Implementing a Wireless Base Station for a Sensor Network

By

Heewon Song

B.S. Civil, Urban and Geosystem Engineering

Seoul National University, 2001

Submitted to the Department of Civil & Environmental Engineering

In Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN

CIVIL AND ENVIRONMENTAL ENGINEERING

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLGOGY

February 2004

Signature of Author _____

Department of Civil and Environmental Engineering

January 16, 2004

Certified by _____

Ruaidhri M O'Connor

Associate Professor, Department of Civil & Environmental Engineering

Thesis Supervisor

Accepted by _____

Heidi M. Nepf

Chairman, Departmental Committee of Graduate Students

Implementing a Wireless Base Station for a Sensor Network

By

Heewon Song

Submitted to the Department of Civil & Environmental Engineering on January 16, 2004
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Civil and Environmental Engineering

Abstract

Using wireless sensor networks for monitoring infrastructure is a new trend in civil engineering. Compared with traditional ways to monitor infrastructure, wireless sensor networks are cheap, safe, and compact. However, there are many available wireless communication techniques and hardware for a wireless sensor network. Therefore, it is an important step to choose the best communication method and hardware to construct a wireless sensor network for a particular infrastructure.

The London Underground project, which is described in this thesis as a reference case study, demands real-time data transmission, low-power network, and wireless network communication, and also a hardware/software system to collect, archive and display data from the monitoring activity. We consider the trade-offs in choosing 802.11b as a communication method. A web service architecture for data visualization is then described. Finally we discuss the appropriate selection of a computer device to serve as the base station.

Thesis Supervisor: Ruaidhri M O'Connor
Title: Associate Professor of Civil and Environmental Engineering

Acknowledgements

I would like to dedicate this thesis to my parents Kyung Eun Song and In Kyung Kang for their love and support and providing me with the best.

Great thanks go out to Professor Ruaidhri M. O'Connor for his expertise and support.

I would also like to thank Professor George Kocur for his advice and guidance.

I would also like to thank Sivaram Cheekiralla for providing invaluable help.

Last but not least, I would like to thank Debbie Leavy for her practical and emotional support ever since I knew her first.

Table of Contents

List of Figures

List of Tables

Chapter 1- Introduction

Information Technology is one of the newest disciplines to influence science and engineering, and civil engineering is one of the disciplines with the longest history. Because of the great time interval between the points when these two fields were born, the combination of two areas might look unusual. As we shall see, however, joining two fields can bring great benefits to both of them. The London Underground project that is referred to as a case study is a good example of adopting information technology in the civil engineering field. Conversely, the implementation of IT system in civil engineering projects brings new needs for hardware and software.

1.1 Information Technology and Civil Engineering

Old infrastructure poses many problems in terms of safety and serviceability. Therefore, to maintain old infrastructure is a great concern for civil engineers, and considerable money and labor are put into this task. Monitoring and maintaining infrastructure, which in the past have been done by very traditional ways, can experience a revolution when these engineering jobs are combined with information technology. With the recent advances in wireless technology and cheap computing power, there is an increasing trend towards the use of wireless sensors for infrastructure monitoring. One such example of using wireless sensors for the civil engineering

tasks is the London Underground project that utilizes the latest wireless technology to monitor the existing tunnel system.

Maintenance and rehabilitation (M&R) should start with measuring the degree infrastructure's safety and serviceability, which stands for the level-of-service provided to the users. Additionally, M&R needs to measure the structural adequacy, which is the capability of the facility to withstand the loads to which the infrastructure is subjected, and the maintainability, which is its ability to perform throughout its design life without continuous or significant maintenance. However, a traditional M&R needs to be reconsidered. First of all, measuring the performance is difficult to perform in real time. The condition of the infrastructure might be changing constantly, but all we can gain from traditional measures of performance is the condition solely at the time of examination. It is impossible to be certain that the infrastructure is safe at a future time based on the past data. Furthermore, manual inspection monitoring infrastructure can be dangerous since it involves exposing personnel to hazardous conditions. Another concern is the cost of monitoring. Older infrastructure requires more components to be monitored more often. Depending only on intermittent monitoring can lead to higher expense for M&R.

Advances in information technology can be very useful in reducing these problems. Modern communication technology provides millisecond, or even smaller, transmission time,

which equals real-time operation, even though there are some delays in a wireless network. Another advantage of using information technology for monitoring infrastructure is that we do not need to worry about the safety of workers. Once installed, a monitoring system sends data to personnel at a remote and safe location. Also, once installed, the cost of labor for manual inspection can be saved. For these reasons, it would be very useful if we were able to come up with a strategy to predict the level of services or hazards involved in maintaining aging infrastructure.

1.2 Case Study of Wireless Sensor Network for London Subway

The London Underground project is a good example of a monitoring infrastructure project that utilizes the latest wireless technology. In this case, monitoring means real-time measurement of displacements and deformation. This project is performed through the CMI (Cambridge MIT Initiative) collaboration of MIT and Cambridge University, UK. The London Underground project requires the measurement of displacements of an existing tunnel while the construction of a new tunnel continues underneath. The new tunnel is a part of the Channel Tunnel Rail Link (CTRL) project. The CTRL line, which is under the Circle line, is part of the London Underground system. The Circle Line is one of the oldest tunnels of the system, which was built in 1884. It is constructed of cast iron, and is 12-16 feet in diameter. The new tunnel

will pass underneath by 10 meters, at a rate of 40 meters per day. The new tunnel is almost perpendicular to the old tunnel (see Figure 1).

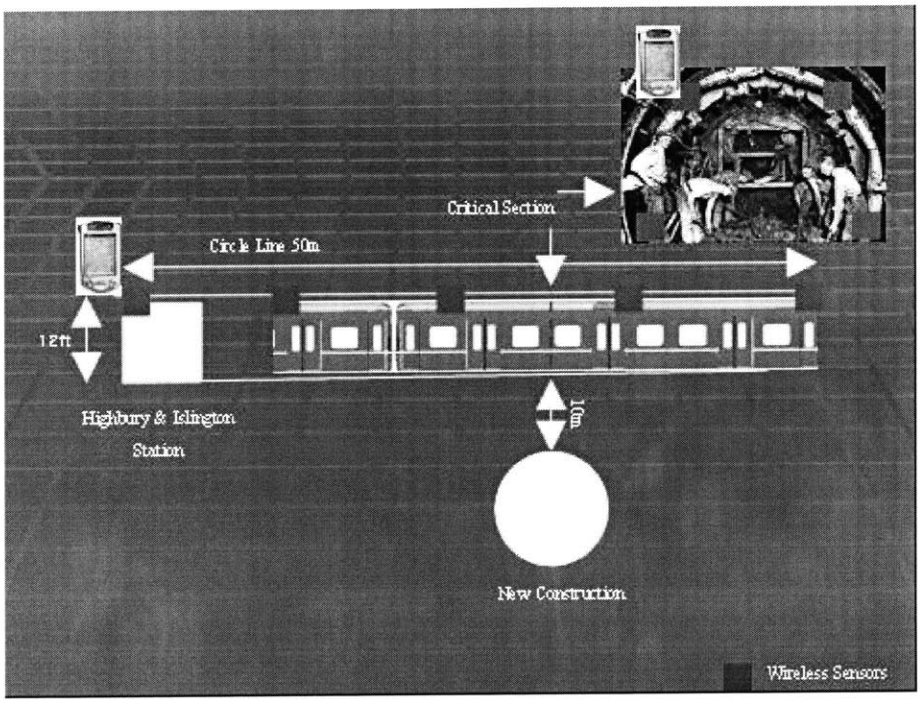

Figure 1. The London Underground Project's Problematic Situation

The new construction of the CTRL line tunnel can cause settlement in the Circle line, but the London Underground office hopes to operate the Circle line during construction. Therefore, the CMI project group has decided to install a wireless sensor network to measure the liner displacement and deformation from the closest station. If the displacement is greater than 5mm, London Underground office must stop the operation of the Circle line. The

construction work for the CTRL line is expected to take place 24 hours a day and constant monitoring is required to allow the Circle line to operate safely. With traditional monitoring methods, it is difficult to check the Circle line's condition continuously.

After reviewing the London Underground project as a reference case study, we designed the system layout for the wireless sensor network project (see Figure 2). Wireless sensor units are placed along the tunnel to measure displacement. This data is transmitted to a base station that runs a program to organize the data into TCP/IP packets. The base station is connected to a computer that runs a TCP server to forward this data. A client program reads the incoming data, publishes this data to a remote web server, and archives the data. A web service provides access to real time results, which can be shown to the authenticated clients. Details about this sensor network will be discussed in Chapter 3 and Chapter 4.

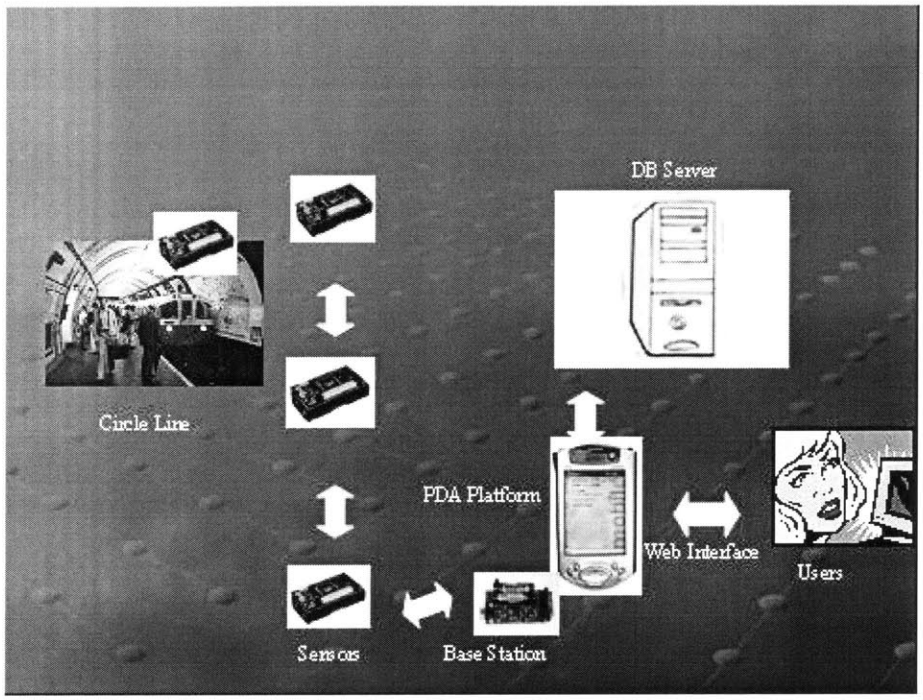

Figure 2. Conceptual System Layout of the Wireless Sensor Network Project

1.3 Difficulties Designing a Wireless Sensor Network

Even though wireless sensor networks offer a new era of monitoring and maintaining infrastructure, there are a number of difficulties in designing wireless sensor network systems for monitoring infrastructure. First of all, low power consumption should be a feature of a wireless sensor network, because most parts of the network are in locations that might have few power connections. In addition, even if there are enough power connections, a wireless network system being fixed next to the power connections and immovable is less valuable. Hence, developing a low-power consuming system should be a primary issue of a wireless sensor

network. A second issue is finding safe places for network devices. The devices should be put inside or around infrastructure, but the devices should not obstruct serviceability of the infrastructure. Therefore, the smaller devices might be more useful if other conditions are the same. We considered these difficulties while we were designing the wireless sensor network project, which is described in Chapter 3 and 4.

