Sensor Network with User Supplied Connectivity

Capstone Project Report

Student Name: Sowoon Pyo

Project Type:
Project: TCSS 702

Committee Names:
Committee Chair: George Mobus, Ph.D.
Committee Member: Sam Chung, Ph.D.
Committee Member: Akur Teredesai, Ph.D.

Submission Date: 5/24/2007

Abstract

This project develops a proof-of-concept demonstration of a unique approach to collecting event-oriented data from distributed smart sensors in an urban setting. It is called a *Sensor Network with User-supplied Connectivity* (SNUC). It allows connections between heterogeneous sensor networks and a service-oriented distributed computing infrastructure via hand-held users' connectivity to a mobile network. There are many cases where sensor networks need to be deployed in an environment not suitable for wired communication. While connecting the sensor network to the computing infrastructure via wires is not an option, the sensor data must still be transferred to the computing infrastructure so that the backend servers can process the data and take appropriate actions. This project demonstrates that the right connectivity in many scenarios can be provided directly by the user carrying a handheld device that can bridge the gap between the sensors' Personal Area Network and the computing infrastructure's Local Area Network through Wide Area Network such as cellular network. In cases where the data from the sensors does not need to be collected in real time, this solution provides a cost and complexity reduction and provides the sensor data and computations at the moment when the user is physically located near the sensors and able to take action if necessary. And because the data is collected on demand, the power usage of the sensor network can also be reduced.

1. Introduction

The wireless sensor networks enable diverse applications that range from simple home monitoring systems to deeply scientific natural phenomena tracking systems. Sensor devices are designed to carry out a relatively small set of operations such as detecting a specific status or event and transmitting the data. Even though a sensor alone cannot achieve a complex operation, when these limited capability devices are connected together, they become a powerful source of data. Moreover, cooperation of Wireless Personal Area Network (WPAN) and Local Area Network (LAN) connectivity can augment the computational capability of the sensor network by exposing their functionality to the Internet.

Much research has been conducted to overcome a sensor device's physical limitation of size and power consumption issue by networking those efficiently using highly optimized protocols. However, this issue becomes more prominent when the scale of network becomes larger. Also, with development of emerging radio technology, these wireless sensor networks become more and more heterogeneous and broaden their functional fields by integrating themselves into our lives. However, there is no common framework that incorporates those heterogeneous networks. It is mainly because each sensor network is discrete such that there is no information sharing between them. In order to solve the scalability issue and interoperability issues described above, we need a component, which is common in the field, that connects small-scale sensor networks efficiently.

Mobile phones have long-range connectivity through general packet radio service (GPRS) or 1x (single-carrier) Radio Transmission Technology (1xRTT) through the mobile operator's service. This connectivity can fill the gap between small sensor networks to the target computing infrastructure thus removing needs of complex protocols, routing, and even extra nodes that only exist as hopping route. Also, the distribution of mobile phones has dramatically increased in the last decade. The proliferation of feature rich mobile phones can provide not only long range connectivity but also short range connectivity such as Bluetooth which enables connection to heterogeneous sensor networks.

This project defines a Sensor Network with User-supplied Connectivity (SNUC), which connects heterogeneous sensor networks to the service oriented distributed computing infrastructure via hand-held users' connectivity to the mobile network. First, this project introduces the small-scale sensor network application scenario that solves current issues by incorporating hand-held device's connectivity. Second, the general sensor network architecture, the properties of network abstraction layer between each node, and communication scheme that enable connection between sensor networks to the computing infrastructure are investigated. Third, the implementation details of software components in each node are introduced and the prototype of SNUC is presented. Fourth, the vision of this project and future work items are discussed.

2. Related Works

RY Fu. et al. [RYF:Afr] presented a Device Capability-On-Demand (DCOD) system framework that introduces a new concept - virtual device. A virtual device consists of one physical handheld device such as a Personal Digital Assistant (PDA) that dynamically reaches out to various devices around it and associates with them in order to overcome its own limited capability and thus provide the user with a richer device experience. In addition, they present a Virtual Device Service Gateway (VDSG) as core to the DCOD framework. The gateway architecture provides peer-to-peer networking of various devices such as computers, audio equipment, projectors, and phones. SNUC shares the same peer-to-peer networking concept among devices as exists in the DCOD framework. SNUC also provides a similar gateway and mobile handheld device connection architecture. In the case of SNUC, however, the devices are small sensor networks and the goal is to use the data collected by the sensor networks.

Frank Siegemund et al. [FRA:The] discussed that the computational capabilities of smart objects – every day objects augmented with small sensor-based computing platform- are very limited. Thus they argue that most of these limitations can be overcome if smart objects can spontaneously access the capabilities of nearby handheld devices. They identify and illustrate six different means by which computer-augmented everyday artifacts can make use of handhelds:1) as a mobile infrastructure access point; 2) as a user interface; 3) as a remote sensor; 4) as a mobile storage medium, 5) as a remote resource provider; and 6) as a weak user identifier. In core, using handhelds to augment a smart object environment is same but the approach of this paper is to focus on the data distribution framework using nearby handheld devices.

