This is to certify that the

thesis entitled

WEBOTS: WEB BASED OBJECT
TRACKING SYSTEM

presented by

Anthony R. Lambert

has been accepted towards fulfillment
of the requirements for

M.S. degree in ~~Computer~~ Science
and Engineering

Major professor

Date 11/25/2002

WEBOTS: WEB BASED OBJECT TRACKING SYSTEM

By

Anthony Lambert

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science and Engineering

2002

ABSTRACT

WEBOTS: WEB BASED OBJECT TRACKING SYSTEM

By

Anthony Lambert

Experts in industry are predicting that wireless, mobile, Internet enabled devices will outnumber desktop machines in the future. This new mobility paradigm has developed a growing interest in location-aware systems and services. Many systems have been developed for the purpose of location tracking each with their advantages and disadvantages.

The main focus of this thesis is the development of a low cost, real-time object tracking system, called Web Based Object Tracking System (WeBOTS), that uses off the shelf, commodity products to perform coarse grained location tracking. The system uses Radio Frequency Identification (RFID) and Bluetooth technologies to obtain location information. The location information can then be viewed anywhere using a web browser.

This thesis investigates RFID and Bluetooth to see if they are feasible for obtaining location information. Research of the technologies and an investigation for a ways to exploit their features for location tracking. Furthermore, the thesis discusses some features missing from the technologies that would make it better for location tracking.

To my wife and family

ACKNOWLEDGEMENTS

I would like to take the time to thank my advisor, Dr. Linoel Ni for taking me on as a student and for his guidance along the way. I would also like to acknowledge my other commitee members, Dr. Matt Mutka and Dr. Abdol-Hossein Esfahanian. Finally I would like to thank Abhishek Patil for his assistance during the course of this work.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

Experts in industry are predicting that wireless, mobile, Internet enabled devices will outnumber desktop machines in the future [23]. This new mobility paradigm has developed a growing interest in location-aware systems and services. In today's world, location tracking has many purposes. Security, advertising, inventory management, and entertainment are just a few of the potential areas that the application of location tracking technologies could improve. These location based services are expected to be context aware, personalized, unobtrusive, and be able to adapt to the users changing location. [23].

These requirements on location aware applications have fostered a large amount of research over the past few years and more and more applications are starting to realize the benefits that location awareness can provide. With the large amount of applications for location tracking, naturally there have been many techniques for accomplishing it. The techniques employed to track the location of objects depends largely on the application in question and the degree of accuracy desired.

The degree of accuracy needed depends largely on the application, as some applications require strict accuracy and others require much less. An example of such an application is real-time augmented vision applications. Most real-time augmented vision (AV) applications require a very high degree of accuracy as to where a user or object is located. AV deals with lining up real objects with virtual objects and the human visual system is very sensitive to even the slightest amount of misalignment.

Therefore the degree of accuracy in location tracking for AV applications has to be within a few millimeters. Many different techniques, all with their own advantages and disadvantages, are used to get this high degree of accuracy including computer vision, magnetic tracking, and sonar based methods.

The techniques used to track objects to within a few millimeters of their actual location are very complex and usually not highly scalable. The main reasons for the lack of scalability are the size of the area they are able to track in and the amount of processing power used in determining the exact object location. If the amount of accuracy is relaxed a new set of applications comes to the fore. These applications need accuracy within a few meters. Applications such as these could include location based advertisements as a user walks through a grocery store. When the user walks by the aisle that contains diapers, an advertisement could display a sale price on a LCD on the user's cart. The location accuracy needs only be a few meters near the aisle in question.

Many other applications can get away with much coarser grained location information however. In a hospital, nurses often need to find a particular doctor. Currently a nurse attempts to locate the doctor by paging them using a numeric pager. However, with a location tracking system, a nurse could go to a computer terminal and click on the doctor's name; instantly knowing what room they were in or if they were in the hospital at all. Here the accuracy of the location information is at a very coarse level, on the order of the size of a room.

This level of accuracy is also well suited to security applications. A system could log the location of each user. In the event of a problem, an administrator could

then look up who was in a certain location around a specific time.

Other applications need can get away with even less granularity in their location information. Cellular phones are an example of one such application. "Location management in the cellular architecture can be viewed as addressing the problem of providing uncertainty bounds for each mobile user. The geographic bounds of the cell constitute the uncertainty bounds for the user. Uncertainty at the cell-granularity is sufficient for the purpose of calling a mobile user or sending him/her a message." [25].

These applications are becoming more and more necessary in today's society. This thesis will allow for an in depth study of location tracking systems. Different techniques and technologies will be investigated as potential solutions to the problem of location tracking and the prototype of a system will be built to test these.

1.2 Problem Statement and Objectives

In order to introduce the problem a sample scenario will be presented. In a hospital there are many people coming and going all over the place at all times. Nurses, patients, doctors, family members, maintenance staff, and many others roam all over the hospital 24 hours a day. In this chaotic environment it can become very difficult to find a particular person especially in a large urban hospital with many floors and wings.

Certain times during a shift a nurse needs to find and talk to a particular doctor for instructions on the care of a patient. It can be very difficult to find a doctor especially if they do not answer their pages. If a nurse could walk up to a

terminal, click on a doctor's name, and immediately be told where that doctor was, they would be able to provide better, more prompt care for their patient.

The overriding objective of this thesis is to build a low cost object tracking solution that could solve the situaiton presented in the above scenario. The system should use low cost off the shelf, commodity products and leverage existing technologies. The system should make use of existing ubiquitous networking infrastructure and protocols to make implementation cheaper and quicker. The information contained in the system should be easily accessable from a variety of devices everywhere.

This project focuses on coarse grained location information on the scale of which room a person is in, within an entire building. The primary objective to be able to walk up to a web enabled terminal, click on a person's or object's name, and have Web Based Object Tracking System (WeBOTS) tell you what room they are located in. In order to monitor the objects a web based framework for real-time location tracking will be built. The system should be able to employ different techniques for getting location information. Therefore the method of gathering location information is abstracted away from the system.

The system aims to use commodity off the shelf products that anyone can go to the store and purchase. Initially the system will be implemented using Radio Frequency Identification (RFID) as the method of obtaining location information. RFID was not designed to be used for location tracking, so these issues will have to be dealt with along the way. RFID will be used to develop the server and database that everything is run off of and to work out any design flaws. Once the system matures, a new method of obtaining location information will be developed to work

concurrently with RFID. The new method will be Bluetooth.