Chapter 2- Wireless Communication for a Sensor Network

To choose an appropriate wireless communication method for a sensor network is important because it affects the ability to monitor infrastructure. Questions that should be answered to choose the best wireless communication method for a sensor network are how wide its scope is, how strong its signal is, how fast its data transmission rate is, and how it can solve problems such as noise and multipath. In this chapter, basic concepts of wireless communication are described first, and then possible communication methods for our wireless sensor network project are listed.

References for this chapter are listed in the Reference chapter at the end of the thesis.

2.1 Radio Spectrum

2.1. 1 Radio and Wireless Communication

Radio is at the heart of wireless communications. Radio is the name given to the types of electromagnetic waves that can be used for communication purposes. Electromagnetic waves are generated whenever an electric charge is accelerated either in the positive direction or negative direction. Wireless communication needs a radio transmitter and a receiver. A radio transmitter works by vibrating electrons, the charged particles that surround all atoms and are responsible for electricity. The frequency of the wave depends on how fast the electrons are

vibrating: the faster they move, the higher the frequency is. A receiver uses the same principle in reverse.

New technologies have expanded the scope of radio constantly. At the low-frequency end, waves of around 5kHz have been used to send signals underwater. At the upper limit, the overlap between radio waves and microwaves become particularly valuable in the late twentieth century, because this region can be used for high-capacity communications over a relatively small area. Microwave radio is employed by all cell phone system and by most satellites.

2.1. 2 Radio Wavebands

Wavelength	***Frequency***	***Common Name***	***Main Purposes***
Above 100 km	Below 3kHz	Extremely Low Frequency (ELF)	Submarine Communications
10-100 km	3-30 kHz	Very Low Frequency (VLF)	Maritime Communications
1-10 km	20-300 kHz	Low Frequency (LF) or Long Wave (LW)	AM broadcasting
100-1,000 m	300-3,000kHz	Medium Frequency (MF) or Medium Wave (MW)	AM broadcasting
10-100 m	3-30 MHz	High Frequency (HF) or Short Wave (SW)	AM broadcasting, amateur radio
1-10 m	30-300 MHz	Very High Frequency (VHF)	FM broadcasting, TV
0.1-1 m	300-3,000 MHz	Ultra High Frequency (UHF)	TV, cell phones
10-100 mm	3-30 GHz	Super High Frequency (SHF)	Fixed wireless, satellites
1-10 mm	30-300 GHz	Extra High Frequency (EHF)	Satellites, radar

Table 1. Radio Wavebands

The radio spectrum is subdivided into a series of regions known as wavebands. Table1 shows the names and uses of these, in order of increasing frequency. Since frequency is measured on a logarithmic scale, each band contains ten times the spectrum of the one above it in the table. For example, there is a thousand times as much spectrum UHF band, used for most broadcast TV, than in the MW band, used for most AM radio. This means that the high-frequency bands are most useful for fast data services, whereas the lower ones are suitable only for broadcasting. Each signal has a different range according to its frequency, so signals with high frequency have a much shorter range than signals with lower frequency. It can be explained by the relationship between wavelength and attenuation. As radiation wavelength gets shorter, more things are able to block it. Broadcast radio is a good example of this attenuation effect. Stations in the LW band are able to cover several European countries with a single transmitter, whereas those in the VHF band typically serve only a single city. The short range is a problem when trying to communicate across a great distance, but it can be a benefit when designing a cellular network because it means that different cells can reuse the same spectrum. Satellite systems use high frequencies, but extend the range by using very powerful transmitters and, in many cases, a parabolic dish to focus the radiation. Many frequencies in the HF band are able to travel right around the world by reflecting off the ionosphere, a layer of electrically charged particles at the top of Earth's atmosphere. Before satellites, this was the only way of

communicating across oceans, but it was not reliable. The ionosphere intensifies with sunspot activity and gets thinner at night, because ultraviolet radiation from the sun creates the ionosphere. However, a short wave radio receiver can be used to hear broadcasts around the world. There are three radio wavebands, UHF, SHF, and EHF, that are collectively known as *microwaves* because their small wavelengths. Since microwaves have short range and high bandwidth, these waves are very useful for communications. The microwave spectrum is subdivided into bands, shown in Table2.

Wavelength	Frequency (GHz)	Band	Main Communication Use
193-769 mm	0.4-1.5 GHz	L	Broadcasting, 1G cellular
57.7-193 mm	1.5-5.2 GHz	S	2G and 3G cellular, WLAN, Bluetooth
48.4-76.9 mm	3.9-6.2 GHz	C	Satellite, WLAN
27.5-57.7 mm	5.2-10.9 GHz	X	Fixed, satellite
17.1-27.5 mm	10.9- 17.5 GHz	Ku	Fixed, older satellite
8.34-17.1 mm	17.5-31 GHz	Ka	Fixed, newer satellite
6.52-8.34 mm	36-46 GHz	Q	Fixed
5.36-6.52 mm	46-56 GHz	V	Future satellite, fixed
3.00- 5.36 mm	56-100 GHz	w	Fixed, WLAN, 4G cellular

Table 2. Microwave Band

Microwaves were developed first for a radar system by the military. Microwaves have become more popular since they were assigned a new role in communications. The most

important and crowded band is currently the S- band, because this has the right combination of capacity and range for cellular systems and WLAN. However, microwaves are easily blocked by obstacles such as walls and hills, and even weakened by rain and clouds because of their short wave length.

Therefore, to select the best wireless communication method for a particular wireless sensor network, we need to examine and identify the scope and range requirements of the network, and the environmental conditions surrounding the infrastructure. For example, if the network needs a wide scope, or many obstacles surround the infrastructure, low frequency waves are better than short frequency ones.

2.1. 3 Noise and the Multipath

One of the reasons why waves are great for communications is that they cannot collide, so two waves will just pass through each other, no matter what the frequencies. However, receivers can have difficulty with multiple incoming waves when a receiver has to pick up two or more signals at the same frequency because it is impossible to distinguish one from the other. Unwanted signals are called *noise* and that can severely limit the effectiveness of a radio signal by causing interference. Some interference caused by noise is strong enough to drown out the real signal. For example, satellites can be drowned by the sun, a much more powerful source of radiation than any communications antenna. More common is two similar communication

signals that meet, either reinforcing each other or canceling each other out.

To make sure that radio signals are safe from noise, frequencies should be carefully designed and managed. However, well-managed interference even can have some applications. Lasers use the constructive interference to increase the energy of their beams. Destructive interference is not used with electromagnetic radiation, but it has been applied with great success to sound waves. In an 'anti-noise' system, a computer makes a wave with characteristics exactly opposite to those of an unwanted sound, thus canceling out the noise. Listeners ideally hear no noise, or at worst a very quiet sound. The beauty of this system is that other sounds are not affected at all. Fortunately, the same principle can work with radio, jamming a transmission in such a way that it appears never to have been sent. Even though the principle of radio anti-noise systems is same with the sound anti-noise systems', radio anti-noise systems would require more knowledge of the transmission than we have now because of the difference of speeds between sound waves and radio waves. Sound waves travel very slowly allowing a computer to intercept them, and then generate the opposite signal before they reach a listener, but radio waves are too fast to do the same process. It would not work with an inherently unpredictable signal, such as nuclear radiation.

Another interference source is multipath. Multipath interference affects every type of wireless communication designed for use indoors or in an urban environment, and this

interference cannot be overcome by turning up the transmission power or through cooperation among users. Therefore, the wireless sensor network that is installed inside infrastructure can suffer because of multipath interference especially when many wireless devices are installed for that network. The various different paths that a radio signal can take from one point to another cause multipath interference. The best path between two points is a straight line, but radio transmitters hardly target their signals directly toward one receiver. They broadcast in many different directions so that a user can pick up the same signal reflected off buildings, walls, and other objects. The different versions of the same wave can cause destructive interference however, making the signal weaker or even unintelligible. Some new technologies are designed specifically to reduce multipath interference. However, it puts an important constraint on many existing wireless systems by reducing the available capacity and making it harder to use devices deep inside a building. Therefore, we should be careful to adopt an anti-multipath system for a sensor network, which is installed inside infrastructure.

2.2 Short-Range Wireless Networks: Possible Wireless Communication Methods for a Sensor Network

The wireless sensor network considered as a case study for this project is designed to be installed inside a tunnel such as the London Underground. Therefore, it is necessary to research

short-range wireless communication methods to find an appropriate one since this environment is severely constrained in terms of physical confinement. In this section, brief theoretical descriptions of Wireless LAN, Bluetooth, cordless Telephony, and IrDA will be presented. We will use this review to help us to choose the most appropriate technology for the wireless sensor network.

2.2. 1 Wireless LAN

Wireless LANs are more flexible than wire-based systems and faster than other wireless networks. Wireless LANs work in a wide range and have a considerably fast and stable data transmission rate compared with other wireless communication methods, such as Bluetooth and IrDA. Therefore, wireless LANs offer a reliable wireless communication method for the sensor network.

2.2. 1.1 History of Wireless LAN

In 1997, the FCC allocated 300 MHz of unlicensed spectrum in the Industrial Scientific and Medical (ISM) bands of 5.150-5.350 GHz and 5.725-5.825 GHz for the express purpose of supporting low-power license-free spread spectrum data communication. This allocation is called the *Unlicensed National Information Infrastructure* (UNII) band.

The IEEE 802.11 Wireless LAN working group was founded in 1987 to begin standardization of spread spectrum WLANs for use in the ISM bands. Despite the unrestricted

spectrum allocation and intense industry interest, the WLAN movement did not gain momentum until the late 1990s, when the phenomenal popularity of the Internet combined with wide scale acceptance of portable, laptop computers finally caused WLAN to become an important and rapidly growing segment of the modern wireless communications market place. IEEE 802.11 was standardized in 1997 and provided interoperability standards for WLAN manufacturers using 11Mbps DS-SS spreading and 2Mbps user data rates (with fallback to 1 Mbps in noisy conditions). In 1999, the 802.11 High Rate standard (called IEEE 802.11b) was approved, thereby providing new user data rate capabilities of 11 Mbps, 5.5Mbps in addition to the original 2 Mbps and 1Mbps user rate of IEEE 802.11, which were retained. The DS-SS IEEE 802.11b standard has been named Wi-Fi by the Wireless Ethernet Compatibility Alliance (WECA), a group that promotes adoption of 802.11b DS-SS WLAN equipment and interoperability between vendors.

System	Theoretical Capacity	Real Max. Throughput	Spectrum	Air Interface	Status as of 2002
IEEE 802.11(FHSS)	1 Mbps	0.5 Mbps	2.4 GHz	FHSS	Obsolete
IEEE 802.11(DSSS)	2 Mbps	1 Mbps	2.4 GHz	DSSS	Obsolete
IEEE 802.11b	11 Mbps	6 Mbps	2.4 GHz	DSSS	Popular
IEEE 802.11g	54 Mbps	31 Mbps	2.4 GHz	OFDM	Near Future
IEEE 802.11a	54 Mbps	31 Mbps	5 GHz	OFDM	New
ETSI HiperLan1	23.5Mbps	Unknown	5 GHz	TDMA	Abandoned
ETSI HiperLan2	54 Mbps	31 Mbps	5 GHz	OFDM	Near Future
HomeRF	1 Mbps	0.5 Mbps	2.4 GHz	FHSS	Obsolete
HomeRF2	10 Mbps	6 Mbps	2.4 GHz	FHSS	Endangered
5- WING/5-UP	104 Mbps	72 Mbps	5 GHz	OFDM	Future

Table 3. Wireless LAN Standards

2.2. 1.2 802.11b Technology

Most of the recent wireless LANs are based on the IEEE 802.11b standard, better known *Wi-Fi* or *Wireless Ethernet.* The name *Ethernet* is mostly intended to reassure computer networkers, but it also describes the kind of multiplexing scheme that all 802.11 networks use. Ethernet is one of several standards for wired LANs that IEEE set in the 1980s, officially IEEE 802.3. It proved to be far more popular than the others, mainly because of its simplicity: Every computer on a network segment shares a single communications channel, which can carry only one data packet at a time. Rather than try to organize specific time slots for each computer, as TDMA would do, Ethernet simply allows each to transit data whenever it needs to. If two

happen to try to use the channel at once, a collision occurs, and both packets are lost. Both computers then try again, each waiting for a random time interval before doing so.

