer, queries the above mentioned services and sends the data back. It can be used to query any information about the infrastructure, from location, to medical and administrative data, to context information.

- Ayushman is also being designed as a testbed, and in order to be able to monitor and control the system, it provides a **Performance Management Subsystem**. The idea is to allow developers to plug in their protocols, change the properties of the system, and measure the protocol's performance. Two services enable this feature:

  - *Performance Analysis Service :* This is a cross-layer service which is used to obtain performance metrics from the System Management Service. This is different from the query processor, which queries patient related data.
  - *System Customization Services :* This is another cross layer service which, on a cue from an application, can change the properties of

the system (e.g.: change bandwidth available to specific medical sensors). This can be used by protocol developers to test their work under different conditions.

- Medical information that Ayushman collects has to be kept secure in order to maintain patients' privacy. The HIPAA (Health Insurance Privacy and Accountability Act) laws [5]mandate that any electronic medical record has to be secure to prevent unauthorized access. As security is important at all levels of a health monitoring system, Ayushman provides another cross layer service for providing security and authentication to all layers within the system through the **Security Subsystem**. The Security Subsystem can be controlled from the application layer to provide variable level of security to different parts of Ayushman.

- The top-most layer of Ayushman is the application layer, which provides applications for managing the whole system. Some of the applications provided include:

  – *Performance Analyzer :* This interacts with the Performance Management Subsystem and can be used by developers to monitor and control system properties.
  – *Data Display :* This application is used for providing a meaningful way of displaying patient's medical data.
  – *Query Generator :* This is used to query data from the system, which it does by contacting the query processor.
  – *System Data Manager :* This is used to monitor and interpret all medical and administrative data in the system.
  – *Security Manager :* This is used to monitor the system security settings and to change them when needed.
  – *Application Manager :* This application, as the name suggests, is used to monitor and manage resource allocation for all other applications.

So far, we presented an abstract view of Ayushman's architecture. Actually Ayushman has been implemented with a hierarchical cluster topology. At the lowest tier are the *sensors*, which collect medical data. The sensors on each patient form clusters which send their data to a leader node called the *local gateway*. The local gateway, implemented on a PDA, is responsible for managing the sensor network: collecting, analyzing and storing information from the sensors. The local gateways form the second-tier of clusters, and send their own data to their leader entity called the *external gateway*. The external gateways form the third-tier of the system, and their cluster leader is the *central*

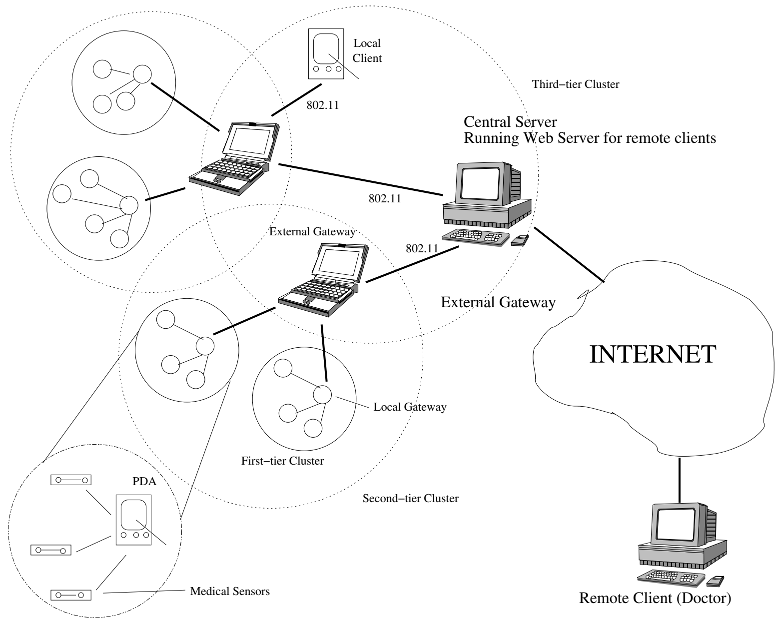

Figure 2. **Ayushman: Physical Organization**

server, which manages the entire Ayushman infrastructure. Each tier is designed to be self-contained in terms of data collection, data storage, quality management, data analysis, and can manage without control from its cluster leader. This feature provides system fault-tolerance which can function even during disconnections and network partitions. We have also implemented a client device for accessing data from Ayushman. Our clients are of two types: local and remote. The local clients, implemented using a PDA or laptop, can contact the leader entity in all three tiers, and after proper clearances, can access patient medical data. Remote clients use the Internet to access Ayushman. The central server provides a web-based access to the remote clients. Figure 2 presents the physical organization of Ayushman.