Another objective of the thesis is to investigate the use of the Bluetooth technology for the purposes of location tracking. Bluetooth is another commodity technology that promises to be as common in computing as a CDROM drive. Like RFID, the specifications for Bluetooth do not explicitly contain provisions for the purpose of location tracking. Therefore a main objective of this paper is to investigate its feasibility as a location tracking paradigm. As Bluetooth grows in popularity and invades offices everywhere, a great location framework will already be built into office environments. Bluetooth devices on the certain non-mobile machines could probe for any devices that come near it. This information could then be sent to a server that can look up the ID of the device, and then know where a user associated with that device is.

Location tracking can quickly produce large amounts of data, but the system needs to remain quick and responsive. Location information for many objects will quickly fill up a database resulting in a degradation of performance. The data in the database is still important, however, so just deleting the data is not an option. This thesis will invetigate a method of archiving the data to keep things running smoothly.

Another way to improve the performance of the database server is to limit the number of queries to the database that are required. This thesis will attempt to build a system that limits the amount of queries to the database. Some of the processing will be pulled out of the server application and pushed to the clients, lightening the strain on the server and therefore improving speed.

1.3 Thesis Organization

The organization of the rest of this document is as follows, Chapter 2 contains a discussion of other projects and research in the location tracking arena. It will focus more on those projects that are closely related to this paper. Then, Chapter 3 will discuss the two technologies, RFID and bluetooth, in detail. The following section, namely Chapter 4, will contain a complete breakdown of the Web Based Object Tracking System (WeBOTS). Chapter 5 moves into a discussion of the experiments performed with the WeBOTS system and the results obtained. The paper then contains some conclusions and future work in Chapter 6. Finally, the thesis finishes up with an Appendix that explains how to get the WeBOTS system up and running.

2 Related Works

This section discusses various projects whose focus was obtaining location information. Depending on the application, there has been a large number of techniques employed for the determination of location that includes computer vision, magnetic tracking, infrared (IR), and radio frequency (RF). This section will focus on the latter two techniques as well as the Bluetooth technology. Lastly, there is a discussion of different techniques for improving location database performance. Before launcing in to those descriptions, however, there will be a discusison of terminology used in location awareness applications.

2.1 Location Sensing Techniques

There are three main methods to achieve location tracking. The first method uses triangulation methods. Here the object to be tracked is located in the space interesected by multiple regions. The distances of all the region centers to the object can be used to calculate the exact location of the object. Figure 2.1 shows an example with triangulation using three regions.

Another primary method of garnering location information is using scene analysis. The most intuitive way of scene analysis is using computer vision techniques to get information from a bitmap taken from a camera. The information discovered by the computer vision algorithm can then be used to compare against a location database to determine location. However, scene analysis techniques do not neccessarily have to deal with computer vision techniques and bitmaps. Later the RADAR

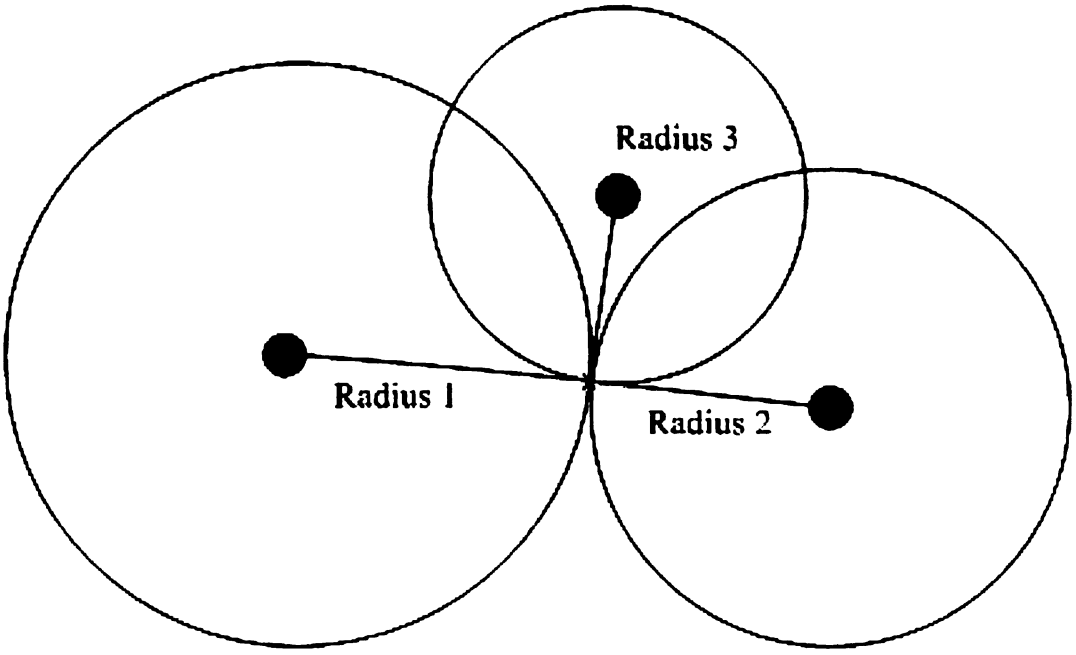

Figure 2.1: Triangulation: The object's location is determined by finding the 3 radii that all interesect at exactly the object's point.

project will be talked about that uses scene analysis techniques by using measurements of physical phenomena instead [12].

The third type of location sensing technique is proximity. With proximity, there is a known point and when an object gets near that point, the object is said to be at that known point. To determine if the object is at a known point three main methods are used: detecting physical contact, monitoring wireless cellular access points, and observing automatic ID systems. [17]

An example of detecting physical contact would be the touchpad on a laptop. A sensor detects where a user's finger is located on the touchpad. Then, when the finger moves across the pad, the touchpad sends relative displacement measurements

to the operating system of the laptop. Therefore, the touchpad device tracks the location of the users finger and mirrors that location on the screen.

Wireless cellular based proximity sensing does not require any physical contact. An example of a wireless cellular based system would be a cellular phone network. A cellular phone network is comprised of a set of stations (antennas) that cellular phones transmit data between. In highly urban areas there can be many of these antennas spread throughout the environment. The user's location can be set to the antenna they are closest to. The accuracy of the system depends on the number of antennas in the environment and how close together those antennas are.

Some cellular systems are becoming more powerful and advanced in their techniques for location tracking. They are starting to use triangulation methods based on signal strength information between the cellular phones and the antennas to more accurately pinpoint the user's location.

The third method of proximity based sensing, that of oberserving automatic ID systems works based on some activity of everyday life. For instance, a user's location could be tracked based on where they charge their credit cards.

Another important classification for these system is what type they are, namely a location tracking, positioning, or sensing systems. A location tracking system will monitor objects in its area without involving the object to be tracked in the computation of its position. Location positioning systems, on the other hand, provide a framework for an object to utilize in the computation of its own location. The third type, location sensing systems, is a hybrid of the tracking and positioning systems [16].