The wireless version of Ethernet is slightly more complex. Since it is trying to anticipate collisions before they occur, 802.11 terminals check that the air is clear by transmitting a short test signal before sending a full packet of data. This system is known by the acronym CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance). This saves on bandwidth by reducing the amount of time that the link wastes on collisions, but is still broadcast in nature: It doesn't reserve a communications channel for any particular user, as do the protocols used for cell phone transmission.

All 802.11 networks use CSMA/CA to share a link among users, but they differ in the type of physical technology used. The original 802.11 specifications, set in 1998, called for three different versions, Optical, FHSS and DSSS. They are mutually incompatible.

• **Optical,** using infrared transmitters such as the ports built into many laptop PCs. Though proprietary wireless LANs based on infrared were once relatively common, infrared 802.11 networks were never actually built.

• **FHSS** (Frequency-Hopping Spread Spectrum), which rapidly cycles between frequencies many times each second. It has 79 different frequencies to choose from in the ISM-2.4 band, so even though some may be blocked by interference, the theory is that others should be clear. The

FHSS version of 802.11 can reach 1 Mbps.

• **DSSS** (Direct-Sequence Spread Spectrum), which is the same as CDMA cellular: It transmits on all frequencies simultaneously, hoping to overcome interference. This increased the capacity to a maximum of 2 Mbps but also used more power. Most people consider a doubled data rate to be worth a slightly shorter battery life, so DSSS became the most popular type of 802.11.

The IEEE later made several enhancements to the physical layer, shown in Figure 3. The first of these to reach mass production was 802.11b, based on an enhanced version of DSSS called CCK (Complementary Code Keying). This uses extra CDMA codes, boosting the data rate when reception is clear. It's backward-compatible with the older DSSS system, automatically dropping down to 2Mbps to communicate with them if necessary. It is not compatible with FHSS-based 802.11 networks, but backward compatibility with any other type of wireless LAN is not often needed. Most of the 802.11 devices, which carry the Wi-Fi mark, support full 802.11b.

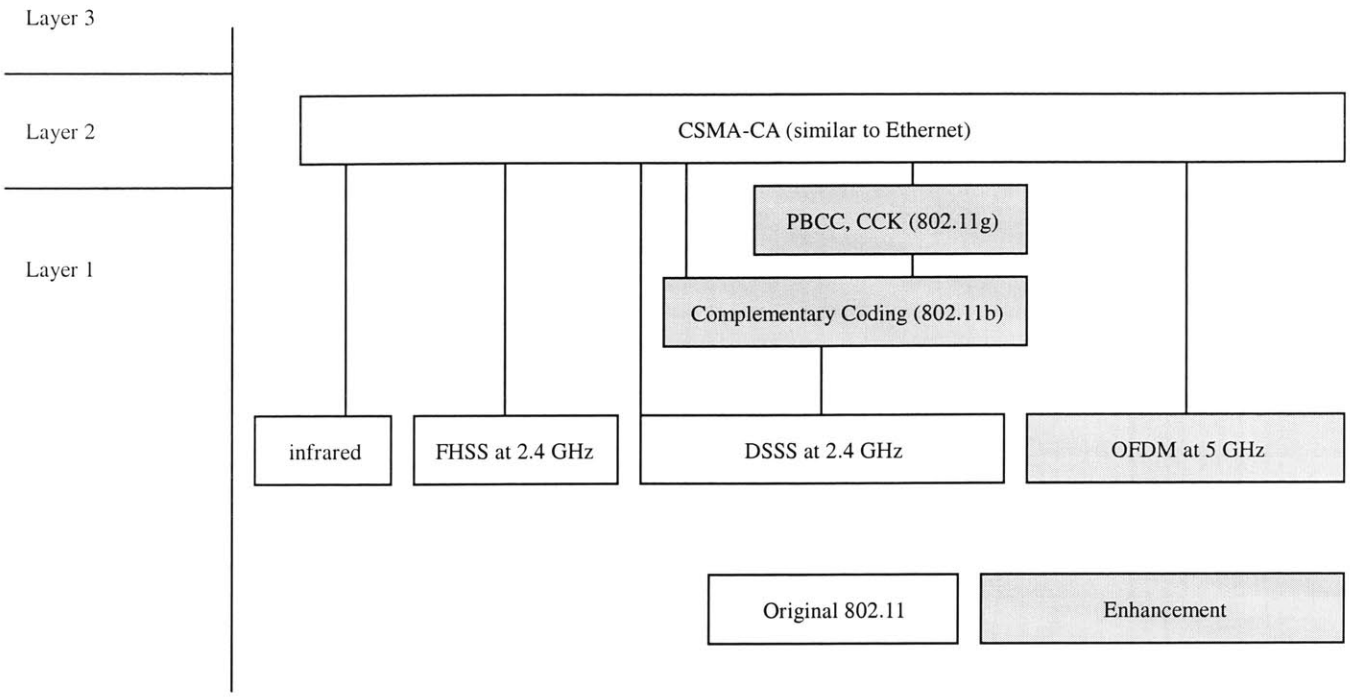

Figure 3. IEEE 802.11 Protocol Stack

The theoretical maximum capacity of 802.11b is 11Mbps. It pushed wireless LANs through an important psychological barrier, matching the speed of the original Ethernet standard. However, the number is misleading. It refers to the total physical layer capacity, much of which is used by the protocol itself, so it is not actually available for data. The maximum data rate of an 802.11b network is really only 6 Mbps, and that can be achieved only under optimum conditions – over a short range and with no interference. It quickly drops when packet collisions or other errors occur. A 50% error rate will reduce the real throughput by about two thirds, to only 2Mbps.

To prevent these errors, 802.11 automatically reduces its physical layer data rate when

errors occur, because lower data rates are more resistant to interference. If reception is particularly bad, it will drop down to regular 802.11 at 2 Mbps or 1Mbps. Collisions and interference can reduce the real throughput to less than a tenth of the touted 11 Mbps.

2.2. 2 Bluetooth

Bluetooth is one of the possible communication methods for the wireless sensor network project mainly because of its small size, low power consuming, and reasonable price. However, Bluetooth also has weak points such as a very short range and an unstable connection. Bluetooth also creates interference with other wireless communication methods in ISM band very easily, so we should be careful to use them together for one sensor network.

Bluetooth is one of the most overvalued wireless technologies, and justifiably so. It promises to cram an entire wireless system onto a single chip, which is cheap enough to be built into all mobile phones, PCs, PDAs, and eventually other devices. Rather than a LAN, Bluetooth's advocates describe it as a PAN (Personal Area Network).

These PANs will surround everyone. Blutooth promoters hope, embracing every device that people carry. That will automatically interface with the refrigerator at home, the security system in the office, and mobile phones on the move. As people interact, their PANs will connect to beam over contact details, data files, or even digital cash.

Bluetooth dates back to an Ericsson project in 1994, which tries to find a way for mobile phones to communicate wirelessly with accessories such as hands-free kits. In May 1998, it joined with Nokia, Intel, Toshiba, and IBM to form the Bluetooth Special Interest Group (SIG), an alliance aimed at developing an open standard. It was named after Harald Bluetooth, a Viking king who unified Denmark and Norway in the tenth century. The founding five companies opened SIG to new members, and the results were unprecedented. In less than two years, nearly 2,000 others, even winning over initial skeptics such as Microsoft, joined them. The main condition of membership is that all members of SIG must let all others use their Bluetooth patents under reasonable terms, preventing particular companies from getting a lock on the standard.

Despite such heavyweight backing, Bluetooth has run into problems. Products were originally supposed to appear by the beginning of 2000, but a year later they were still very scant. It is simply such an ambitious project that it has taken far longer than expected. Bluetooth's stated aim is to create a single-chip radio with a range of 10meter, with a peak throughput of 720 kbps, and a cost under $5. Up to eight Bluetooth devices can be directly connected to one another in a *piconet.* Networks greater than eight are possible, but in this case, not every device will be able to transmit to every other because each one can see only seven. A network of more than eight is known as a *scatternet* and should be subdivided into piconets.

The $5 price point is important; Bluetooth will have to reach this level, and ultimately cheaper, if it is to be fitted to all electronic devices. The first chips cost substantially more and were put inside relatively high-priced PC or CF+ cards for existing laptops or PDAs. These are relatively pointless without other Bluetooth devices because a far higher capacity wireless LAN card can be purchased for around the same cost.

Bluetooth has been criticized for the way it used the crowded 2.4GHz ISM band. It is based on loosely on the frequency-hopping technique of the original 802.11 standard but cycles through frequencies much faster, hopping up to 32,000 times every second. Some in the wireless LAN industry describe it as a rude radio, because this rapid cycling is almost guaranteed to interfere with other systems. However, this should not harm Bluetooth itself: 802.11, HomeRF, and Bluetooth are all competing for the same spectrum, and the rapid cycling means that Bluetooth is likely to win a head-to-head battle.

More serious questions hang over interoperability, both at the radio layer and between the applications that will actually be using Bluetooth. In this area, it has learned from some of the mistakes made by IrDA. The Bluetooth SIG will not let any devices use the Bluetooth logo until they have been tested for compatibility with the reference design, and some protocols originally developed for IrDA have actually been licensed for Bluetooth. To help with interoperability, IEEE is trying to develop an official standard based on Bluetooth, under the

auspices of its 802.15 group. There are four separate sub-standards within this.

• **802.15 1** is the standardized system based on Bluetooth, with almost the same capabilities.

• **802.15 2** is an attempt to make 802.15 and Bluetooth radio less "rude," so that they can coexist with 802.11 LANs that use the same 2.4 GHz band. The most likely solution is a dual-standard transceiver, which will co-ordinate transmission to prevent interference.

• **802.15 3** is aimed at increasing the data rate to at least 20 Mbps. This is necessary for some multimedia applications, which use large files that would take several minutes to transfer over regular Bluetooth.

• **802.15 4** will be an even lower power version, with an unspecified (but probably also low) data rate. The theory is that lower power will mean a battery life of many months or even years, permitting it to be embedded inside smart cards or used as a security tag.

Bluetooth is intended to be used for many futuristic applications, but the first draft of the specifications specifies nine. Each application is described by a separate set of protocols and procedures called a *profile*. Not every Bluetooth device supports every profile, and the cheapest one supports only one. The profile system has attracted some criticism because it means that Bluetooth is effective nine standards, not one, and the number is likely to grow as more profiles are added. However, this SIG says that it is necessary to ensure interoperability. The profiles

share many common protocols, which are illustrated in Figure 4. This common ground should make it simple to adapt existing hardware for applications when additional profiles are published. The first draft includes 13 profiles, listed below. In addition to the nine applications, there are four system profiles (number 1,2,5, and 10), which include features common to one more applications.