Behcet Sarikaya [SAR:Nom] introduced a novel mobile wireless sensor network architecture: nomadic user based sensor network architecture. In this new architecture, the wireless sensor network reacts to the event that is initiated by nomadic users. Event-based deployments are cost-efficient and do not require a dense sensor node population. A peer-to-peer networking approach is needed in order to communicate with sparsely populated sensor nodes in order to satisfy the nomadic user's needs. This nomadic user approach concept to build the event-triggered small-scale sensor network is directly used in this project and further extended to a larger scale framework that connects many small-scale sensor networks to the computing infrastructure to enable service oriented data distribution.

Another approach to connect the small-scale sensor network to the mobile network has been introduced by Srdjan Krco et. al. [SRD:Ena]. The authors proposed the architecture of the sensor network gateway that interacts with users on behalf of sensor networks and provide attributes based on access and querying. The connection has been made through a modified JXTA (Juxtapose) peer-to-peer networking.

3. Objectives

The purpose of this project is to present a distributed network framework that provides service to users by connecting sensor networks to traditional computing infrastructure by using the mobile network.

Hand-held devices with the 2nd Geenration (2G) or the 3rd Generation (3G) capabilities have enabled continuous communication over mobile network to Internet [SRD:Ena]. The 3G mobile communication system has been brought into service and it supports up to 1920 Kbit/s data transfer rate. As Moore's Law also applied to hand-held devices, they have become more powerful and feature rich. Current mobile phone technology may still have a long way to go to achieve a single device model [RYF:Afr], but it has nearly enabled the ubiquitous communication aspect of any time, any where.

The main characteristics of the sensor nodes are the resource and size constraints. They have to keep their power usage fairly low so that they can have a long lifetime without changing the battery and their deployment has to be unobtrusive. The issues come from connecting sensors, self-organization, and data aggregation [SRD:Ena]. However, within small areas, most of these issues found in large area sensor networks become trivial issues. The connection between sensors are relatively simple, maybe just one hop is enough to reach the gateway node from sensor nodes. In ubiquitous computing, the small-scale sensor network is more realistic in real world application usage.

When dealing with heterogeneous small sensor networks connected by a mobile network, the event-based architecture is useful [SRA:Nom] in some scenarios. The traditional sensor networks are described as *directed-diffusion* systems that are deployed in patches and they are connected to the main gateway node. By triggering the sensor networks whenever needed, the resource requirements can be significantly reduced.

As small-scale networks are connected together, it enables the possibility of new generation applications, but interoperability between sensor networks becomes an issue. As part of an effort to standardize the protocols used in sensor networks, IEEE 802.15.4 has been established as a specification of the RF channel and signaling protocol to be used [JAS:The]. IEEE 802.15.4 task group produced ZigBee [ZigBee], a high level communication protocol based on the IEEE 802.15.4.

3.1 SNUC application scenario

There are many cases where sensors networks need to be deployed in an environment not suitable for wired communication. This project considers an example application scenario of a network of humidity, temperature and other sensors scattered throughout a large building. The sensor network is responsible for measuring a variety of environmental factors such as corrosive effects, air pressure, etc at each wall on each floor. Connecting the sensor network to the computing infrastructure via wires is not an option due to the costs of running and routing the wiring and the potential for accidental wire damage after installation.

The sensor data must be transferred to the computing infrastructure so that the backend servers can process the data and take appropriate actions. This project proposes that the right connectivity can be provided directly by a service technician carrying a PDA. As the technician walks through the various floors of the building, the PDA queries the sensors via a Personal Area Network (PAN) wireless protocol and passes along the data to the building's computing infrastructure via a Wide Area Network (WAN) wireless protocol.

In this scenario, the PDA provides the critical link between the sensor network and the computing infrastructure. As a valuable side effect, the technician is also strategically positioned to act on many decisions made by the backend servers in real time, such as replacing low batteries, diagnosing faults in the system or investigating environmental anomalies.

3.2 SNUC architecture

A. Layered architecture of SNUC

SNUC consists of sensor nodes and gateway nodes as sensing units and hand-held devices such as mobile phone as a communication unit. Figure 1 shows the layered architecture of SNUC. Each cluster of sensor networks consists of one or more sensor nodes, and gateway nodes. The gateway node queries sensor nodes to acquire data when an event occurs such as a timer event or user command. In this project, the sensor node streams data out based on timer events once the gateway connection is made. The nearby handheld device, which is equipped with WAN as well as PAN capability, makes a connection with the gateway node using short range radio and sends the data over cellular network.