This thesis will focus on proximity based location tracking systems. The RFID

and Bluetooth systems will both rely on proximity based location information. Furthermore, the clients are not involved with the determination of their positions, so it is a location tracking system.

2.2 IR Technologies

Infared (IR) technologies are named after the frequency band they employ for transmitting data wirelessly. The frequency band that IR uses is very near that of visible light on the electromagnetic spectrum. The wavelengths of IR range from a fraction of a micron to almost 1 centimeter. IR signals are very directional and typically cannot operate over great distances without a lot of power to boost the signal. IR is typically used for short range data transmission. A remote control for a television uses IR to send the signal to the television.

2.2.1 HiBall

The University of North Carolina has done a lot of research in location tracking for the purposes of Virtual Reality (VR) and Augmented Reality (AR). [24] came up with an IR based tracking system called the HiBall Tracker that "generates over 2000 head-pose estimates per second with less than one millisecond of latency, and less than 0.5 millimeters and 0.02 degrees of position and orientation noise, everywhere in a 4.5 by 8.5 meter room" [24]. The performance of this is quite good, however there are a few areas where the system fails. First is the scale in which objects can be tracked. It would not be feasable to wire up the ceiling of an entire hospital for use with the system and doctors would not want the obtrusive head gear required for

the system to work.

The HiBall Tracker is a location positioning system. The system works by placing light emitting diodes (LEDs) in a known pattern on the ceiling and using a special LED detector, called a HiBall, on a person's head. The HiBall senses the LEDs on the ceiling and uses this information to compute location using Kalman-based filters to negate the effects of noise.

2.3 RF Technologies

Radio Frequency (RF) technologies are also named for the frequency band they use for data transmission. The wavelength of the RF band is much longer than visible light or IR. The wavelength varies from a few centimeters to multiple kilometers. The frequency range for the RF band starts out in the gigahertz range and finishes up in the kilohertz range. RF technologies are used to broadcast radio stations, television stations, garage door openers, cordless phones, and many other wireless devices used everyday.

2.3.1 RADAR

Another location aware system that uses scene analysis is the RADAR system developed at Microsoft Corporation's Research Laboratory [12]. The objects in the system use 802.11b devices in the area to obtain signal strength (SS) measurements from. There are three base stations that are positioned in such a way as to cover an entire floor of a building. The three regions the base stations operate in overlap one another. A mobile host that wishes to be tracked must periodically transmit special broadcast

packets (termed beacons) to the three base stations. The base stations then store the SS and signal to noise ratio (SNR).

The system has two modes: offline and real-time. In offline mode statistics are gathered for specific locations around the floor. Data for 70 positions about the the floor was stored in a database. Then, in real-time mode, as the mobile host travelled about the floor it transmits its beacon signal. The base stations receive the beacon and get the SS and SNR information. The signal information is then used to do search through the database of positions obtained during the offline phase. The best match from the search is used as the "guess" of the location of the mobile host. The RADAR project has a 50 percentile around 2.37m-2.65m and a 90 percentile around 5.93m-5.97m [12].

The size of the floor that the system worked on was 43.5m by 22.5m. This technique could be extended to work on larger floors, however, given more base stations. This is a good example of a system that tries to utilize technologies that have already made a substantial infiltration into industry. A system such as RADAR has a much better chance of being used because they use devices that already exist in many enviroments so a company would not have to invest in large quantities of new equipment.

The RADAR system is an example of a triangulation and scene analysis location tracking system. The scene analysis component is evident in the measurement of SS and SNR and the triangulation is evident because the system uses signaling information from multiple base stations to pinpoint the objects location.

The project does not take into account what would happen if a large amount of

traffic was present on the floor. They cite it as future work, but for a real system this is of the utmost importance. They also make mention of the fact that the orientation of the user can greatly affect the SS. In one orientation the user blocks the line of sight (LOS) to a particular base station, while in another orientation there is direct LOS. During different times of the day SS information could vary greatly in a hospital floor. Nurses, doctors, patients, family members, friends, and all manner of different medical carts, and devices makes for a very dynamic enviroment and would greatly affect the quality of the "guess" made by the RADAR system.

2.3.2 SpotON

A system that uses RFID called SpotON was developed at the University of Washington. Their system used a combination of commercial and custom hardware to accomplish their goal of location sensing. Since the system is a location sensing system, it uses a hybird of location positioning and tracking techniques.

The hybrid approach is carried out by using a set of smart tags. The smart tags compute their own location which is reminicent of a location positioning system. The smart tags can act as readers determining location information as well as sending out location information to the devices around them. An application server can be attached to one of the smart tags, and can therefore read in location data from all the tags in the area, thus acting as a location tracking system. The system makes use of measured signal strength from surrounding tags and therefore makes the system a scene analysis location sensing application [16].

2.3.3 LANDMARC

The Location Identification Based on Dynamic Active RFID Calibration (LANDMARC) project used RFID as its means of obtaining user location [19]. A main goal of this project is to eliminate the need for more readers to increase the accuracy of the system. In the RADAR project, for instance, to get more accurate location infomation the addition of more base stations is necessary. The inclusion of more base stations typically includes a great cost; either monetarily or in the time to callibrate the new base station.

The LANDMARC used products from RFCode Inc. [11] whose readers cost many times that of the simple RFID tags. In order to get more accurate location information, [19] prepared the space ahead of time by placing a set of statically positioned reference tags, termed landmarks, in the environment. These tags stayed in the same location and were used as a benchmark to compare with a mobile object as it moved through the system. These landmarks are in the same environment as the the tags to be tracked and are therefore subjected to the same interference effects. This method allows the calibration of the system to change dynamically with the changing environment, unlike the static database style of a system like RADAR discussed above.

The LANDMARC system had some very promising results in that the 50 percentile has an error distance of around 1 meter while the maximum error distances are less than 2 meters [20]. The RADAR system used 3 base stations and the median was as high as 2 to 3 meters.

The LANDMARC project can be classified as a location tracking system that uses proximity to compute location information. The mobile tags do not compute their location, the system itself does which makes LANDMARC a location tracking system. Furthermore, the method of determining where a particular tag is uses the proximity of the tag to the readers.

2.3.4 Cricket

The Cricket system, built at Massachusetts Institute of Technology, was designed with several goals in mind: user privacy, decentralized administration, network heterogeneity, and low cost. It uses a combination of RF and Ultrasonic technologies to provide distance information that is used in the location computation. [22] term their system a location-support system rather than a location tracking system. Location-support is synonomous with location positioning system as defined above.

