

SENSOR NETWORK TEST-BED FOR ENVIRONMENTAL MONITORING

Aung Htein Maw, Zaw Tun and Khin Thanda Soe

University of Computer Studies, Yangon, Myanmar
{ahmaw73, zawtun78, ktdsoe}@gmail.com

ABSTRACT:

Sensor networks consist of embedded devices deployed in different types of environments in order to sense its surroundings. Disaster management and rehabilitation is very important for all human beings. Environmental monitoring is the most important issue for everyday emergency situations and a significant driver for wireless sensor network research. Recent advances in electronics and communication technologies enable the devices to produce micro-sensors capable of performing a variety of different environmental applications. An implementation of a wireless sensor network test-bed is presented in this paper. This test-bed is a reactive network for environmental monitoring and evaluates the effectiveness of the network for data gathering using both laboratory and field tests. This sensor network test-bed monitors the environment: temperature and humidity in real-time. This paper deals with a test-bed that consists of a personal computer (PC) where an end-user can communicate with the sensor nodes through a base station (BS), which is implemented by combining HyBus kit (Hybus kit2420).

KEY WORDS: Sensor Networks, Environmental Monitoring, Micro-sensors, Test-bed

1. INTRODUCTION

Advances in processor, memory and radio technology have made it possible to build small and cheap devices that are capable of sensing and altering their environment, communicating with nearby devices using short-range wireless channels, storing a small amount of state, and performing a limited amount of computation. We will be able to deploy thousands to millions of these devices to perform a wide range of applications such as monitoring poorly accessible or dangerous environments like disaster areas, hostile territories and active nuclear fields. Hostile or dangerous terrains make it impossible to build fixed infrastructures of powerful and expensive hosts. The individual node monitors its environment and collects data to the end-user.

Wireless sensor networks are a collection of sensor nodes each having the capability of sensing, processing, and communicating, which will lead to thousands of potential applications ranging from biomedical to health, military, habitat, environmental, industrial, traffic control, and many others. Most of the sensor network applications are designed for military research (Akyildiz and et.al, 2002). Nowadays some civilian applications are using sensor network designed to perform information processing tasks in other commercial area, space exploration, chemical processing, disaster relief, etc.

Environmental monitoring is a significant driver for wireless sensor network research, promising dynamic, real-time data about monitored variables of a landscape and so enabling scientists to measure properties that have not been observable.

This paper describes the design and implementation of a reactive network for environmental monitoring and evaluates the effectiveness of the network for data gathering using both laboratory and field tests. The system investigates sensor network test-bed for environmental monitoring by combining HyBus kit (Hybus kit2420) in wireless sensor network lab at University of Computer Studies, Yangon (UCSY).

The remainder of this paper is organized as follows: Section 2 presents some sensor network test-bed and early warning system in brief. Section 3 provides communication components on HyBus motes. Hardware verification of HyBus motes are presented in section 4. Section 5 discusses experimental setup and evaluation of our test-bed. Finally, we conclude the paper in section 6 for future disaster management applications.

2. BACKGROUND

Several wireless sensor networks test-beds and early warning system were designed and implemented. We will briefly describe some of them.

TrueMobile (Johnson and et.al, 2005) is a test-bed for mobile wireless sensors, based on Emulab consisting of Garcia robots that carry 900 MHz mica2 motes and XScale-based Startgate small computer system running Linux through a fixed field of sensor-equipped motes. It provides mobility effects on sensor-driven applications.

Mirage (Chun and et.al, 2005) is a test-bed for a resource allocation system that uses combinational action to manage the test-bed resources.

Motelab (Werner and et.al, 2005), open source software installed at MIT, Berkeley, and Harvard,

consists of multiple Ethernet connected sensor network nodes linked to a central server where users employ a reservation-based scheduling system with quotas. It uses a per-experiment MySQL database to store all information needed for test-bed operation and provides two different ways to the user.

EmStar (Girod and et.al, 2004) test-bed aims to deploy the EmStar software that integrates high capability nodes, iPAQ or Crossbow Stargate platforms, to develop and deploy more complex applications.