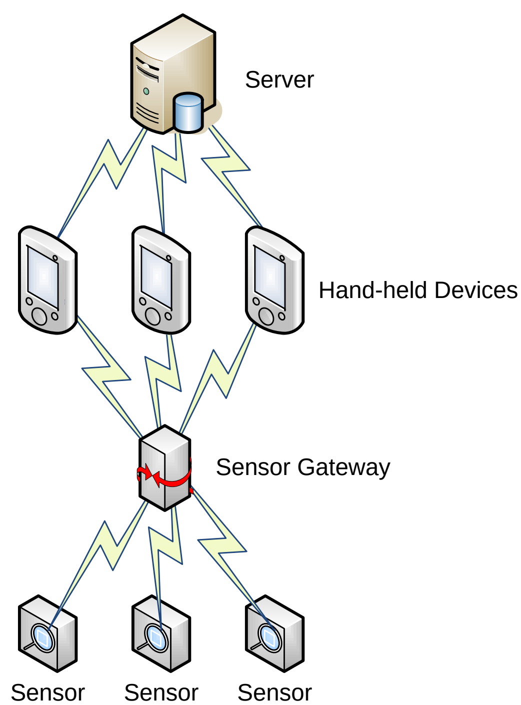


Figure 1. Layered hand-held device enabled sensor network

The data from each sensor cluster is transferred to the traditional computing infrastructure for use. The hand-held devices send the data through standard Hyper-Text transfer Protocol (HTTP) protocol by using a Web Service to the backend computing infrastructure such as database/web servers (SNUC service center). The Windows Mobile 5.0/6.0 operating system supports web programming interfaces to send the data through HTTP protocol. Once the data is delivered to the service center, many applications can be created. Figure 2 shows the clusters of sensor networks connected to nearby hand-held devices through PAN and then the hand-held devices are connected to the SNUC service center through the WAN.

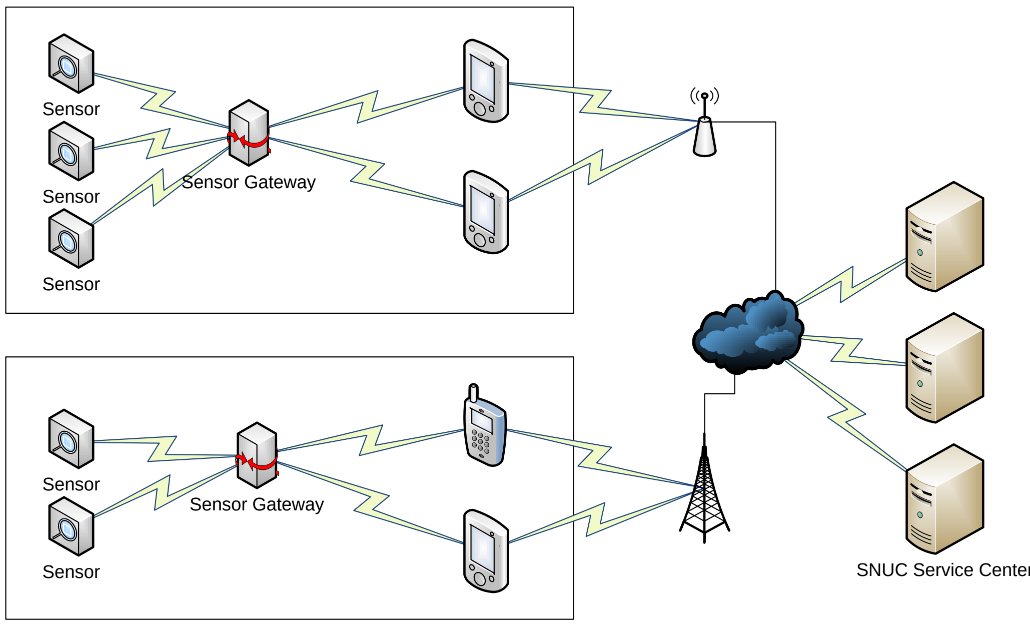


Figure 2. Sensor network clusters and web/database servers

Unlike a large-scale wireless sensor network, each sensor node can be autonomous and may or may not communicate with each other depending on application requirement. Handheld device nodes generate events to signal the nearby gateway node and the gateway pushes data upon successful connection establishment. Although this event driven architecture can also be applied to the typical large-scale sensor network such as the habitat monitoring, the main benefit of SNUC framework is the synergy created by the connection between mobile network and small-scale autonomous sensor networks.

B. Radio technology used in SNUC connections

The general rule of radio technology is that the more coverage, the less throughput and the more throughput, the more power. Each layer of the SNUC architecture is connected using various radio technologies depending on the characteristics of the layer. Figure 3 shows the diagram of each connection between each layer. A 2.4 GHz wireless radio (Chipcon CC2420 module) is used for Sensor/Gateway connection. This radio provides up to 250 Kbps throughput and may attain 50 meter range indoors and upwards of 125 meter range outdoors. The same radio would be ideal for the gateway and the handheld node connection but since there are not many mobile phones equipped with the IEEE 802.4.15 radio and cellular network together, the Bluetooth radio is used. Bluetooth provides up to 10 meter coverage and 1 Mbps throughput. This project uses wide area cellular telephone network for Mobile phone/Web Server connection which is available anywhere the cellular transmitter is available by mobile operators.