1. **Generic Access Profile.** This is the core Bluetooth profile, responsible for maintaining links between devices. All Bluetooth devices need to include this profile, but in itself it is not sufficient for any useful applications. It includes functions necessary to use the entire core Bluetooth protocols, shaded light gray in Figure 4.
2. **Service Discovery Application profile.** This profile enables a user to access the Service Discovery Protocol (SDP) directly, to find out which Bluetooth services are available from a given device. SDP is included as part of the core, but without this extra profile it can be accessed only by application, not the user directly.
3. **Cordless Telephony Profile.** This is designed for what the Bluetooth SIG calls a "3-in-1-phone," meaning a cell phone with a Bluetooth chip that enables it to be used as a cordless phone. It can also be used for cordless-only phones or adding cordless telephony functions to any other Bluetooth-equipped device, such as a digital watch. It runs over the Telephony Control Service (TCS) protocol.

4. **Intercom Profile.** Also based on the TCS protocol, this allows two-way voice communication. It differs from the Cordless Telephony Profile in that it supports only connection between two Bluetooth users within a range of each other, not full telephony over the phone system or the Internet. Because Bluetooth has such a short range, it is unlikely to be used extensively.

5. **Serial Port Profile.** This profile allows Bluetooth devices to emulate a PC's serial port, using the RFComm protocol. It can emulate either an older RS232 or a newer USB (Universal Serial Bus) cable and is used by many higher level profiles.

6. **Headset Profile.** This specifies how Bluetooth can provide a wireless connection to a headset containing earphones and perhaps a microphone for use with either a computer or mobile phone. It uses the serial port profile and the AT (Advanced Technology) commands, which were originally designed to control modems and are so called because they are taken from an IBM computer of the same name.

7. **Dial-UP Networking Profile.** This is designed for Internet connection via a cell phone. At present, they have to do this through a special cable or perhaps IrDA, but in the future they should be able to use Bluetooth. It includes the serial port profile and Point-to Point Protocol(PPP), the same protocol used by ordinary modems connection to the Internet.

8. **Fax Profile.** This is very similar to the dial-up networking profile. It enables a mobile

phone to emulate a fax modem. Like the dial-up networking protocol, it uses PPP and the serial port profile.

9. **LAN Access Profile.** This is potentially the most useful profile. It is intended for IP data networking, enabling Bluetooth-equipped PCs to form a small wireless LAN among them or to connect to other LANs via a special access point. The SIG envisages these access points being placed within private networks and in public places, so that people with Bluetooth devices will be able to connect to the Internet.
10. **Generic Object Exchange Profile.** This profile controls how Bluetooth uses OBEX (Object Exchange), a client-server protocol taken from IrDA. It allows applications to exchange data directly, without having to use Internet-style packets.
11. **Object Push Profile.** This profile governs the exchange of electronic business card, short files that contain the same information as a regular business card but can automatically be filed into a device's database. The "push" in the name refers to the fact that this information must actively be given out by a user. To protect privacy, electronic business cards are usually not just beamed out to every device in range.
12. **File Transfer Profile.** This profile allows a device to access files stored on another. It can be used by any application that needs to transfer files between devices, from one device to another, or directly by users. Apart from linking two computers together, devices that

might use this profile include MP3 players and digital cameras, which would transfer music and image files.

13. **Synchronizations Protocol.** This helps to keep the data stored on different devices up to date. It is supposed to automate synchronization, so that a computer will automatically synchronize data with a mobile phone or PDA whenever they are within range of each other.

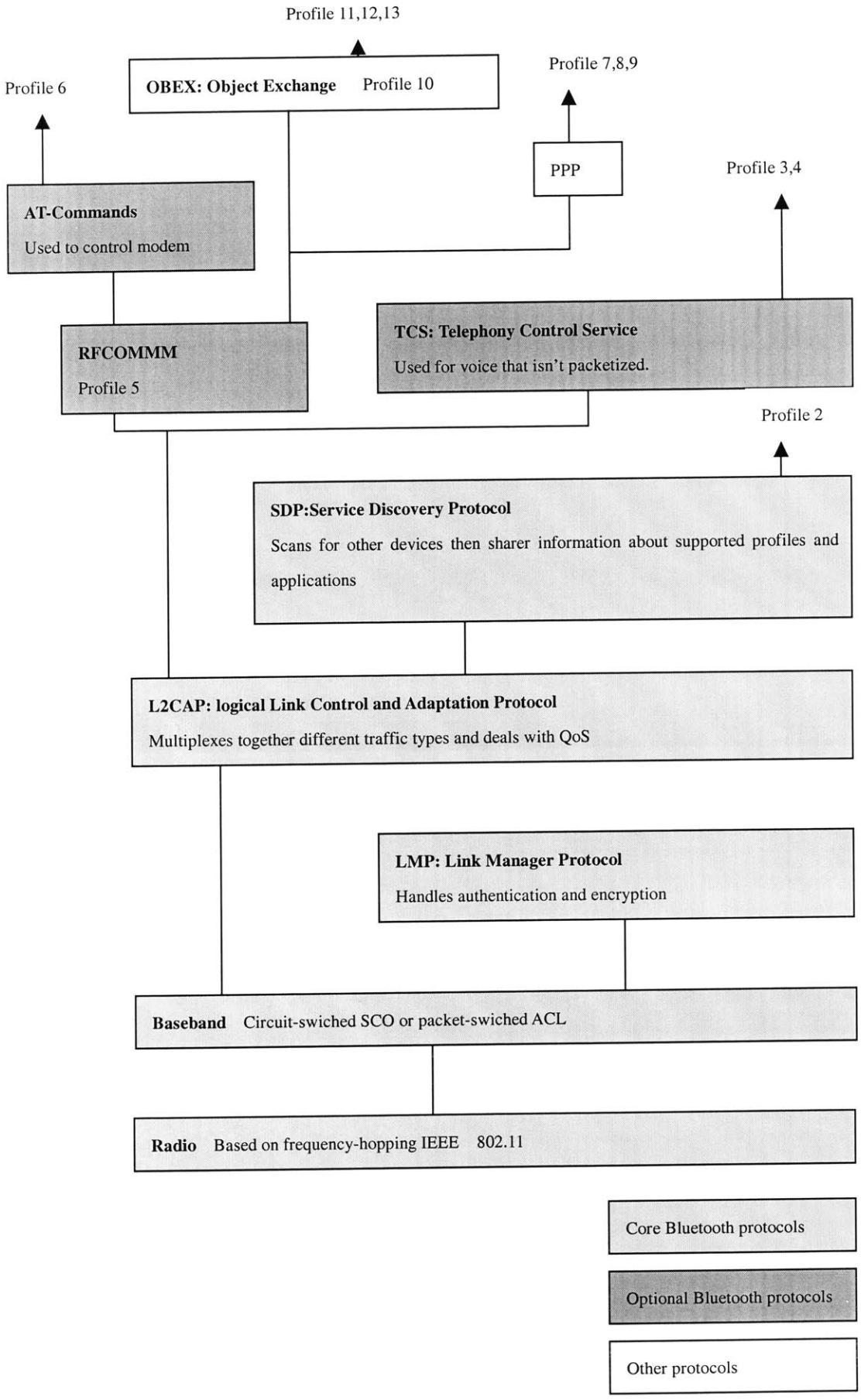

Figure 4. Bluetooth Protocol Stack

2.2. 3 Cordless Telephony

A cordless phone was supposed to be a major wireless communication method when it was born, but its applications have not been successful. Nowadays, there are not many existing applications. However, its rage is about 100m, which is long enough a small wireless sensor network, and its transmission power is low, so we looked at this technology to see if it could be the right communication method for our project.

A cordless phone, as opposed to a mobile, works with only one specific base station and uses unlicensed spectrum. The base station is known as a fixed point because it is connected to the regular fixed phone network. Transmission power is very low, for a range of 100 m or less. One of the applications that use the cordless telephony is telepoint system. The theory behind telepoint systems is that the systems allow the same phone to be used in both private and public networks. Telepoint operators set up hundreds of cordless fixed points around cities, often in areas such as shopping malls and subway stations, which are originally difficult for true mobile systems to reach. A home cordless phone is an attractive alternative to the long lines that usually formed in front of public payphones. The major weakness of telepoint system is that it could be used only make calls, not receive them. Because of this flaw, telepoint was never tried in the United States, and many European and Asian networks failed. An ETSI standard known as CT-2

(Cordless Telepoint System 2) was eventually abandoned.

Most American digital cordless system use proprietary spread spectrum technologies in one of the ISM bands. European and Japan have both defined standards, known respectively as Digital Enhanced Cordless Telephony (DECT) and Personal Handyphone System (PHS). The standards are very similar. Both use time-division duplexing (TDD) and TDMA, and offer a data rate of 64 kbps one way (or 32 kbps each way). This results in very clear voice quality and Internet access at near ISDN speed. But they have been put to different applications. DECT is still used almost entirely for private cordless systems, whereas PHS was the basis of a very popular telepoint network. Table 4 compared the technical details of the two systems. Because they employ TDD, the maximum number of users is only half the number of TDMA slots; each needs at least two slots, one to transmit and one to receive.

	PHS	DECT
Spectrum	1895-1918Mhz	1880-1900 Mhz
Modulation	QPSK	GMSK
Channel bandwidth	300 kHz	2 MHz
Gross data rate per channel	384 kbps	1152 kbps
Usable data throughput per channel	128 kbps	768 kbps
TDMA slots per channel	4	24
Maximum users per channel	2	12
Available in	Japan	Europe

Table 4. Comparison between PHS and DECT

Private base stations based on one of the public cellular standards, usually GSM, can allow people to use the same phone wherever they are, both to make and receive calls. If a cell were placed in the home or office, calls could be routed over the fixed rather than the mobile network. This system is called picocells, and it can be connected to the outside world in two ways. One is via the public telephone system, and the other is via the mobile network. Using the public telephone system, the network can be truly private, under the full control of whoever set

up the base station. It can be the easiest type to set up, but means that advanced mobile services, such as text messaging, are not available. People dialing the mobile number may also fail to get through, unless the mobile is still within range of a public base station or the user has remembered to set up a call divert system. Using the mobile network, some operators and companies collaborate on building picocells so that they can also be used by other subscribers to the mobile network. This would seem to be used by other subscribers to the mobile network.

2.2. 4 Infrared Light Waves

Infrared Light Wave is one of the most popular wireless technologies for mobile devices nowadays. However, to communicate using Infrared light wave, two devices should face each other directly and nothing should be between them. For these reasons, it is hard to adopt Infrared light waves for the wireless sensor network, but it is still a convenient and prevalent communication method for mobile devices.

IrDA is short for ***Infrared Data Association,*** a group of device manufacturers that developed a standard for transmitting data via infrared light waves. Increasingly, computers and other devices come with IrDA ports. This enables you to transfer data from one device to another without any cables. For example, if both your laptop computer and printer have IrDA ports, you can simply put your computer in front of the printer and output a document, without

needing to connect the two with a cable. Therefore, IrDA is supposed to be a simple way to replace cables when printing or sending data to another PC.

There are now four different versions of it, with no obvious way to tell which is built into a given device. The four versions of IrDA are like this: SIR(Serial Infrared) was the original standard and ran at 115kbps. This is the same speed as a standard serial port, on which the IrDA protocol was based. MIR (Medium Infrared) runs at 1.152Mbps, fast enough to transmit or receive television-quality video. It is not widely implemented, with most systems choosing either the serial or fast version. FIR (Fast Infrared) offers speeds of up to 4 Mbps. This is built into most new computers and is the standard setting on Windows 98 and 2000. VFIR (Very Fast Infrared) goes up to 16 Mbps and is not yet widely implemented. Undeterred, the Infrared Data Association is working on even future upgrades and claims this IrDA version can push the ports up to a blistering 50 Mbps. Such a high data rate, which is achieved by VFIR may seem like overkill for simple file transfer between computers, but the plan is for IrDA to be incorporated into other devices, such as digital cameras and MP3 players. At the current 4 Mbps, a full CD's worth of music in MP3 format would take longer than six minutes to transfer to a player, compared with less than 30 seconds at 50 Mbps.