4. Experiences

Now that we have presented Ayushman's architecture, we would like to share some of our experiences with developing it. As mentioned, we are building Ayushman using off-the-shelf components. It consists of hardware and software components from different manufacturers. Though this approach is cost effective, it has also presented a significant set of challenges in integrating the various components. So far, Ayushman supports two medical monitoring devices (EKG monitors, blood pressure monitors), which collect health data.

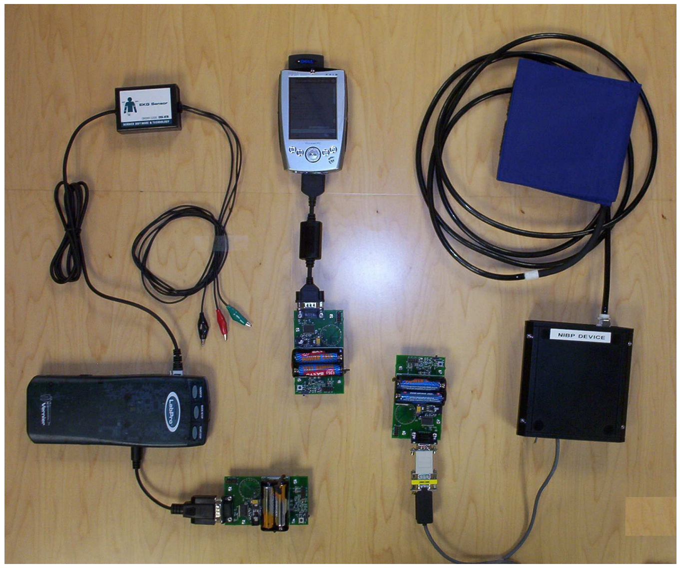

Figure 3. **Ayushman Testbed Implementation**

We did an an extensive yet unsuccessful survey of the market to find medical devices which were programmable and which could communicate using wireless channels. We therefore decided to buy medical devices which communicated using a serial interface and use Berkeley Mica2 motes [1]running TinyOS [2]to control them and to provide wireless communication. The EKG device and BP(blood pressure) monitor both are controlled using Mica2 motes through serial cables.

In our architecture, we have implemented the local gateway on a Dell Axim PDA running Windows CE using Java. The PDA has a Java runtime environment called J9, developed by IBM [3]. This Java runtime environment was a bit unstable and frequently crashed the PDA, even without any problems with the code. This problem has still not been completely resolved. The PDA communicates using an 802.11b card and cannot communicate directly with the mote connected to the medical device, therefore we use a Mica2 mote connected to the PDA acting as a bridge between them. The data collected by the PDA is sent over an 802.11b channel to a laptop, which run the external gateway software. The wireless connection within our building is a bit unstable, resulting in frequent disconnections between the PDA and the laptop (the local gateway and the external gateway). This resulted is some loss of data at the external gateway, which is usually not acceptable. Therefore, we are currently working on quantifying the disconnection rate and the corresponding data loss