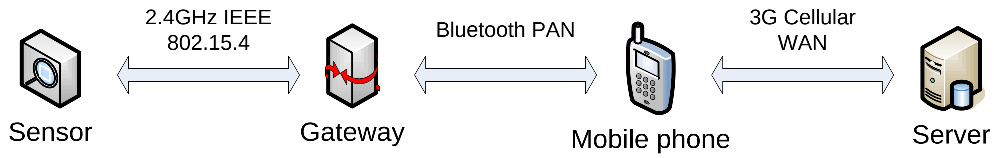


Figure 3. Radio technologies connecting each layer in SNUC

C. Communication

In the SNUC project, a patch of small-scale wireless sensor networks is composed to stream data from the sensor node to the gateway node. Sensors send out raw data packets by using a short range but power efficient radio. A handheld device in the vicinity of the sensor network triggers an event to make a connection to the gateway unit of the small sensor network. Then, the gateway pushes out the sensor node data in eXtensible Markup Language (XML) format to the handheld device by using Bluetooth Personal Area Network. A user can review the data showing on the handheld device and perform the necessary action. For example, a building maintenance engineer would further diagnose corrosion of pipes if the humidity level were higher than expected. The data transmitted to the handheld device can now be transmitted to the database server so that information can be shared among many users. Once the data is stored on the database server, the data can be processed by more complicated equipment than handheld devices and shared to multiple users in the type of service through Internet. Figure 4 shows the communication flow from the sensor network to the SNUC service center.

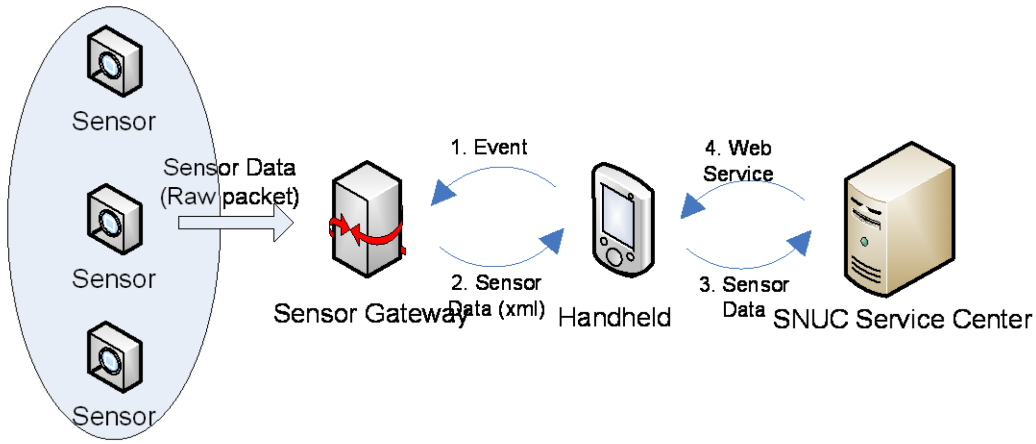


Figure 4. Communication architecture.

4. Software component implementation details and results

Figure 5 shows key components of each node in SNUC, sensor node, gateway node, the handheld node and SNUC center, the traditional computing infrastructure.

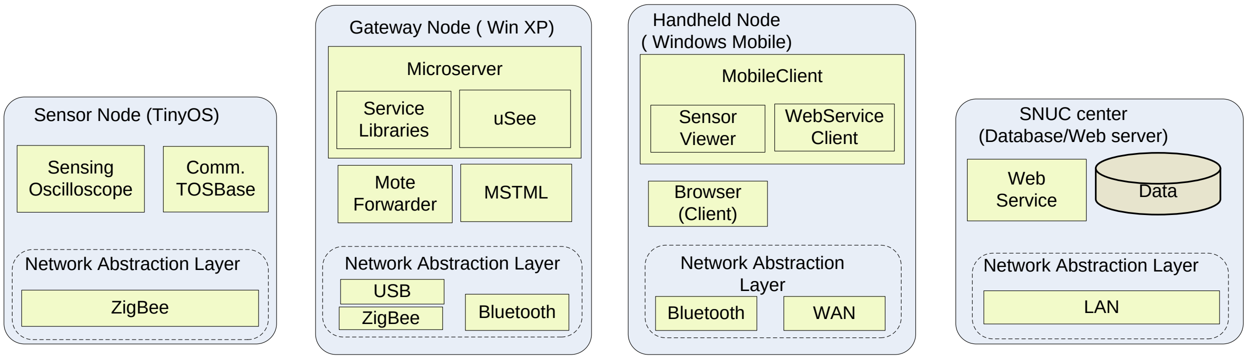
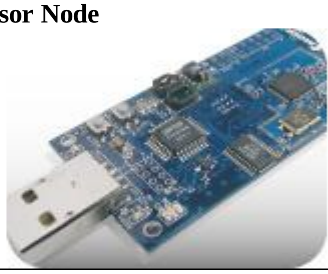


Figure 5. Distributed component view of SNUC system

4.1 Node Specifications and Development Environment

Table1. shows system specification and the development environment of each node in SNUC system.