One laptop may be able to send and receive infrared more than a hundred of times faster than another, even if they were built at the same time, cost the same, and had similar

processor, display, and memory configurations. Some high-speed transceivers are wrongly configured to run at the lower speeds, meaning that they can often be upgraded with software alone. The other advantage of IrDA is that the different versions of IrDA differ only in their data rate and are entirely backward compatible. A new device ought to be able to work with no older one, albeit at the slower speed.

Millions of them are installed worldwide, theoretically making it one of the world's most popular standards. But despite of its ubiquity, IrDA is hardly ever used. This is partly because at first it did not work very well; different companies implemented the system in different ways. Although these early problems have been almost solved now, the perception has prevented people from using it, which in turn has stopped the vendors from implementing it on more devices. This creates a vicious circle- users won't ask for IrDA unless it is fully supported, and it will not be fully supported until users ask for it.

IrDA's creators envisaged a world where people would print simple by pointing a computer at a printer, or link up to a network by pointing it at an access point. The problem is that few manufacturers have developed these printers or access points, meaning that the only thing many laptops get the chance to connect to is another laptop.

Few people have two laptops, but most laptop owners do have a desktop PC. The most practical application for many users would be to synchronize the two, but so far this has been

impossible. Although infrared transceivers cost less than one dollar, almost no companies build them into desktops. A notable exception is Apple, one of IrDA's first adopters. It has been promoting infrared data since the 1980s, using a proprietary system before the standard was defined.

IrDA ports support roughly the same transmission rates as traditional parallel ports. The only restriction on their use is that the two devices must be within a few feet of each other and there must be a clear line of sight between them.

Although designed only to connect two points together, IrDA can also be used to form complex networks. The principle is the same as for most wired LANs – every computer is connected separately to a central switch through which all data passes. An IrDA-based LAN is not as useful as a radio-based one, because the range is usually shorter, and it requires a line of sight between the switch and the terminal. But it has one enormous advantage – the infrared port is already built into many laptops. The port also consumes much less power than a separate plug-in card, and people can't lose it or forget it (unless they lose the entire computer). Because the data rates available from IrDA vary so much, nearly all these systems are switched. An ordinary hub would try to transmit all data to every computer, which would limit PCs equipped with 16 Mbps VFIR to the 0.1 Mbps of SIR.

Chapter 3. Conceptual Design of a Wireless Sensor Network

In previous chapters, the benefits of a wireless sensor network to monitor infrastructure and the theoretical background of wireless communication were described. This chapter presents a conceptual design of a wireless sensor network for the London Underground project. For better understanding of the design, details about motes, applications, and Web UI in this conceptual design are explained.

References are available at the end of the thesis.

3.1 Design Overview

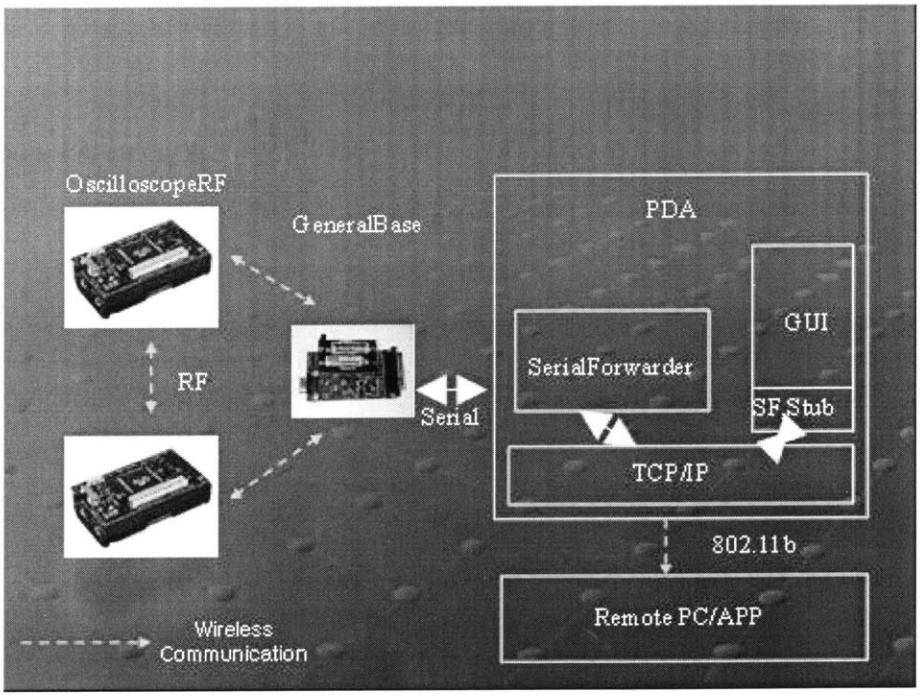

Figure 5. Logical Connectivity

Figure 5 explains the logical connectivity of the programs conceptually developed for the wireless sensor network project. The two main applications that were used for this sensor communication are OscilloscopeRF, and GeneralBase. Another application, SerialForwarder, was used for connecting the base station mote and TCP/IP, which is communicating with clients or remote applications.

3.2 MICA motes

The MICA mote is a second-generation wireless sensor module developed by the University of California, Berkeley and commercialized by crossbow.com. These devices are intended to provide a low power wireless sensor system to examine the possibilities of using large numbers of sensors for monitoring. Therefore, these sensors can be used for not only monitoring infrastructure but also in other research fields such as environmental monitoring.

The University of California, Berkeley (UCB) and Crossbow Technologies Inc. have made progress in the area of developing low-powered wireless sensors. These wireless sensor modules, called MICA motes, are low-powered, autonomous and have the capability of forming ad-hoc networks. To run applications on the MICA motes, the motes use TinyOS, which was developed by Davis Culler in University of California, Berkeley. TinyOS is a component-based runtime environment designed to provide support for deeply embedded systems that require

concurrently intensive operations while constrained by minimal hardware resources.

Figure 6. MICA mote

The MICA motes (see Figure 6) have eight channels that convert analog to digital, and each of these channels has a resolution of 10 bits. The RF transceiver on the mote is installed with 916Mhz or 433Mhz frequencies. The maximum data rate of this mote is 40Kbps and its maximum range is 100 ft. Another feature of MICA is that the mote comes with a sensor board containing a sensor such as an accelerometer, temperature sensor, or light sensor. Another product, the i-Bean, is similar to MICA. The i-Bean was developed by Millennial Net Inc, having capabilities similar to the MICA mote. The i-Bean also has ad-hoc networking capabilities and a maximum range of 100 feet. The only difference is that i-Bean is just an RF module without any sensors. Even though the i-Bean has a higher data rate of 115 Kbps, sensing is one of the most essential components of the monitoring infrastructure system. Therefore MICA is the better choice.

However, MICA has some limitations as the choice for wireless sensors networks. First of all, the MICA mote does not have enough resolution of ADC to detect very low vibrations such as ambient traffic disturbance. Thus, we might fail to detect every single vibration. Formally the MICA motes use the inbuilt ADC of the microcontroller to perform analog-to-digital conversion. This inbuilt A/D converter has a 10-bit resolution, and there might be lack of resolution for monitoring applications. Another disadvantage of MICA is that this mote does not come with a generic sensor board. MICA should be matched to each sensor's application such as a camera, a humidity sensor, or a pressure sensor. Our work would require development of sensor boards for our application.

3.3 Applications

3.3.1 TinyOS

As mentioned earlier, the MICA motes run an operating system called TinyOS, which enables the users to customize the mote using NesC programming language. TinyOS is a component-based runtime environment designed to provide support for deeply embedded systems which require concurrency intensive operations while constrained by minimal hardware resources. The latest version, TinyOS Ver.1.0.0, is written in NesC. The NesC language is primarily written for embedded systems like the MICA motes. It has a C-like syntax. The available primitive data types are uint8_t, uint16_t, uint32_t, and double.

A TinyOS application is built from different components and compiled to form the final executable file, which is installed on to the micro controller. Components are the basic building blocks of a TinyOS application. There are three types of components: module, configurations, and interface. Here is an explanation of the three TinyOS Application components.

1. Module: A module is a NesC component consisting of application code in a C-like syntax.
2. Configurations: A configuration is a component that " wires" other components together. The idea here is that one can build an application as a set of modules, wiring together those modules by providing a configuration. Every NesC application has a single top-level configuration that specifies the set of components in the application and how they invoke one another.
3. Interface: An interface is used to provide an abstract definition of the interaction of two components (the user and the provider). Similar to a languages like Java, an interface contains no actual code or wiring; it simply declares a set of functions that the interface's provider must implement (called commands), and another set of functions that the interface's users must implement (called events). In this way, it is different to Java interfaces. A java interface specifies one direction of call, while NesC interfaces specifies two directions. For a component to call the commands in an interface, it must implement the events of the interface. Any single component may require or provide multiple

interfaces and multiple instances of the same interface.

For readers who want to know more about TinyOS, details are available on the web site: http://today.cs.berkeley.edu/tos..

3.3.2 OscilloscopeRF, GeneralBase, and SerialForwarder

TinyOS also provides some basic applications to access the sensor data. We describe OscilloscopeRF, GeneralBase and SerialForwarder.

The OscilloscopeRF application, which is installed on the MICA motes, reads data from the ADC (Analog to Digital Converter), pocketsizes the data, and sends a radio packet (ISM band, 916 MHz). The ADC reads analog data from the accelerometer every 100 millisecond. Therefore, the mote can send 10 sets of readings in each radio packet. Actually, the 10-set of readings is the default for the radio packet. Originally, the OscilloscopeRF program was written to read data from the photo sensor, but it was modified to read data from the accelerometer. As mentioned before, MICA motes are programmable, so we can modify the behavior of the motes in ways that we desire. For example, the motes can be programmed to send the vibration information only when that exceeds a certain threshold. The base station mote, which has the GeneralBase code, will receive data from the motes that have OscilloscopeRF installed. GeneralBase also sends data to the base station through the serial port.

The data from the serial port is delivered to a computer, which is connected to the base station mote, and the SerialForward application forwards the data to the TCP (port number 9000) port. The computer can now connect to a central database through the Internet using 802.11b. Web services can be written that sift through the information in the database and present them in the desired format. A front end should collect data from the web services and provide information to web based clients connect to this application.

3.4 Software Architecture

We identified 5 major actors: Sensors, server, clients (or the end users), base station, and the database. Figure 7 is UML (Unified Modeling Language) for this project.