The Cricket system aimed to provide location information within 1 or 2 square feet of the actual location. In order to provide this level of accuracy, beacons are placed along the walls and ceilings of the environment that will be supported by the system. These beacons transmit their location on an RF signal. Concurrently the beacons transmit an ultrasonic signal as well. When a mobile host in the system receives the signals, it computes the distance between itself and the beacon based on the propagation times for the two signals.

The Cricket system is a location positioning system that uses scene analysis. The system provides the tools for the a mobile unit in the system to calculate its own location. The mobile unit then has the choice of where, if anywhere, to send

its location information to. This has two benefits, first it solves the problem of user privacy because the user does not have to send out his/her location. Second, it solves the decentralization issue, because there is no central database or server storing data for the system.

2.4 Bluetooth

Bluetooth is a RF based technology, but has great importance to the rest of the thesis so it has a seperate related works section. Bluetooth was designed as a cable replacement technology. The vision is to one day have all the peripherals of a computer, personal digital assistants (PDAs), digital cameras, etc be interconnected wirelessly. A much more detailed description of the Bluetooh technology can be found in Chapter 3.2.

Bluetooth, like many of the other technologies used, was not designed with location awareness in mind. At the time of this writing no systems were found that utilized Bluetooth as a method of obtaining location information. However [21] investigated the possiblity of using Bluetooth as a means of obtaining location information. This research found that Bluetooth had a lot of features that made it desirable for location tracking.

The research compared 802.11b to Bluetooth by investigating their resistance to interference. During testing it was found that concurrently running 802.11b and Bluetooth devices cause interference to one another, but that Bluetooth was much less effected by it. Furthermore, in order to get accurate location information, a

number of devices are needed to attempt to triangulate the location of the object. Bluetooth showed a lot of resistance to interference when multiple Bluetooth devices were placed within a close proximity of each other. This is most likely due to the frequency hopping scheme that Bluetooth utilizes. A longer description of the Bluetooth technology can be found in Chapter 3.

The ubiquitousness of Bluetooth in the future makes it a compelling technology to investigate location tracking with. [21] wrote "Bluetooth technology is getting popular and cheaper and almost all the handheld devices and cellular phones now come with a Bluetooth version." As more and more devices are equiped with Bluetooth, a network for location tracking becomes built in to offices everywhere.

Tables 2.1, 2.2, and 2.3 taken from [21] show some results of interference between Bluetooth and 802.11b devices. Table 2.1 shows the results of tests performed to measure the affect Bluetooth devices had on a 802.11b network. The "middle cubicle" rows refer to a cubicle 7 meters and 1 partition away. The "last cubicle" row refers to a cubicle 10 meters and 2 partitions away. The Average column was the result of 10 attempts. It is evident from this data that there is a definite effect of the Bluetooth on 802.11b networks. They both use the 2.4Ghz ISM (Industrial, Scientific, and Medical) band and therefore some interference is to be expected. The distance and the number of solid objects in the way of the two devices also contributes to the degree of interference. The further away the Bluetooth devices are from the 802.11b devices, the less the impact.

Table 2.2 shows the effects of 802.11b networks on Bluetooh devices. The times measured are the times it took a Bluetooth device to send a 4437kB file to

Condition	Reciever Throughput (10^6 bits/s)		
	Lowest	Average	Highest
No Bluetooth device present	6.11	6.12	6.15
One Bluetooth Pair in close vicinity	4.37	5.33	5.91
One Bluetooth Pair in middle cubicle	5.64	5.88	6.11
One Bluetooth Pair in last cubicle	6.10	6.12	6.14
Two Bluetooth Pairs in close vicinity	3.60	3.94	5.08
Two Bluetooth Pairs in middle cubicle	6.07	6.09	6.11

Table 2.1: Performance of 802.11b in the precense of Bluetooth [21].

another Bluetooth device in the presence of varying numbers of 802.11b pairs. The amount of interference of 802.11b on Bluetooth is very minimal when compared to the reverse case. According to [21] this is the case because the rate of frequency hopping of 802.11b is 600 times slower than that of Bluetooth. Therefore a Bluetooth device will hop to the same frequency as the 802.11b device several times before the 802.11b device hops to the next frequency.

No. of 802.11b Pairs	File Transfer Time
1	6min 4sec
2	6min 5sec
3	6min 7sec

Table 2.2: Effect of 802.11b on Bluetooth [21]

Table 2.3 shows the effects of multiple Bluetooth devices on each other. As with the 802.11b affects on Bluetooth, the Bluetooth devices do not interfere with each other too badly. This is an important feature of the technology however. Bluetooth is a cable replacement technology, and it is important that a keyboard, mouse, scanner, printer, digital camera, Personal Digital Assitant (PDA), and other devices be able to communicate with each other in a small space. Therefore the affects of multiple

Bluetooth devices on each other is neccessarily small. The data in this table follows that requirement, showing that as more pairs of devices are added to the system, the effective bandwidth is not crippled to too large a degree.

No. Bluetooth Pairs	Effective Bandwidth
1	6min 7sec
2	6min 9sec
3	6min 11sec
4	6min 12sec

Table 2.3: Effect of Bluetooth on Bluetooth [21]

This data reaffirms the fact that Bluetooth has some properties that make it applicable for location tracking. Perhaps the one limitation to Bluetooth for location tracking is the specification that allows SS information to be optional. At this point, no known device is available that incorporates this feature. Many existing techniques for indoor location tracking with relatively good accuracy (i.e., within 3 meters of accuracy) rely on SS information to perform their calculations, so this is a severe limitation at least using current methodologies.

2.5 Database and Server Performance

The contents of an object tracking database grow at an alarming rate. Everytime a tracked object moves around a new entry is placed in the database. Tracking the location of all the staff at a hospital, for instance, would produce quite a bit of data. The reponse of the system degrades as new entries are added to the database, so a method of dealing with this should be discovered. Furthermore, the more objects

there are to track, the more processing the server has to do. Ways to eliminate this amount of processing have been studied.

These problems were investigated by [23] and [25]. [23] identified the problem of information expiration. They defined it as "if such an object does not report the new, up-to-date position and velocity, after some time, its positional information becomes too imprecise to be useful we say that it expires." Their research involved storing not only the object's location, but also the velocity. Using this information predictions can be made as to near future positions. This reduced the number of location updates to the database because relatively accurate predictions could be made instead of having every location explicitly put in the database.

The research performed by [25] involved different ways to limit the number of database queries needed in order to keep the data up to date. Their solution involved the use of dead-reckoning policies. The premise of dead-reckoning begins with the fact there are two locations for an object, the database location and the actual location. When the actual location of an object moves beyond a threshold of the database location, the system updates the database location for m to the actual location. The object being tracked was in charge of updating the database. The object knows what its current location is and what the last entry it sent to the database was, so it is in charge of sending the new data to the database.