Table 1. Node specification and Development environment

Node	**Operating System**	**System Specification**	**Development Environment**
Sensor Node 	TinyOS	CPU : MSP 430 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Tranceiver RAM : 10 KB Flash : 48 KB External Flash : 1 MB	Boomerang (Mote-iv's TinyOS Open source distribution
Gateway Node	Windows XP	CPU : Intel T2500 RAM : 2 GB	* MSR Networked Embedded Sensing Toolkit V.0.2 Alpha * Visual Studio 2005 * Microsoft .NET framework * Microsoft SQL Server & Management Studio Express

			
Handheld Node 	Windows Mobile Classic 6.0	CPU : TI OMAP 1710 RAM : 64 MB Flash : 128 MB Radio : Bluetooth, Cellular	* Windows Mobile 6.0 Classic SDK * Visual Studio 5.0 * Microsoft .NET Compact framework
Web Server 	Windows Server 2003 Microsoft SQL Server		* Senseweb project web service * SensorMap web site

4.2 Sensor node

Sensor node implementation used Boomerang [Boomer], Moteiv's TinyOS open source distribution. NESC [Nesc] is a language that is used in TinyOS applications which is highly optimized for resource constraints characteristic of sensor nodes. There are two kinds of applications running on sensor nodes. One is the sensing node that reads all the sensor value on the Tmote sensor and communicates over the radio and the other is the base station sensor node that maintains queues for receiver and sender and interfaces to the gateway node. This project modified the Oscilloscope application from Boomerang to adjust frequency and remove unnecessary data channels to reduce radio communication and processing load of the gateway node.

4.3 Gateway node

Gateway node is implemented by using Microsoft MSRSense kit 1.4 [MSRSen]. MSRSense is a collection of software running on PC that enables processing, collection, and visualization of the data. The software components running on the gateway node are a mote forwarder, a

MicroServer execution environment, and a set of libraries which process specific tasks. In this project, because the key feature is the ability to connect patches of wireless sensor networks by using a device with WAN capability, the assumption is that the sensor network is small and sensors are connected to a gateway. The gateway node receives data from a base sensor which communicates data with other sensors that are physically connected to a Universal Serial Bus (USB) port of the gateway. The gateway node requires the most complex network abstraction layer for interfacing sensor networks and handheld devices. Though most handheld devices support USB connection, to keep this system wireless, Bluetooth radio is used for connection with the handheld device. In summary, the base mote device receives data through 2.4 GHz radio, sends data over USB port, and gateway software outputs data through Bluetooth radio.

1) Moteforwarder
   Moteforwarder a tunnel component that is responsible for transferring data from the base mote on the serial port. Moteforwarder detects the device on the port, makes a socket server in the localhost while receiving data and waits for the connection from the client which is MicroServer in this system. Once the client connection is made, the tunnel is fully connected and transferring the data.

2) MicroServer
   MicroServer is the core component of gateway. MicroServer consists of two major components, Miusee and Service Libraries. Miusee is a runtime application that provides a framework to compose tasks, called services. Miusee is given a user specified tasks through MicroServer Tasking Markup Language (MSTML) and composes services described in MSTML from the service libraries. Meanwhile, MicroServer connects itself to the Moteforwarder port to register as a client and receives the data. The received data are passed from one service to the next service for processing. This project added a service library that provides an interface to connect the handheld devices nearby. Once MicroServer establishes connection with the handheld device, it starts to stream the data to the handheld device through the socket server. In summary, the raw data from the mote node are transferred to the MicroServer and processed to XML tokens, then transferred to the handheld device.