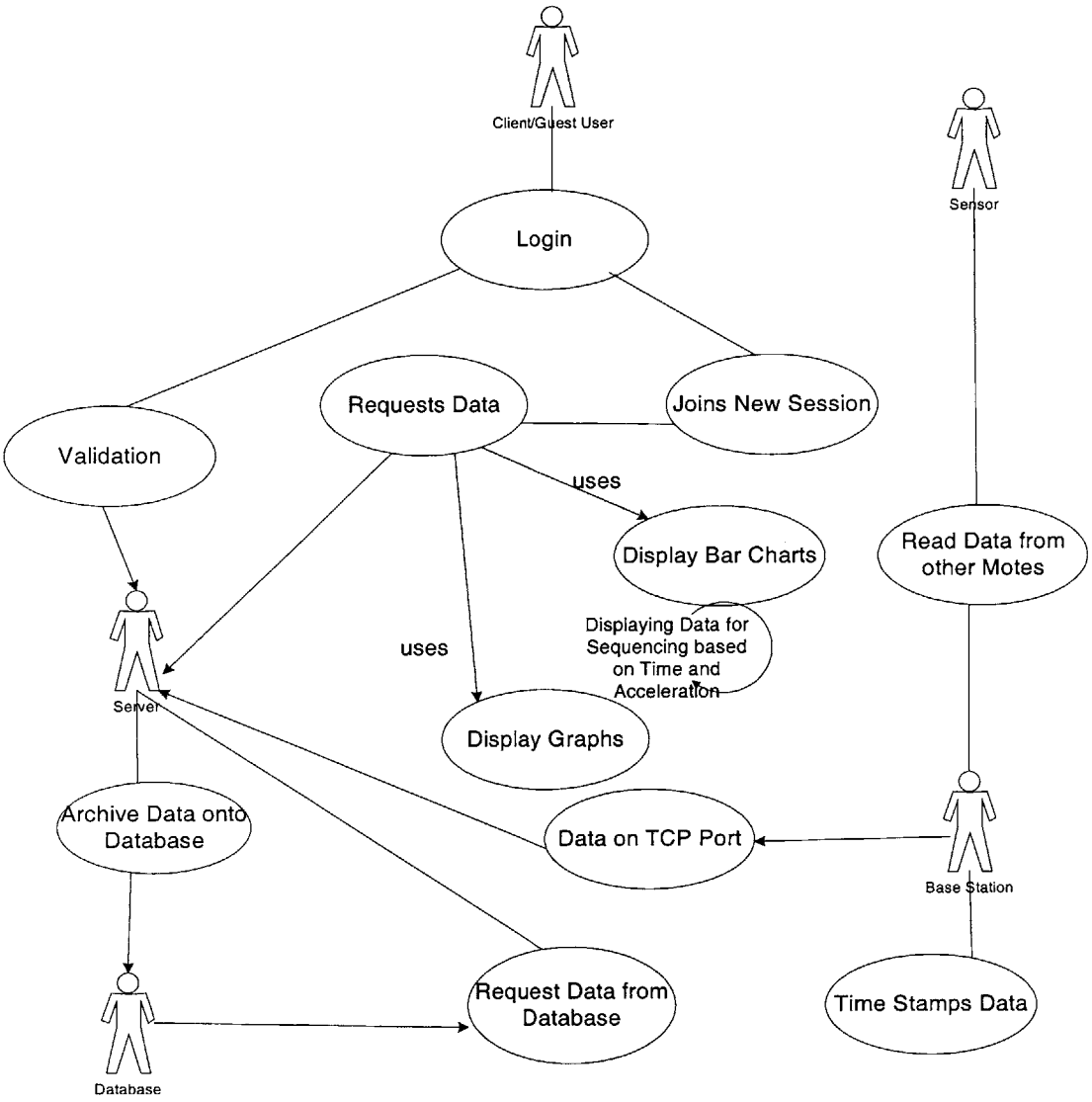

Figure 7. Unified Modeling Language

The data that the sensors collect are forwarded to the base station. The logic of this connectivity is described in Figure 5. The base station, one of the motes, is connected to the computer through a serial port connection. Through the program, SerialForwarder, a TCP server is run in this base station ready to serve real data to clients. A client connects through a

direct TCP/IP socket connection and forwards the data to a central Web Server. The central Web server has an application that constantly archives the data to a in SQL Server database. If any authenticated user requests real time data, then a new thread is launched and publishes real time data through a web service. The input to the web service is an array of data readings. When an authorized user requests some archived data through the web UI, a new thread should be spawned off (treating the database like a client). The data base retrieval is done using Stored SQL procedures, for faster performance

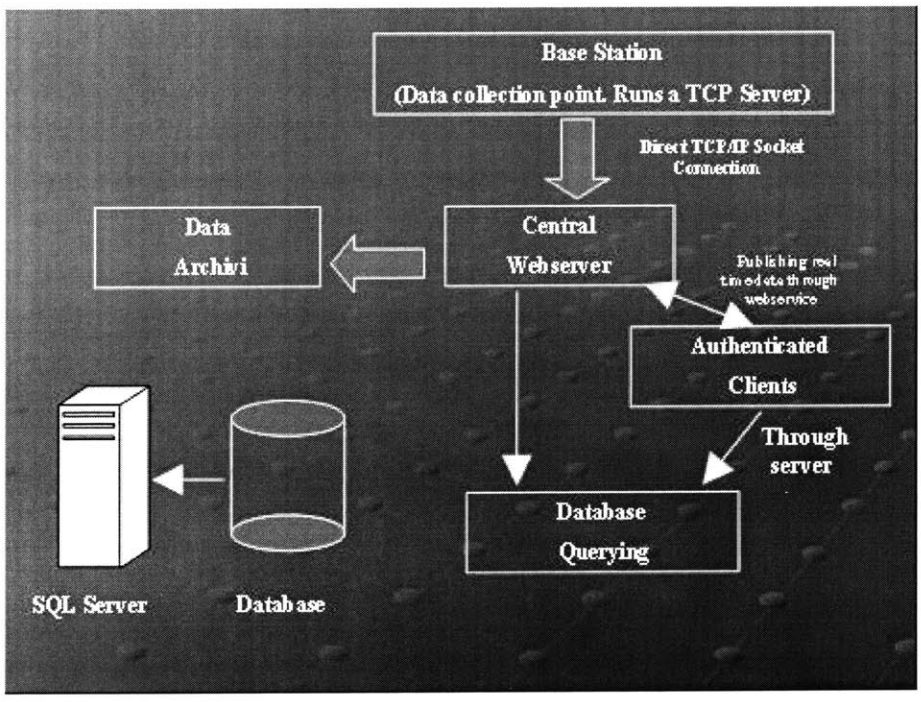

Figure 8. Software Architecture for the Wireless Sensor Project

Even though the code was originally written for the 'Wireless Sensor Based

Infrastructure Monitoring' project for the Parsons Lab, we modified the code considerably to suit our project requirements. The projects and classes that we have defined are listed below:

- **TCPPortReader**: This is a class library that is used to read data from the sensor base station, via a direct socket TCP/IP connection. It also contains a class called 'ParsedDataReadings', which is the fundamental class in which we store all our data values. Besides the acceleration and time stamp, this class contains information about the data such as source mote ID, channel no., and a counter used by the mote to measure time. This project also contains code for inserting the data. Though the sensor returns data every 100ms, we don't require that level of accuracy for the purposes of our project. The data from the sensor itself comes in sets of 10; so we only retain the maximum value of those 10 readings (which is the maximum value in 1 second) and store it in the database and publish it in real-time. The data is processed only once in 10 seconds.

- **TestTCPPort**: This is a console application that is written to test if the data was read correctly from the TCP/IP port. It is a good alternative incase the web server is down.

- **SensorDataProvider:** This is a web service that takes the data from the TCPPortReader. TCPClientReadData method (which returns an array list of parsed data reading objects) and publishes it real time. Even though a web service is slow compared to a remote socket connection or a direct socket connection, we decided to go with a web service since this way

we could serve multiple clients, without having to manage multiple client connections ourselves (using multi-threading), each time someone connected and disconnected from the service. Again, to avoid multi-threading, we are only storing values in the database when at least one client is connected to the web service (the data goes to the database first before being published).

- **RealTimeSensorDataUI**: This is a web application that is used to provide the user interface for real-time data publishing (SensorDataProvider) web service. The SensorDataProvider web service renders the data to be published to the webform. We use user.xml to provide authentication.

- **ProceduralSQLQuery**: This is a web application that contains all the queries that are to be made to the database. One is to find the acceleration at a given time and the other is to find all the acceleration values, which are greater than a user-specified one. We have included a BarChartUI.aspx also, which plots a bar chart of all the accelerations. The SQL queries are done by means of stored procedures.

- **GraphService**: This is a web service that plots Excel graphs. The web service outputs an image, which is then published onto the screen.

3.5 Web User Interface

The web user interface is used to provide 5 different functionalities:

1. Observing Real Time Sensor Data (text)

2. Observing Graphical Real Time Data (acceleration vs. time)

3. Past Acceleration Value Query

4. Maximum Acceleration Query

5. Web Chart Plotting of Acceleration

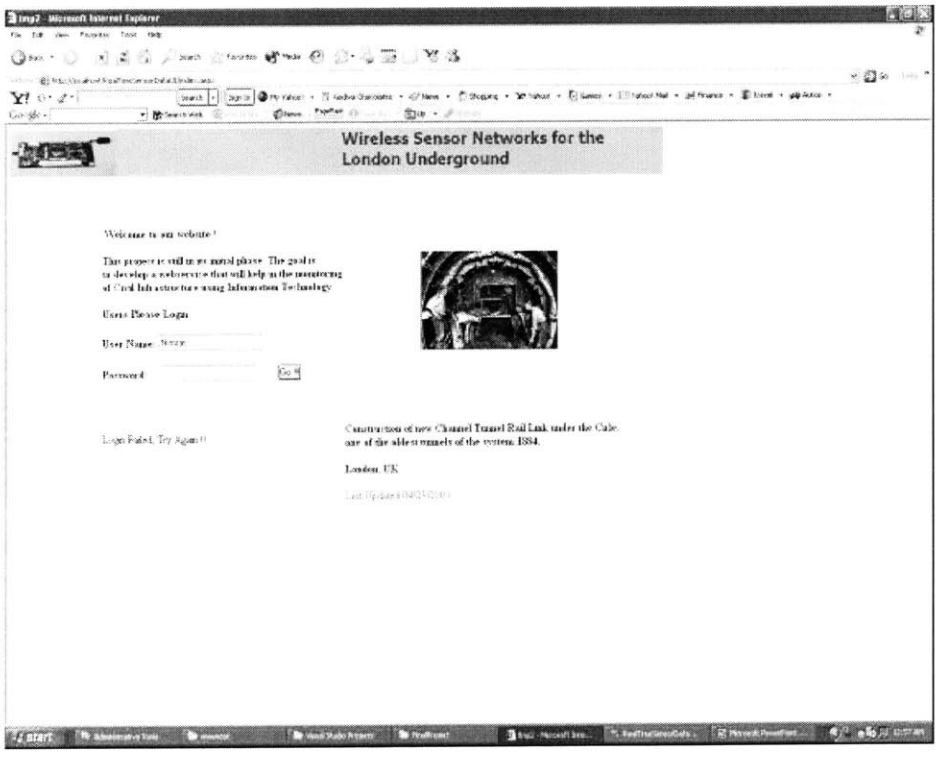

Figure 9. Homepage of web UI

For security reasons, we provide user authentication before allowing the user to access

real time data. Fig 9 is a snapshot of the home page and it asks for authorized users to login. When the user enters a username and password in the corresponding fields on the website, the system checks for permissions. An authorized user can see a menu bar on the left side, which contains links to various parts of the web site (see Figure 10).