[25] had three different versions of their dead-reckoning idea. The three were adaptive (ADR), disconnection detecting (DTDR), and speed dead-reckoning (SDR). SDR used a fixed threshold for the entire length of the tracking. The ADR technique adapts to the current situation. The tracked object will send a new threshold to the

database every time it sends its new location. This new threshold is computed using a cost based approach.

The third approach, ATDR, was developed because of a problem that is common to both the SDR and ADR techniques. The problem they exhibit is that a moving object that gets disconnected or cannot generate location updates cannot tell the database that they have moved. ATDR overcomes this limitation by decreasing the threshold over time. Eventually the threshold will become zero. If no location updates are received, the database assumes that the object is disconnected or otherwise cannot update its location information and flags it as such. The drawback of this approach, however, is an increase in the number of location updates required as the threshold goes down to zero.

3 Technology Frameworks

The WeBOTS system was built to facilitate the inclusion of different object tracking techniques. A framework was developed that allows a user to develop their own object tracking application. New object tracking techniques can even be added to the system without having to shutdown the whole system.

The system was built with with two types of object tracking built in: RFID and Bluetooth. RFID was chosen because the equipment was readily available with in the Experimental Laboratory for Advanced Networks and Systems (ELANS). The Bluetooth technology was chosen because no one else has investigated it for location tracking and if predictions come true, Bluetooth devices will become ubiquitous. Both technologies are off the shelf products with no modifications which fits in with the objectives of this thesis.

Neither of these two technologies was built for location tracking, so this thesis will be evaluating how well the technologies work for location tracking as well as investigating what features the technologies could use in order to be more applicable for location tracking.The rest of this section contains a discussion of these two technologies.

3.1 RFID

Radio Frequency Identification (RFID) has become very popular for many applications such as inventory tracking. There are not many standards for RFID so every manufacturer has their own implementations and some individuals have even come

up with their own custom implementations of the RFID idea such as [16] and [22]. Therefore, a discussion of some of the various properties RFID systems can and will possess follows. Then a desciption of the RFID hardware used in this project will be discussed.

A given RFID system contains one or more of the three following devices: an antenna, a tranceiver, and a transponder. The tranceiver contains a decoder and the transponder contains some unique information. The antenna can be placed anywhere; in the ceiling, floor, door frame, railroad track, etc. The antenna can pick up a signal from any transponder that passes through. The antenna is typically packaged with the tranceiver and decoder to create an RFID reader. The reader then can sense the presence of another RFID device through its antenna. The unique information is then sent from the transponder to the tranceiver to be decoded.

The RFID device that the reader senses is usually called a tag. RFID systems typically consist of two types of devices: RFID readers and RFID tags. Some systems, such as [16], have one type of device that can act as either a tag in the environment or a reader sensing the tags.

Tags can be classified as either passive or active. Passive tags are very simple devices and usually consist of a specifically positioned coil wire. As the passive tag enters a RFID reader's region of space, the signal the reader emits gets bent by the special coil in the tag. The reader can then uniquely identify the tag that entered its region based on the characteristics of the bent electro-magnetic field.

Passive tags can be as small as the tip of a pencil and have even been embedded into the skin of animals for tracking purposes. All the power with a passive tag system

comes from the reader and its antenna. Since the tag needs to be within the radius of the reader, the distance that the reader can still sense the passive tag is a lot smaller than the distance possible with active tags.

Active tags contain a transponder and antenna through which they can send out a signal. Active tags also contain their own battery and may also contain some memory on which to store pertinent information. On these devices the memory may be read only (RO), while on others, read/write (RW). Some tags even have up to 1MB of memory that they can store data in. The memory can contain anything from a set machine instructions to exectued, to the history of the tag, to any other application specific data neccessary. RO active tags, on the other hand, typically just store a special ID that can be used to lookup a name in a database. Active tags can vary greatly in size just like the passive tags discussed above. They can be the size of as small as a quarter and as large as 5 by 4 inch used to track heavy machinery and railroad cars [10].

3.1.1 RFCode Inc. Spider III System

This thesis makes use of the Spider III RFID system from RFCode Inc. [11]. The system has three main parts: RFID readers, RFID tags, and a piece of software called TagTracker Concentrator LI. Figure 3.2 shows these three components and their interactions. The tags in the system broadcast their unique ID over RF. The readers then read the unique ID off the air and send this information to the TagTracker software.

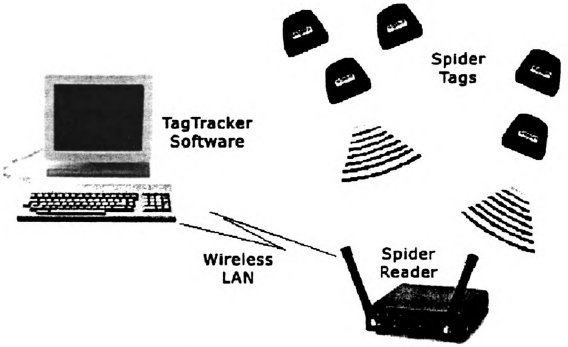

Figure 3.2: The RFCode Spider III System [11]

3.1.2 RFCode Inc. Spider III Reader Specifications

The RFID Spider III Reader sense tags in their area using one of eight different power ranges. They can send the ID of the tags they sense over traditional ethernet. Another option the readers have is to connect to a network using 802.11b, which has the added advantage of not having to route ethernet cabling to a remote location where the reader will be stationed. The following list defines the characteristics of the readers.

- Operating frequency: 308 MHz
- Reporting interface: Ethernet or IEEE 802.11b wireless network
- Detection range: 150 feet (can be increased to 1000 feet with external power)

- Online range control: provide digital control of readers' read range via providing configuration software and API (8 incremental ranges).
- Simultaneous identification (collision avoidance): can detect up to 500 tags in 7 seconds.

3.1.3 RFCode Inc. Spider III Tag Specifications

The RFID Spider III Tag is an active RFID tag. The tags are about 1.5in x .75in x .25in in size. The size is mainly due to the plasic casing around the silicon chip and battery inside. The tag is small enough that it could be attached to the name tag of an individual for tracking. The following list contains the specifications for the tags.

- Unique ID: each tag is pre-programmed with a unique 7-character ID for identification by reader.
- Battery life: 3-5 years
- Transmission rate: every 7 seconds

3.1.4 RFCode Inc. Spider III TagTracker Concentrator LI

The TagTracker Concentrator LI (TCLI) software consists of a Microsoft Windows [7] application and Application Programming Interface (API). This software is the medium through which the Spider III Readers are configured. This is also the interface through which the readers are told to begin looking for tags in the environment. The readers only operate when the TagTracker software is loaded and reading data. The software reads data coming in from the readers and writes it to two places. The first is to an

archive database and the other is to a socket if another application is connected to that socket.