3) MSTML (MicroServer Tasking Markup Language)
   MSTML is a modified Modeling Markup Language (MoML) that is used for defining service composition of the gateway node. In this project I used the below MSTML to compose a series of 3 services, TOSReceiver, ToXML, and ToMobile. TOSReceiver service processes raw data packets to defined form, ToXML service changes the packet to XML format, and ToMobile creates interface to the handheld node and push out the data to the interface.

```
<?xml version="1.0" standalone="yes"?>
<entity name="OscilloscopeApp" >
  <port name="mote" type="AMHandler">
    <property name="input"/>
    <property address="-1:10"/>
  </port>
  <port name="port1" type="Socket">
    <property name="output"/>
    <property address="localhost:5000"/>
  </port>
<entity name="TOSReceiver" type="ComplexTOSPacketReceiver">
<property name="messageType" value="ArrayOscopeMsg"/>
</entity>
<entity name="ToXML" type="DataToXML">
</entity>
<entity name="ToMobile" type="DataToMobile">
  <property name="hostip" value="192.168.0.119"/>
</entity>
  <relation name="relation1"/>
  <relation name="relation2"/>
  <relation name="relation3"/>
  <link port="mote" relation="relation1"/>
  <link port="TOSReceiver.input" relation="relation1"/>
  <link port="TOSReceiver.output" relation="relation2"/>
  <link port="ToXML.input" relation="relation2"/>
  <link port="ToXML.output" relation="relation3"/>
  <link port="ToMobile.input" relation="relation3"/>
</entity>
```

4.4 Handheld node

The handheld device node is a core component of the SNUC system that provides long range connectivity which ultimately enables information sharing among users and other heterogeneous sensor networks. In this project, Microsoft® Windows Mobile® 6.0 [WinMob] device is used for the implementation. This particular Windows Mobile 6.0 device, Blackjack3 (i607) available through the mobile operator AT&T in the US, supports 3G voice and data network and Bluetooth for personal area network through Internet Sharing feature available in the Windows Mobile OS. In this handheld node, MobileClient application provides a way to see the sensor data visually, and send the data to database server through Web Service interface. Once the data is published to the Web Service, the data can be processed by more complicated equipment to perform detailed examination and necessary actions or commands can be delivered back to the user. The data shared on the internet can be checked at the handheld node by using the client browser built in Windows Mobile OS.

1) MobileClient

MobileClient is a Windows Mobile 6.0 application running on a Smartphone (Blackjack3 i607). As a development environment, the Windows Mobile 6.0 classic Software Development Kit (SDK) and Microsoft .NET Compact Framework are used. This application simulates a socket client on the handheld device to gateway node and receives sensor data. By starting a client socket, MobileClient signals the gateway node to make a Bluetooth connection. Once the connection is established, 3 channels of data from the sensor node are retrieved and they are level of humidity, temperature, and light, measured at a frequency of once in approximately 3 seconds. MobileClient consists of two major components: the sensor data viewer and the data publisher to Web Service. Figure 6 shows the user interface of the sensor data viewer in MobileClient. The data is transferred from the gateway node in XML format. The sensor viewer parses the XML to extract necessary data, converts data to a user friendly unit such as Celsius and shows the user in the list box. The sensor viewer also saves the XML data in the file format for storage purposes. The streaming data from the sensor node is displayed in a continuous manner.

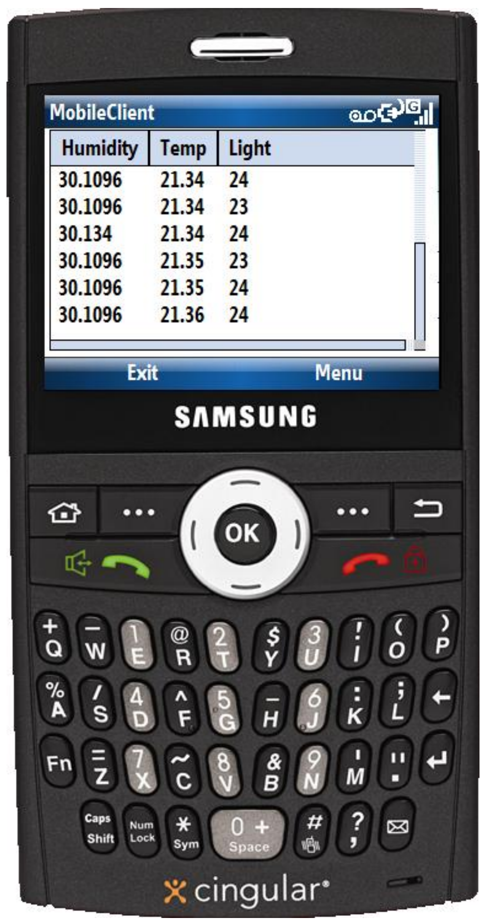


Figure 6. Sensor viewer in MobileClient

An example of sensor data saved in XML format on the handheld node is as follows.


```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOscopeMsg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<sourceMoteID>2</sourceMoteID>
<lastSampleNumber>3650</lastSampleNumber>
<channel>1</channel>
<data>
<unsignedShort>6719</unsignedShort>
<unsignedShort>6720</unsignedShort>
<unsignedShort>6720</unsignedShort>
<unsignedShort>6720</unsignedShort>
<unsignedShort>6721</unsignedShort>
<unsignedShort>6722</unsignedShort>
<unsignedShort>6721</unsignedShort>
<unsignedShort>6721</unsignedShort>
<unsignedShort>6720</unsignedShort>
<unsignedShort>6720</unsignedShort>
</data>
</ArrayOscopeMsg>
```