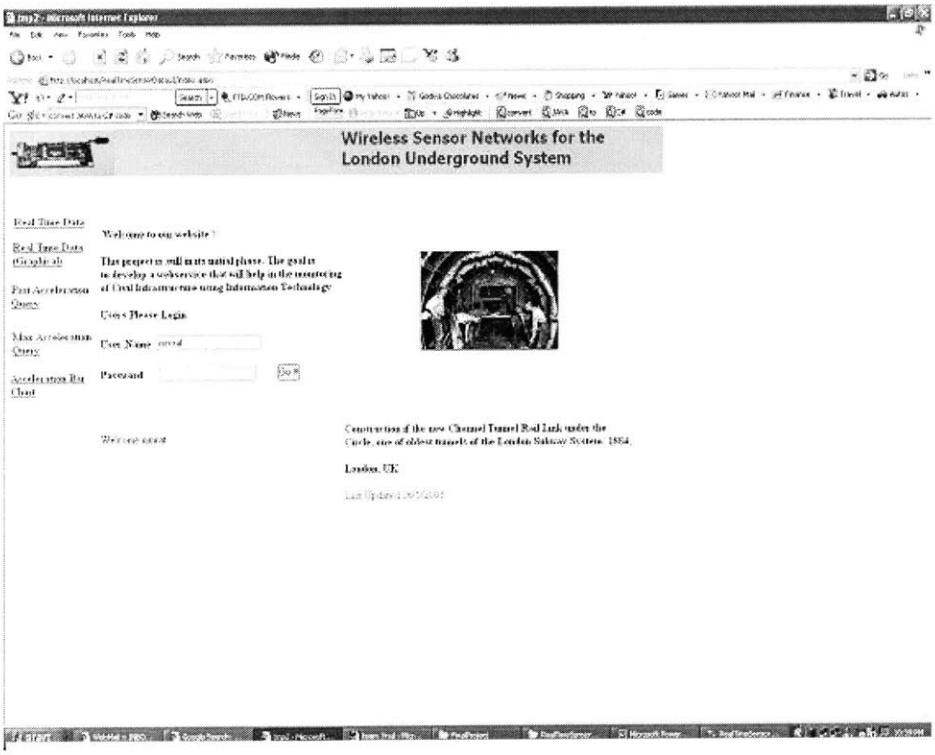

Figure 10. Home page of web UI after successful Log-on

- Observing Real Time Sensor Data (Text): Users can access real time sensor data by clicking on the 'Real Time Data' link on the menu. It takes a while to connect to the web service that publishes real time sensor data over the web. Another thing to note here is that the mote (sensor) has to be running at the time the data is accessed. Figure11 is a snapshot of the page when connected to the service. The page is refreshed every 10 seconds, displaying new

sets of 10, 1 second readings.

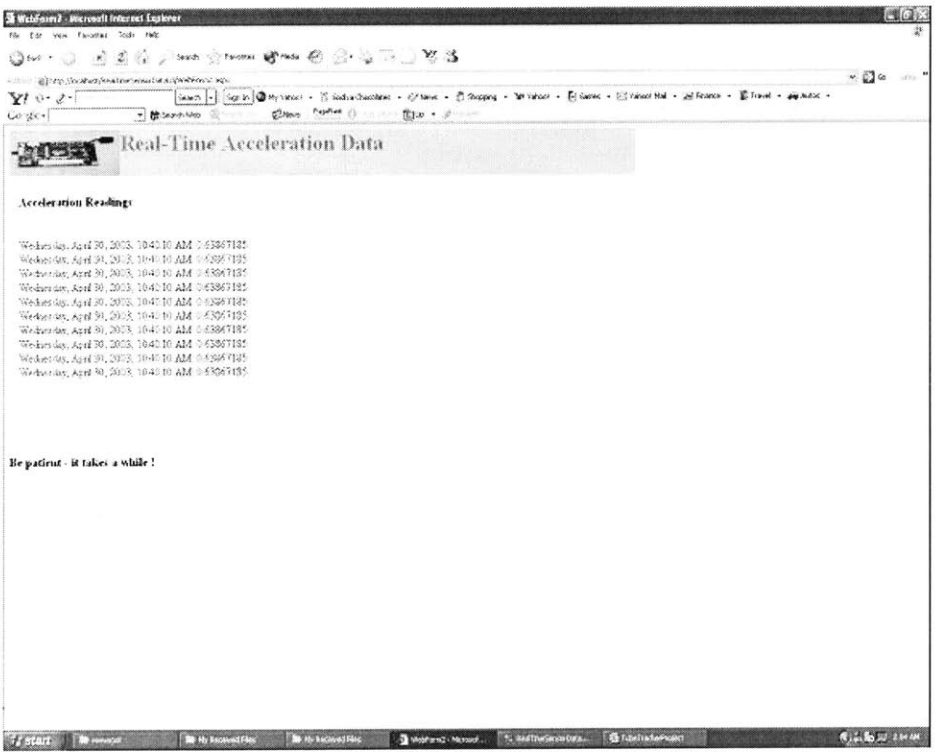

Figure 11. Real-Time Acceleration Data page of web UI

• Observing Real Time Data (Graphical): Basically, users see graphical plots of the same real time data using this functionality. This application makes use of Graph web service to plot the stream of data and then renders the new image once every 10 seconds to the browser.

• Past Acceleration Query: The user interface allows users to query the database that contains one-second readings of acceleration data. Users can look at the acceleration value at a particular instant in the past. The users need to submit the date (format: month/day/year e.g. 30/12/2003) and time (format: Hr:Min:Sec e.g. 11:56:34 PM). This need not be a complete

time, for e.g. 11:56 will return all the readings starting with 11:56). The application returns the results based on a stored SQL Query (stored procedure) running over a SQL server database. The corresponding data is displayed in a data grid as shown in Figure 9.

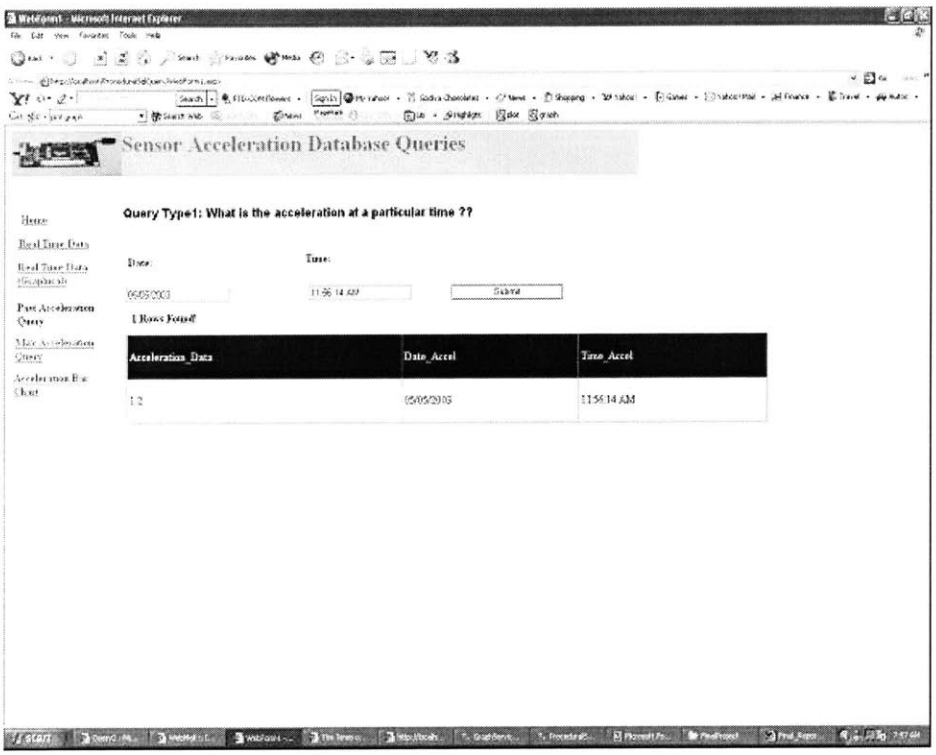

Figure 12. Sensor Acceleration Database Queries page of web UI

- Maximum Acceleration Query: Another query feature allows the user to find the acceleration readings greater than a particular user-specified value. It asks users to choose the minimum value over which acceleration values are required. Once the user enters the value, the stored SQL procedure finds out all the values greater than that value from the database and displays them in a data grid as shown in Figure 13.

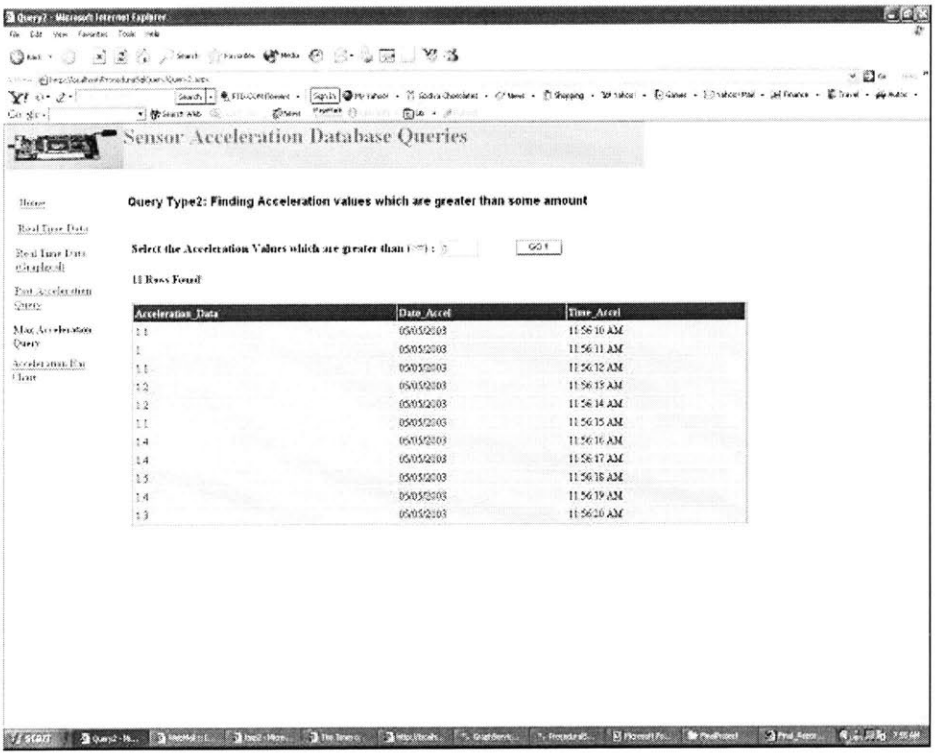

Figure 13. Sensor Acceleration Database Queries page of web UI

• Dynamic Acceleration Bar Chart: Currently, for the purposes of demonstration, only the first few values in the database are taken and plotted in the bar chart format as shown in Figure 14. But we plan to extend it to make it a dynamically generated web chart where users can choose an arbitrary, valid time zone on a particular day. Then the user should be able to visualize the acceleration data in that time through a bar chart. We use the GDI+ classes of the .NET library for plotting the bar charts.