The TCLI allows the configuration of the various parameters for the RFID readers. From the TCLI interface you can start up the readers and tell the readers which tags to look for. Alternatively, you can tell the reader to look for any tag that lies within its region instead of just particular ones. Furthermore, you can set the readers to work in one to two modes: continuous or exception. In continuous mode, everytime the reader senses a tag it passes it along to the TCLI for collection. So, the TCLI will receive data for the entire time a particular tag is in the region of the reader. The exception mode on the other hand, only sends data to TCLI when something new happens, i.e. when a tag enters the region of the reader and when the tag leaves the region of the reader. The TCLI also allows the user to set the power range of the RFID readers.

3.2 Bluetooth

Bluetooth is a specification for wireless communication between multiple devices over short distances. "A very key characteristic of Bluetooth that differentiates it from other wireless technologies is that it enables combined usability models based on functions provided by different devices" [14]. It was designed as a cable replacement technology. The idea is that one day a keyboard, mouse, scanner, printer, digital camera, web cam, joystick, cell phone, PDA and many other peripherals will be "connected" to a desktop wirelessly.

The specification for this new wireless technology began when industry leaders Ericsson, IBM, Intel, Nokia, and Toshiba formed the Bluetooth Special Interest Group (SIG) to develop this new technology. A goal of the system was to be global in scope. Therefore, a Bluetooth device that is purchased and used in the United State of America can also be used in England, Germany, Italy, Japan, etc. This is possible because the Bluetooth specifications put the frequencies used in the 2.4Ghz ISM band. This band is free throughout the world for short range wireless communication.

3.2.1 Bluetooth Architecture

The SIG came up with a layer based architecture for the Bluetooth wireless communication paradigm. The architecture has layers which are reminiscient of the Open Systems Interconnection (OSI) reference model developed by the International Organization for Standardization (ISO) [5] or Transmission Control Protocol/Internet Protocol (TCP/IP) stacks [18]. Figure 3.3 shows the application framework at the top and Bluetooth architecture at the bottom. The Bluetooth specification very strictly defines the purpose of each layer and their operating requirements.

The Radio layer's purpose is to be the transmission medium for the protocol. It is equivalent to the physical layer of the OSI model. [13] states that it will operate in the 2.4 - 2.4835 Ghz frequency band. This band was chosen because the vast majority of countries do not have limitations in this band and because Bluetooth is a world-wide standard, this was a logical choice. Certain countries have limitations on this band and special frequency hopping schemes have been developed for them. Also, it is important to note that devices implementing the full band cannot interoperate

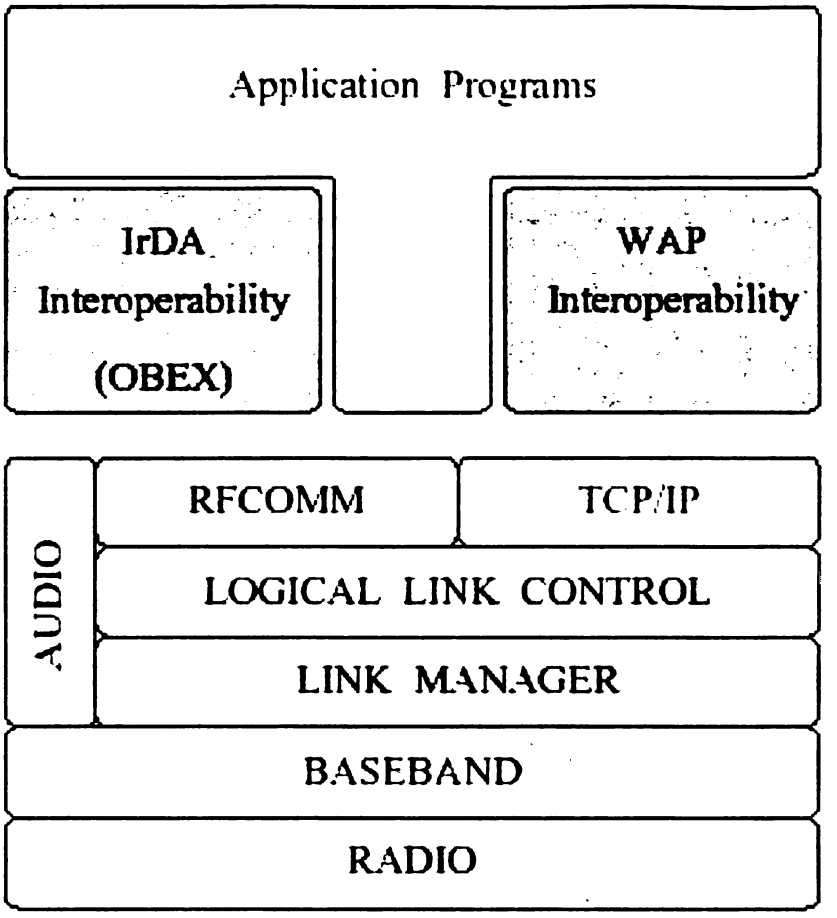

Figure 3.3: The Bluetooth Application Framework [14]

with devices implementing the specialized versions. Therefore the specialized versions are considered local versions for a single market.

There is an important feature of the Radio layer that, unfortunately for location tracking techniques, is optional. The optional feature is a receiver signal strength indicator. The purpose of this feature in the specification is to allow the receiver to tell the sender to increase or decrease its power output levels. The Bluetooth specification defines 3 power output levels. The receiver, wanting to keep an adequate signal, can tell the sender to increase the power output level. If the signal is stronger

than neccessary, the receiver can inform the sender that it can reduce its power outlet, helping to reduce power consumption.

The next layer to investigate is the Baseband layer. The Baseband layer performs similar functions to the data link layer of the Open System Interconnect (OSI) protocol stack. The Baseband layer controls when a unit can send, handles error correction, and retreives data from the Radio layer and slices it up into segments for the higher layers. [13] defines many packet types to carry the different types of data that can be transmitted across a Bluetooth network.

The Baseband controls the transmission of data on the Bluetooth network. It forces the radio layer to operate on a frequency hopping scheme. The channel is divided up into 79 or 23 RF channels that are hopped between in a psuedo-random order. The channel is divided up into 625μs long time slots. Each slot of time is numbered ranging from 0 to 2^27 - 1 that cycles with a length of 2^27. The devices may transmit only during the time slots. In order to support full duplex operation, the sent and received data is sent on different hop frequencies. Data sent by a device usually uses one time slot, but can take up to five time slots if the situation warrents it. So, the Baseband layer is the portion of the Bluetooth protocol stack that actually puts the data packets out on the RF network and pulls packets destined for the device off the network.