MobileClient also provides a user interface for publishing data that is retrieved from the sensor node to the database server through the Web Service interface. After a user of SNUC checks the data using the sensor viewer, the user may want to publish data for the further processing of data which cannot be done in his or her site. The data is published through a Web Service interface by using the long range connectivity, 3G network, provided by mobile operators. At this point, the data can be shared among all internet users which provide huge potential for the creation of many new services. Figure 7 shows the user interface details of the data publish implementation. This project implemented three web service APIs provided by the SensorMap [SenMap] project. SensorMap provides web service called DataHub [DatHub] and its APIs which mainly deal with maintaining sensor data. API ‘Register Sensor’ command registers the sensor node with location data, ‘Send Data’ command sends current temperature data transferred from the sensor node, and ‘Retrieve Last Data’ retrieves the latest data from the database server. The list box control displays the response from DataHub web service APIs.

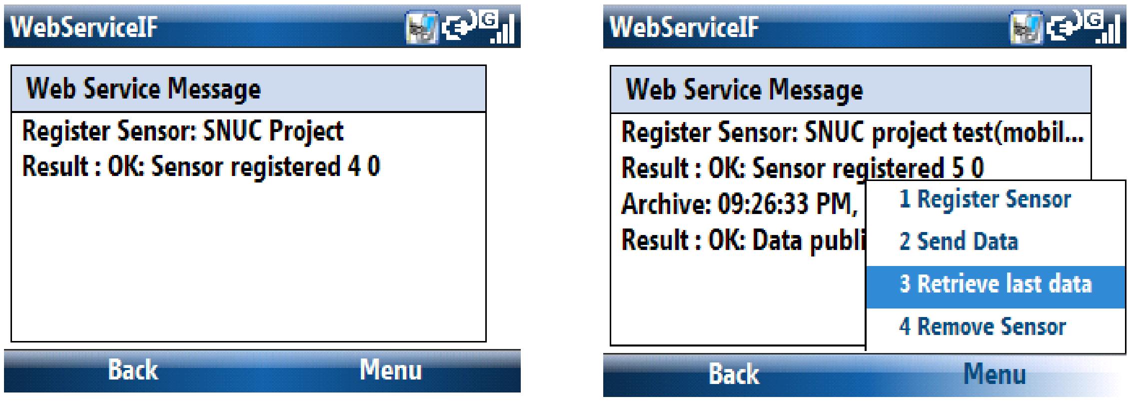


Figure7. WebService Interface in MobileClient

2) Client Browser
Windows Mobile operating system provides an Internet client browser. Through 3G connection, a handheld node user can browse the web site that contains information regarding the current sensor data. For example, the building maintenance engineer can delegate the diagnosing job of current data set to the main computer running a more sophisticated program in his or her company and review the result using the client browser on the fly.

4.5 SNUC Service Center

SNUC Service Center is simulated by using Microsoft Research's SenseWeb [SenWeb] project. SenseWeb project provides web service interface that allows data owners to post data and data owners can visualize the data through a geographical web site called SensorMap [SenMap]. The data published through the web service interface on the handheld device node are stored in the database server. Once the data is stored in the database server, any web server can process or examine data and tailor information to provide service to the clients of the SNUC system. The network layer of the SNUC service center uses a local area network since the service center is built on a traditional distributed computing infrastructure. Figure 8. shows published sensor data by this SNUC project in Tacoma, WA area. Figure 9. Shows another example of a more dense population of sensor networks published to SensorMap.