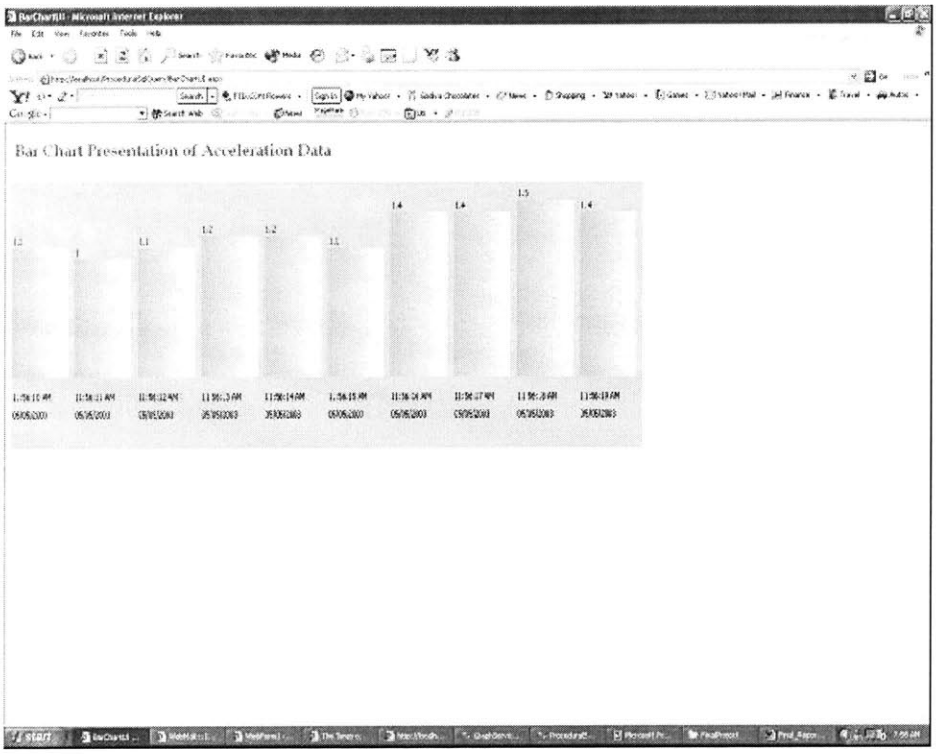

Figure 14. Bar Chart Presentation of Acceleration Data page of web UI

Chapter 4- Implementing a Base Station

The previous chapter described the system design and the software architecture for a wireless sensor network project. This chapter describes another crucial component of a wireless sensor network, the implementation of a base station. We considered two options: a laptop computer and a PDA, the iPaq 3955. During this study, there was a project that used the MICA motes to monitor a building on the MIT campus that housed sensitive experimental equipment. Construction activity near the building had been disrupting the sensitive alignment of these instruments. This project used a laptop PC as a platform for the base station because the building had enough power outlets and is more accessible than the tunnel environment. Obviously, a PC has more resources and available software than a PDA, so implementing a base station with a laptop PC was considered a better choice in this case. However, a wireless sensor network system might need to be installed in a less friendly environment such as the tunnel mentioned earlier. Over 3 million passengers use the London Underground system a day, and 150,000 people enter the system every hour. On the top of that, the London Underground system cannot guarantee enough power outlets for the sensor network because the infrastructure was designed only for transportation. Since a PDA is more compact and can run on batteries for reasonable but short periods of time, e.g., hours to 1-2 days, a PDA might be a possible solution to some sensor networks.

However, implementing a PDA base station is not simply replacing the laptop computer with a smaller computer. Even though PDAs are becoming increasingly powerful nowadays, as so-called pocket PCs, their actual technical capacity is much lower than real PCs, and software for PDAs is still more constrained. Imperfect software and the small storage capacity of PDAs are the main obstacles to implementing the base station with a PDA platform.

4.1. The iPaq 3955 as a PDA Platform of a Base Station

The iPaq 3955 model, available since 2002, is manufactured by Compaq. This machine has 32 MB ROM and 64 MB SDRAM, but it is still possible to extend memory using a Compact Flash card or secure digital slot. The iPaq 3955 runs MicroSoft Pocket PC 2002 as the Operating System and contains 400 MHz Intel X-Scale Processor. As with many other pocket PCs, the iPaq 3955 uses a 1400 mAh lithium polymer rechargeable battery, which usually lasts 12 hours. Its width is 3.3 inches, height is 5.3 inches and depth is 0.6 inches (see Figure 14). The iPaq 3955 weighs 6.5 ounces; that is less than one third of a bottle of water. This pocket PC connects to PCs using Microsoft ActiveSync 3.5 software. A PC that is synchronized with the iPaq 3955 should have a 486/33 MHz or higher processor and use Windows 98, NT, 2000 or XP Operation Systems.

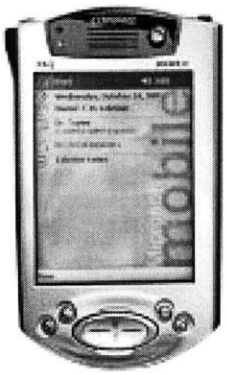

Figure 15. iPaq 3955

The OS on the iPaq, Pocket PC OS, can be compared with the Palm OS. The top-end processor now available for the Pocket PC is the Intel StrongARM or X-Scale processor, which runs at 400 MHz, whereas the top-end Palm OS runs with a Texas Instruments OMAP 310 processor, which runs at 144 MHz. We note that the Pocket PC OS was designed for a more generic processor and hardware combination and uses substantially more of the processor for basic overhead. The extra power available for the Pocket PC also correlates to faster battery drain on the Pocket PC devices. In general, a Pocket PC can do more than a Palm OS processor, but at the cost of battery life.

In terms of memory, the pocket PC operating system is also designed to handle larger memory allocations than the Palm. What this means is that users can work with larger files and more sophisticated programs requiring more memory. Additionally, in the area of memory efficiency, the Palm makes very effective use of its memory as far as applications go, whereas Pocket PC and Pocket PC applications are often larger. However, when it comes to raw data, the

usage is essentially the same.

Convenient expansion is another advantage of the Pocket PC. The Pocket PC can be expanded through one of four standards (or any combination) depending on the hardware options. The most common expansion technique is through CompactFlash (CF) cards. The iPaq uses an expansion sleeve to support CF cards. The second option for expanding an iPaq is through the PCMICA port. This port is a PC card standard commonly used in laptop computers. This fact means that Pocket PC OS is compatible with any kind of PCMICA cards that a laptop computer uses. The third option is to use Secure Digital (SD) cards. This expansion format allows for a postage stamp-size card to be inserted into iPaq to provide a wide range of functionality, such as external storage. The fourth option is the Bluetooth standard. The combination of CompactFlash, PCMICA, Secure Digital, and Bluetooth means that there is a tremendous range of expansion and connectivity options on the iPaq.

After comparing general features of Pocket PCs and Palms, Pocket PC seems to be a better OS for a PDA base station platform. Among Pocket PC devices, the iPaq 3955 is a powerful PDA choice with a reasonable price, around $500.

4.2 PDA vs. Laptop

To decide on a device for implementing a base station, it is necessary to review the

environment of the project and characteristics of possible devices. As mentioned, the two most convenient and efficient choices for the wireless sensor project in this thesis were a PDA (iPaq 3955) and a laptop PC. Both devices have pros and cons as a base station platform, but a laptop PC was finally chosen mainly because there are more stable software than pocket PCs, and also a laptop PC has superior capacity and data processing speed compared to a pocket PC.

While the PDA has advantages, such as lower-power consumption, small size, and a low price, a laptop computer has much faster and more stable data processing ability and most of the applications needed for the project are designed for a laptop PC, not for a pocket PC. Even though an iPaq consumes less power than a laptop PC, the iPaq loses all data when it is discharged and there is no built-in backup memory as a default. Therefore, if we fail to charge an iPaq regularly, data in the iPaq might be lost and this is very dangerous, particularly for a project where safety is the main concern. As mentioned, the other problem using an iPaq is there is not enough software support to operate the base station application. Microsoft released a beta version of .Net for the Pocket PC OS, but that version still has unstable behavior, producing errors and problems on the Pocket PC. Another factor is price. The iPaq's unit price is definitely lower than an average unit price of laptop PCs, but again this is directly proportional to the resources available to handle sensor data. If we try to find a laptop PC that has the same ability as the iPaq 3955, it can be as cheap as the iPaq 3955, or even cheaper because a laptop PC does

not need to be as compact as the iPaq. Therefore, we conclude that price is not an actual advantage of the iPaq.

For these reasons, using a laptop PC seemed to be a better choice than an iPaq as a platform for the base station. However, it is obvious that PDA has its own advantages as a base station platform, so further research will be helpful for the future of wireless sensor networks.

Chapter 5- Conclusions and Future Work

Using wireless sensor networks, we are able to avoid the high cost for labor, time-consuming maintenance and exposing humans to hazards that are associated with traditional ways of monitoring infrastructure. However, these advantages can be achieved only when we use an appropriate communication standard and hardware that match the purpose of the monitoring task and environment. For the case study of the wireless sensor network project in this thesis, we recommended the use of 802.11b as a communication method and a laptop PC as a base station computer. While 802.11b was not actually applied in the real monitoring project by CMI, this decision was primarily due to time constraints. Further tunnel monitoring tasks will reconsider this choice.

This project gives us a better understanding of the issues and directions for future research in wireless sensor networks. One of the interesting future research areas in monitoring infrastructure is the use of optical communication. Optical communication has some advantages over RF communication. Optical communication is usually done using a laser light. Also, by using 3D laser scanning principles, one can actually scan a structure and construct 3D profile. Laser light can be used for multiple purposes in performing wireless monitoring

- as a means of getting the 3D profile of the structure
- as a means of optical communication

- as a source of energy to power MEMS devices and
- as a means of interrogating photonic crystals

By proceeding in this direction, one can obtain a lower power-consuming and more efficient wireless sensor system.

Reference

Chaplin, Kevin, "Wireless LANs vs. Wireless WANs", Sierra Wireless, Inc., <http://www.sierrawireless.com/news/docs/2130273_WWAN_v_WLAN.pdf>, November, 18,2002

Cheekiralla, Sivaram, "Wireless Infrastructural Monitoring" MIT Project progress Report, 2003.

Cheekiralla, Sivaram et al. "Wireless Sensor based Infrastructural Monitoring." Project Report for Course 1.961 at MIT, Fall 2002.

Dellarocas, Chrysanthos, "Wireless Communications", MIT 15.564 Information Technology 1 Lecture Notes, 2003.

DeeSimone, Antonio and Nanda, Sanjiv, "Wireless data: Systems, standards, services", Wireless Networks Journal, 241-256, 1, 1995.

Dornan, Andy, "the Essential Guide to Wireless Communications Applications", Second edition, 2002.

Farserotu, J., "Integration of Terrestrial and Satellite Networks: Technology, Service and Applications", Wireless Personal Communications Journal, 17:283-290

Fisher, Josh and Wang, Rosemary, "Feature- Wireless Wide Area Networks", <http://www.pdafn.com/vertical/features/wireless_4xml>, March, 13, 2003.

Greene, David C., "Sensor Technology and Applications to a Real-Time Monitoring System" M.Eng Thesis. *In Department of Civil and Environmental Engineering, Massachusetts Institute of Technology,* 2000.

Krenik, William, "Wireless User Perspective in the United States", Wireless Personal Communications Journal, 22:153-160, 2002.

Lawson, Stephen, "Wireless visions collide", InforWorld, <http://www.inforworld.com/article/03/03/18/Hncitiakeynotes_1.html?wireless>, March, 18, 2003.

Lin, Yi-Bing, Rao, Herman and Chlamtac, Imrich, "General Packet Radio Service (GPRS): architecture, interfaces, and deployment", Wireless Communications and Mobile Computing Journal, 1:77-92, 2001

Morris, Jeff, "Guide to Wireless Data Networks", Sierra Wireless, Inc, <http://www.sierrawireless.com/news/docs/wirenet.doc>, July, 1997.

Rahman, Mohammad and Imai,Hideki, "Security in Wireless Communication", Wireless Personal Communications Journal, 22:213-228, 2002.

Rappaport, Theodore S., "Wireless Communications: Principles and Practice" Second Edition, 2002

Sinclair, Ian R., "Sensors and Transducers" 3rd edition, 2001.

Official website for Bluetooth standard.
<http://www.bluetooth.org>

Official website for WINS Project at UCLA,
<http://www.janet.ucla.edu/WINS>

Official website of Crossbow Technology Inc.,
<http://www.xbow.com>

Official website for TinyOS,
<http://webs.cs.berkeley.edu/tos>.