Here the basic layered scheme of the protocol stack changes and there are two options. There is an Audio layer as well as the Link Manager (LM) layer. Here the LM will be discussed. The LM is analogous to the Network layer of the OSI model or the IP layer of the TCP/IP protocol stack. The Baseband link actually moves

the data over the link, while Link Manager tells the Baseband what to send. There is a specific set of messages that are sent and received by the LM that do not move above the LM. These messages deal with link setup and control. These special LM messages have more priority than user data and as such will not get delayed by user data. Received user data will be pushed up the protocol stack to the Logical Link Control and Adaptation Protocol (L2CAP) layer.

The L2CAP has a lot of duties which include "protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service (QoS) information" [13]. L2CAP provides connectionless and connection-orientated data services to upper layers. The L2CAP layer is analogous to the Transport layer of the OSI model, or TCP layer of the TCP/IP protocol stack.

The upper layers passed the L2CAP get more application specific. Only certain applications will need RFCOMM and TCP/IP. TCP/IP is a very commonly used protocol and will not be discussed. The RFCOMM layer will be discussed, however, as it is a new protocol specific to Bluetooth. RFCOMM provides emulation of serial ports over the L2CAP protocol. The RFCOMM protocol can support up to 60 simultaneous connections between two Bluetooth devices. A complete RFCOMM application requires a program to be running on each device to handle the communication at the lower layers. The specification even allows for a Bluetooth dongle to be connected to legacy serial port devices such as an external modems that can therefore be used as a Bluetooth modem from another Bluetooth enabled computer.

The higher level layers take advantage of this protocol stack and do not have to be concerned with the complex low level details. Each layer of the protocol stack on

the local device communicates with the analogous layer on the remote host. Figure 3.4 shows the communication between the different layers of the Bluetooth stack.

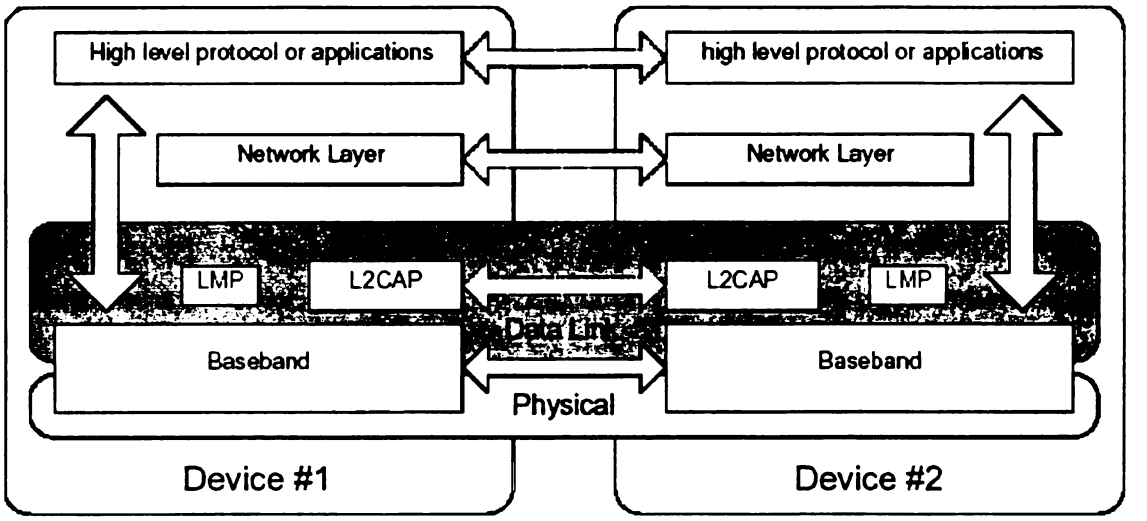

Figure 3.4: The Bluetooth Stack In Action

3.2.2 Piconets and Scatternets

In any given Bluetooth network, called a piconet, there is one machine, termed the master, that controls the network. Other devices in the piconet are called slaves. A single piconet can have up to 8 devices; 1 master and 7 slaves.

The master controls who gets to transmit and when. The master will create a frequency hoping pattern for all the slaves in its piconet to adhere to. All data is sent between the master and a slave. No data is exchanged between two masters or between two slaves. A single device can be a slave in multiple piconets concurrently, but a device can only be a master for one piconet. Devices that span piconets, i.e. are a slave in two different piconets concurrently, form a scatternet. Bluetooth piconets

can coexist within the same space and time independently of one another.

Figure 3.5 shows an example of a scatternet. The items in the diagram indicate which device in the piconet they are. The item 0 in a piconet represents the master for that piconet. So, in Piconet A device 0 is the master and has 3 slaves: 1, 2, and 3. Piconet B on the other hand has master 0 and 2 slaves. Device 3 in Piconet A and device 2 in Piconet B are the same device, thus connecting the two piconets together forming a scatternet. A possible situation, not depicted in this figure, is for a device that is a master in one piconet to concurrently be a slave in another piconet.

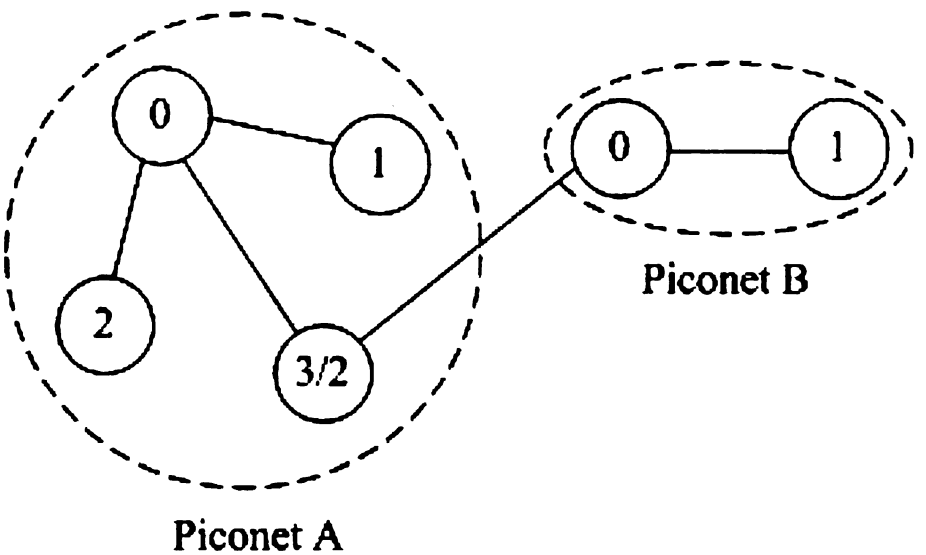

Figure 3.5: A Small Bluetooth Scatternet

In a given piconet, only 7 slaves may be active and communicating at a given time. Other devices in the area may be registered with the master however. Actually the Bluetooth specification allows for 200 devices in its vicinity to be registered with the master and sitting idly in parked mode.