Scale (Cerpa and et.al, 2003) is built on top of the EmStar programming model to measure the link quality of the low radio channels and displays this data to users through a web interface.

The FloodNet (Zhou and et.al, 2006) project at University of Southampton investigated the use of pervasive computing and Grid computing to provide early warning of possible flood. By deploying wireless sensor nodes on the floodplain, FloodNet used to make flood predictions to improve warning times. The information collected about water level is fed through a gateway. The incoming data will be further utilized by the predictor.

3. COMMUNICATION SYSTEM COMPONENTS

The HyBus communication system can be divided into two components: sensor hardware and HyBus software environment.

In order to understand the capabilities and limitations of the HyBus motes (Hmote), we need to know the specifications of sensors (HyBus). The HyBus motes (Hmote) come in two parts: a processor/radio board and add-on sensor boards.

Hmote: A new platform for low power research in monitoring applications with long lifetime, low power and low cost. MSP430 microprocessor and 512KB of EEPROM is used. It operates 915 MHz and Chipcon CC2420 RF transceiver with a maximum data rate of 250kbps is embedded on board during data transmission. The interface is used as programming through USB port for sensor motes and powered by 2 AA batteries. The sensor board, used in this test-bed, is the MSP430 that offers four basic environmental sensors: light, temperature, humidity and sound in accessible environment. Figure 1 shows the Hmote and H-sensor.



Figure 1. Hmote and H-sensor of Hybus Kit

Hmote also contains an expansion connector that handles different sensor boards and runs the TinyOS multithreading event-base operating system.

TinyOS: TinyOS is an open source software platform developed by researchers at UC Berkeley and actively supported by a large community of users. TinyOS is component-based operating system and platform targeting wireless sensor networks (WSNs). The key to TinyOS functionality is the nesC (network-embedded-system-C) compiler, which is used to compile TinyOS programs.

The TinyOS applications, library, and system are written in nesC, extension of C programming language. Every program in nesC is built out of components, which are wired using interfaces to form whole programs. The interfaces in NesC are bidirectional; it specifies a set of named functions, called commands, to be implemented by the interface's provider and a set of named functions, called events, to be implemented by the interface's user (Philip ,2006). NesC defines a concurrency model, represented by tasks and events. Tasks are not time critical and run to completion following the FIFO scheduling with respect to other tasks which implies that a single stack is only needed.

4. HARDWARE VERIFICATION OF HYBUS MOTES

In this section, we discuss the hardware verification of Hybus kit for radio communication (RF) between two motes.

The first step for verifying the Hmote is test with a simple Blink application. Blink is a basic application that starts a 1Hz timer and toggles the red LED every time it fires. It is a very simple program that is little more than a demonstration of TinyOS programming.

The second step is radio communication (RF) between two motes. The two applications used here: *CntToLedsAndRfm* and *RfmToLeds*. *CntToLedsAndRfm* maintains a counter on a 4Hz timer; it displays the lowest three bits of the counter value on its LEDs. The red LED is the least significant of the bits, while the yellow is the most significant. It also sends out each counter value in an *IntMsg (AM)* packet. This will be programmed on mote ONE. *RfmToLeds* listens for messages of type *IntMsg (AM)*. When it receives one, it sets its LEDs to the lowest three bits of the received value combined with *CntToLedsAndRfm*. This will be programmed in mote TWO. Then the mote ONE is connected to the Interface Board to load *CntToLedsAndRfm*. The LEDs on the second mote should start blinking now. When the first mote is turned off, the blinking of LEDs on the second will stop too. This indicates that the radio communication is working

fine. Figure 2 shows the detail of radio communication between two motes.

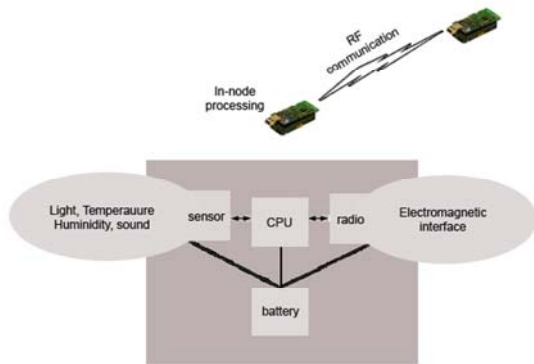


Figure 2. RF Communication between two motes

5. EXPERIMENTAL SETUP AND EVALUATION

In this section, we present experimental setup and evaluation of the proposed test-bed.