Figure 8. SNUC data published to SensorMap

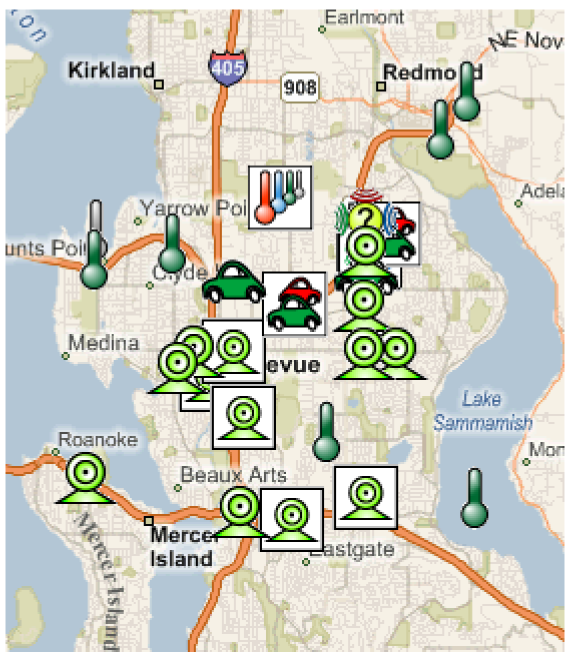


Figure 9. Dense population of sensor publishers in Bellevue, WA area

4.6 SNUC example scenario

As described in the 3.1 SNUC application scenario, here is an example of the SNUC implementation details in action. Assume that small patches of wireless sensor networks are deployed in a mattress factory building in Tacoma, WA in the necessary areas such as the boiler room and electricity control room. Sensor nodes in each patch of sensor networks are connected to a gateway through 2.4 GHz 802.15.4 RF signal. The gateway node receives streams of data from the base-node sensor that is connected to the USB port. The Moteforwarder receives raw sensor data and invoke a socket server so that MicroServer component can attach to. MicroServer runs pre-defined services configured by MSTML XML and waits for connection from the handheld node. A building maintenance engineer walks in with his or her Bluetooth capable handheld device and triggers in the gateway an event that the handheld device wants to connect. The connection is made over Bluetooth and the MicroServer processes raw data and streams sensor readings in XML format. The MobileClient application on the handheld device node parses the data and shows interesting data to the engineer. The engineer checks the data and if necessary, sends the data to the back-end server for further processing or information sharing. The engineer can examine the processing result by using the client browser on the handheld device over a 3G WAN network. The data stored in the back-end server can be tailored for sharing widely such as with SensorMap.

5. Conclusion and Future Research

The Sensor Network with User-supplied Connectivity (SNUC) collects data from the sensor network through hand-held device user's connectivity and shares the resources through a distributed computing paradigm. This project showed the possibility to leverage the data from heterogeneous small-scale sensor networks through a mobile network thus creating valuable business opportunities by fortuitous users' contributions – their nomadic interactions with the networks.

In this project, the handheld devices are the main method of connecting the sensor networks to the traditional computing infrastructure. By providing long-range connectivity to the sensor networks, data from heterogeneous networks are available to interested parties in real time (agility). In terms of ease of deployment, the handheld node removed complexity of routing and power management of traditional wireless sensor networks necessary to overcome physical limitation of the sensor, which makes the deployment simple. Also compared to the rigid hierarchy of data distribution structure of most wireless networks, SNUC is open to many handheld device users thus providing more opportunity to gather interesting data from data owners.

The SNUC project can be further improved by adding a few features. First, controllability on the handheld device is the most in need. Currently the data streams from the sensor node to the handheld node once connection is made. Providing database to gateway node and retrieving data through a common query set would enable a true event based system that saves power of the sensor network. Second, the SNUC service center can be improved by providing a relational database on the server, and a web site that uses the database through web service. This service center can then be used to display SNUC data to other services such as SensorMap.

The SNUC project envisions a complete framework that provides ways to join the framework, publish data through nomadic handheld users' contribution of connectivity to world. One possible future research direction is replacing the gateway node with a handheld node. The cost of sensor network deployment will be reduced as well as complexity. This lightweight sensor network will be especially useful for applications such as traffic condition monitoring since the denser handheld to sensor network formation can be interpreted as more traffic.

References

[RYF:Afr] RY Fu, H Su, et al., A Framework for device capability on demand virtual device user experience. IBM Journal of research and development; Sep-Nov 2004; 5/6; ABI/INFORM Global pg.63

[FRA:The] Frank Siegemund et al., The value of handhelds in smart environments. Personal and Ubiquitous Computing. London: Mar 2005. Vol. 9, Iss. 2; p. 69

[JAS:The] Jason Hill et. al., "The platforms enabling wireless sensor networks", Communications of the ACM, volume 47,issue 6,pages 41-46, Jun. 2004.

[SAR:Nom] Sarikaya, B.“ Nomadic User Approach to Build-. ing Mobile Wireless Sensor Networks,” Proceedings of International Workshop on Network Security and Wireless Communications, Sendai, Japan, Jan. 2005.

[SRD:Ena] Srdjan Krco et.al., "Enabling ubiquitous sensor networking over mobile networks through peer-to-peer overlay networking", Computer Communications, Volume 28, Issue 13, Pages 1586-1601, 2005

[ZigBee] http://www.zigbee.org/

[MSRSen] MSR Networked Embedded Sensing Toolkit. Available at http://research.microsoft.com/nec/msrsense/

[Boomer] Moteiv’s TinyOS distribution available from http://www.moteiv.com/software/

[Nesc] TinyOS programing language http://sourceforge.net/projects/nescc/

[WinMob] Windows Mobile Web Site available from http://www.microsoft.com/windowsmobile/default.mspx

[SenMap] Microsoft Research, SensorMap site available at http://atom.research.microsoft.com/sensormap/

[SenWeb] Microsoft Research, Sense Web project available at http://research.microsoft.com/nec/senseweb/

[DatHub] SensorMap web service interface http://atom.research.microsoft.com/sensordatahub/datahub.asmx