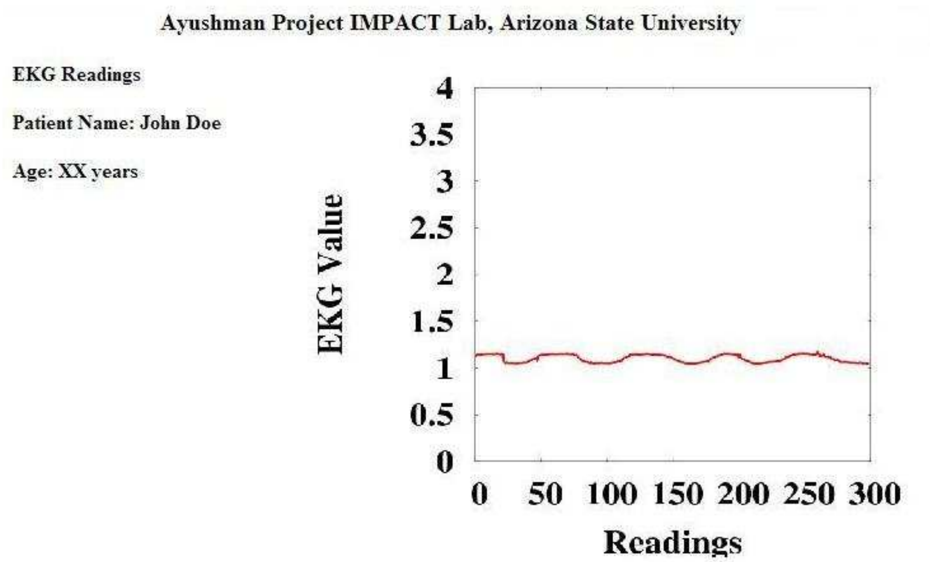

Figure 4. **EKG Readings Displayed to a Remote User**

and also on developing a synchronization protocol which will prevent any data loss during such disconnections.

Once the communication and storage infrastructure were set up, we faced an additional problem in terms of reliable communication. We use Mica2 motes for collecting and transmitting data from health monitoring devices. These motes provide 19.2kbps (2400bytes/sec) of channel bandwidth. The EKG device, which collects and transmits data continuously, uses about 315 bytes/sec of the bandwidth. At this rate, we can connect about 6 devices continuously transmitting data. However, we were facing many challenges with reliable communication within the infrastructure particularly high packet loss. We decided against maintaining reliability by retransmission because of the real-time needs of the medical data being transmitted, and we are currently researching efficient FEC (Forward Error Correction) schemes for improving communication reliability. Additional problems also resulted due to lack of synchronization between devices and a limited bandwidth availability when multihop communication is used within the sensor network to send data to the local gateway. Figure 3 shows part of our prototype, with the medical devices communicating with the PDA using motes as intermediaries (On the left is the EKG monitor, at center is the local gateway (PDA), and on the right is the BP monitor). The readings are sent to a laptop (not pictured) acting as the external gateway, which collects data from multiple PDAs and uses a desktop PC to publish the data over the Internet. Figure 4 shows the EKG data collected through Ayushman as seen through a browser by a remote user. The EKG curve is slightly blunt because of the high packet loss, a problem which we are currently addressing.

5. Conclusion and Future Work

In this paper, we presented our work on developing Ayushman, a health monitoring infrastructure and testbed. It provides a reliable, secure, and real-time patient monitoring system, able to function in diverse scenarios and in a realistic environment, for testing communication protocols. It has a hierarchical, cluster-based architecture, which ensures a highly dependable and efficient system. It is being built using off-the-shelf components, which provides many challenges in system integration and operation.

So far, we have completed the development of the communication, query, and data storage services. Currently, we are working on the resource management, reliability, and synchronization issues, following which we will implement the security and data aggregation primitives.

Notes

1. IMPACT Lab - http://shamir.eas.asu.edu/∼mcn/

References

[1] Crossbow Technologies Inc.: http://www.xbow.com

[2] TinyOS - Open Source OS for Embedded Sensor Networks: http://www.tinyos.net

[3] IBM's Websphere J9 Virtual Machine: http://www-306.ibm.com/software/wireless/wsdd

[4] MobiHealth Project http://www.mobihealth.org

[5] Summary of HIPAA Health Insurance Probability and Accountability Act US Department of Health and Human Service (2003)

[6] Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing IEEE Trans on Dependable and Secure Computing **1(1)** (2004) 11-33

[7] Gagliano, D. and Xiao, Y.: Mobile Telemedicine Testbed American Medical Informatics Association Symposium (1999)

[8] Lorincz, K. et al: Sensor Networks for Emergency Response: Challenges and Opportunities IEEE Pervasive Comp. **3(4)** (2004) 16-23

[9] Shimizu, K.: Telemedicine by Mobile Communication J. Engg. in Med. & Bio. **18(4)** (1999) 32-44

[10] Shneidman, J. et al.: Hourglass: An Infrastructure for Connecting Sensor Networks and Applications Harvard Technical Report TR-21-04 (2004)