5.1 Experimental Setup

Our test-bed consists of a personal computer (PC) acting as an end-user of 3.4 GHz Pentium 4, 512 MB RAM, running the WindowXP operating system, IBM JDK 1.4.2 and TinyOS was utilized. One Hmote acted as the base station (BS) and was connected to the computer via USB port and the two active H-sensor nodes as sensing devices. Figure 3 shows the overview of the test-bed. The test-bed shows the two nodes sense temperature and humidity of its environment from about 3 m from base station. The two nodes are separated in 4 m distance in the test environment. The values sense from each node can be seen on PC by an end-user.

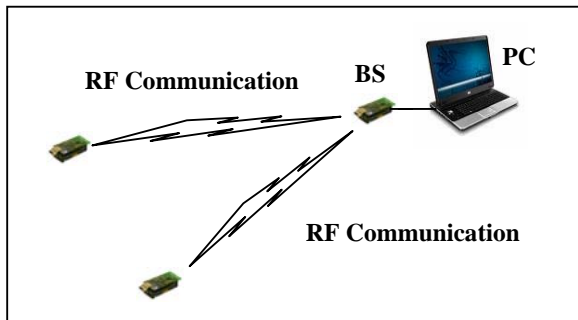


Figure 3. Overview of Test-bed

5.2 Evaluation

Hsensor nodes are installed with individual programs and deployed in the testing area. Each node then individually monitors its environment and sends the acquired data over a reliable communication network to

the base station. Each node directly communicates with the base station via a single-hop network and the base station has unlimited battery power, performs the additional processing and reports the temperature and humidity of testing area to the PC.

Figure 4 shows the sensing values of each node. The program is simply printing the raw data of each packet received from the serial port.

```
7e 00 0a 7d 1a 01 00 0a 00 01 00 46 03
8e 03 96 03 96 03 96 03 97 03 97 03 97
03 97 03 97 03
```

Figure 4. Sensing Data packet of node

Each data packet that comes out of the mote contains several fields of data. Some of these fields are generic Active Message fields, and the overall message format is as follows:

- Destination address (2 bytes)
- Active Message handler ID (1 byte)
- Group ID (1 byte)
- Message length (1 byte)
- Payload (up to 29 bytes):
 - source mote ID (2 bytes)
 - sample counter (2 bytes)
 - ADC channel (2 bytes)
 - ADC data readings (10 readings of 2 bytes each)

Then based upon the raw data we make processing of raw data. A custom graphical user interface, The *SerialForwarder* tool reads packet data from a serial port and forwards it over the wireless connection. This class processes sensor data from programmed H-sensor nodes via a base station. The java applet provides a graphical view of the logical network topology.

It also permits variation of the sample rates sending pre-defined commands to the surge nodes. The *Listen* program is the most basic way of communicating with the mote; it directly opens the serial port and just dumps packets to the screen. Obviously it is not easy to visualize the sensor data using this program. This will open up a GUI window that looks similar to the following figure 5.

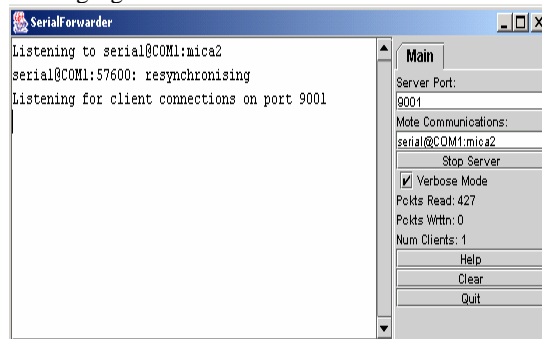


Figure 5. GUI for SerialForwarder Listen tool

The test-bed provides the following information to users:

- Detects the existence of all the motes in a wireless network.
- Displays mote information, including the mote identification number (ID), the number of messages sent from each mote, etc
- Displays the topology of network

Figure 6 shows the real-time experimental results of our test-bed using Surge Demo. The node 0 is base station, the rest are different motes. When a mote sends a message, Surge displays two concentric circles around the node. The color bars underneath each node display the Quality and Yield associated with the node. "Quality" is a measure of how good the link to a neighboring node is and takes values from 0 to 1. The color of the edge is the same as the color of Quality bar. The color bar labelled "Yield" is a measure of the rate of message sending from a mote as a fraction of the expected number of the sensor messages. In this results node 2 transmits its environment data to the base station node 0. Hence we can see changing of values in reactive manner in this test-bed.

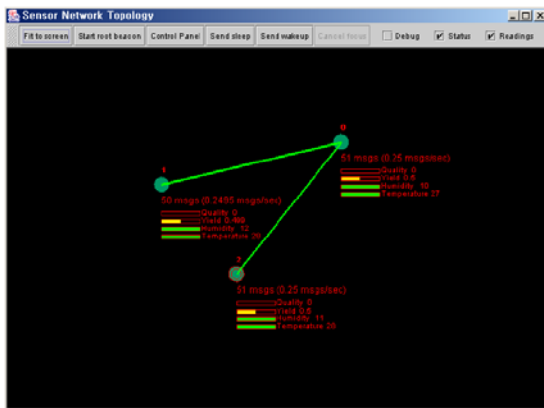


Figure 6. Experimental result of test-bed

6. CONCLUSION AND FUTURE WORK

Environmental monitoring is a significant driver for wireless sensor network research, promising dynamic, real-time data about monitored variables of a landscape and so enabling scientists to measure properties that have not been observable. In this paper we developed the design and implementation of a reactive network for environmental monitoring and evaluates the effectiveness of the network for data gathering using both laboratory and field tests. We investigated sensor network test-bed for environmental monitoring by combining HyBus kit (Hybus kit2420). Disaster management and rehabilitation is very important for all human beings. This test-bed can be further developed using more nodes and multi-hop communication through internet for early warning systems. The future trends of wireless sensor networks is really the vision of "anytime,

anywhere" communications for disaster management and rehabilitation process.

REFERENCES

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002, A Survey on Sensor Networks. IEEE Communications Magazine, Vol.40, Issue.8, pp.102-114.
- Cerpa, A., Busek, N., Estrin, D., 2003. Scale: A tool for Simple Connectivity Assessment in Lossy Environments. Center for Embedded Networked Sensing, University of California, LA, Tech. Rep.CENS Technical Report 0021.
- Chun,B. N., Buonadonna, P., AuYoung, A., Parks, C., Shneidman, J., Snoeren, A.C., Vahdat, A., 2005, Mirage: A Microeconomic Resource Allocation System for SensorNet Testbeds. 2nd IEEE Workshop on Embedded Networked Sensors, Sydney, Australia.
- Gay, D., Levis, P., Culler, D., Brewer, E., 2003. nesC 1.1 Language Reference Manual.
- Girod, L., Elson, J., Cerpa, A., Stathopolous, T., Ramanathan, N., Estrin, D., 2004. EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks. In Proceedings of the UNENIX Technical Conference, Boston.
- Johnson, D., Stack, T., Fish, R., Flickinger, D., Ricci, R., Lepreau, J., 2005, TrueMobile: A Mobile Robotic Wireless and Sensor Network Testbed. Flux Technical Note FTN-2005-02, University of Utah.
- Philip, L., 2006. TinyOS Programming.
- Werner, G., Swieskowski, P., Welsh, M., 2005, Motelab: a wireless sensor network testbed. Information Processing in Sensor Networks, 4th International Symposium, pg. 483-488.
- Zhou, J., De Roure, D., 2006 Designing Energy-Aware Adaptive Routing for Wireless Sensor Networks. In the 6th International Conference on ITS Telecommunication, pg. 21-23, Chengdu, China.
- HyBus., Wireless Sensor Network Hmote-Ubiquitous System. <http://www.hybus.net>