Devices in parked mode do not transmit data to the master and sit idle waiting for the master to make them active. So, a digital camera that is not needed for a while may sit in parked mode near the Bluetooth master. Eventually, when the user wants to pull some pictures out of the camera, the master can tell another device in the piconet to go into parked mode if there are already 7 active devices, and then make the camera an active member of the piconet. Once active the Bluetooth master can request the image files from the digital camera and store them locally.

Uses of the Bluetooth technology such as the camera example above fall into different profiles. The Bluetooth SIG defined a number of application profiles. Chapter 3.2.4 discusses the various application profiles.

3.2.3 Host Controller Interface

The Host Controller Interface (HCI) was defined by the Bluetooth SIG. This is a set of operations that can be performed on a Bluetooth device to control the lower layer protocols of the Bluetooth system. Using the HCI it is possible for a programmer to establish links, close links, look for Bluetooth devices in the area, probe for what services a particular device has, and much more [13].

Table 3.4 shows the commands the HCI provides. Each of these commands is described in detail in [13]. Using the official Bluetooth protocol for GNU/Linux, BlueZ, it is possible to get access to these commands.

Command	Command Summary Description
Inquiry	The Inquiry command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discover other nearby Bluetooth devices.
Inquiry_Cancel	The Inquiry_Cancel command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode.
Periodic_Inquiry_Mode	The Periodic_Inquiry_Mode command is used to configure the Bluetooth device to perform an automatic Inquiry based on a specified period range.
Exit_Inquiry_Mode	The Exit_Inquiry_Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inqury Mode.
Create_Connection	The Create_Connection command will cause the link manager to create an ACL connection to the Bluetooth device with the Bluetooth device specifed using the unique Bluetooth address.
Disconnect	The Disconnect command is used to terminate an existing connection.
Add_SCO_Connection	The Add_SCO_Connection command will cause the link manager to create a SCO connection using the ACL connection.
Accept_Connection_Request	The Accept_Connection_Request command is used to accept a new incoming connection request.
Reject_Connection_Request	The Reject_Connection_Request command is used to decline a new incoming connection request.
Link_Key_Request_Reply	The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth device.
Link_Key_Request_Negative_Reply	The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Host Controller if the Host does not have a stored Link Key for the connection.

PIN_Code_Request_Reply	The PIN_Code_Request_Reply command is used to reply to a PIN Code Request event from the Host Controller and specifies the PIN code to use for a connection.
PIN_Code_Request_Negative_Reply	The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Request event from the Host Controller when the Host cannot specify a PIN code to use for a connection.
Change_Connection_Packet_Type	The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established.
Authentication_Requested	The Authentication_Requested command is used to establish a authentication between two devices associated with the specified Connection Handle.
Set_Connection_Encryption	The Set_Connection_Encryption command is used to enable and disable link level encryption.
Change_Connection_Link_Key	The Change_Connection_Link_Key command is used to force both devices of a connection to generate a new link key.
Master_Link_Key	The Master_Link_Key command is used to force both devices of a connection associated to the connection handle to use the temporary link key of the Master device or the regular link keys.
Remote_Name_Request	The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device.
Read_Remote_Supported_Features	The Read_Remote_Supported_Features command requests a list of the supported features of a remote device.
Read_Remote_Version_Infomation	The Read_Remote_Version_Infomation command will read the values of the version infomration for the remote Bluetooth device.
Read_Clock_Offset	The Read_Clock_Offset command allows the Host to read the clock offset of remote devives.

Table 3.4: Host Controller Interface Commands

The most interesting commands for the purposes of this thesis are the inquiry commands. The Inquiry command will be used to scan the area for new devices, and will be the basis for the proximity based location tracking methodologies utilized for the Bluetooth portion of this system.

3.2.4 Bluetooth Application Profiles

Bluetooth is a multipurpose system for wireless communication. Often called a cable replacement technology, it is also much, much more. Table 3.5 shows a listing of the different application profiles that the Bluetooth SIG defined. These profiles were defined to showcase some of the options the technology can be used for. Furthermore, having a standard for all devices to adhere to will force interoperability between all systems.

3.2.5 Bluetooth Devices Used

One of the Bluetooth devices used was the 3Com 3cREB96 [1]. It is a Universal Serial Bus (USB) device that conforms to version 1.1 of the Bluetooth specifications. The device comes with two different connection options depending on where you are using the device. One connector has a wire that allows you to set the device on your desktop space. The other connector will just hang off your machine right at the USB port. The devices also comes with the Bluetooth Connection Manager and driver software for Microsoft Windows 98SE/ME/2000/XP [7]. Many of the aforementioned application profiles are possible using this device and the personal digital assistant (PDA) Compaq Ipaq that was also used for this thesis.

Application Profile	Description
Service Discovery Application	Describes the protocols and procedures neccessary to discover services on remote Bluetooth devices.
Cordless Telephony	Describes the protocols and procedures neccessary to enable cellular phones to access fixed network telephony services through a Bluetooth base station.
Intercom	Describes protocols and procedures that allows the ability to perform walkie talkie type usage using Bluetooth.
Dial-up Networking	Describes the protocols and procedures neccessary to allow a computer or other device to use a cellular phone as a bridge to a dial-up internet access server.
Fax	Describes the protocols and procedures to allow a cellular phone or modem to use a nearby computer to send and receive fax messages.
Headset	Describes the protocols and procedures that allows full duplex audio between a headset (consisting of headphones and microphone) and another device like a cellular phone or desktop.
LAN Access	Describes the protocols and procedures for a device to use the Point to Point Protocol (PPP) to connect to another host that has access to a Local Area Network (LAN). Through this connection the original host then has access to the LAN as well.
File Transfer	Describes the protocols and procedures for transfering a file from one device to another.
Object Push	Describes the protocols and procedures neccessary for one device to push some encapsulated data structure to another device.
Synchronization	Describes the protocols and procedures neccessary to exchange data between two devices so they are both up to date.

Table 3.5: Bluetooth Profiles and Descriptions

Figure 3.6: 3Com 3CREB96 Bluetooth Device [1]

The device that was the tracked object during testing of the WeBOTS system was the Compaq Ipaq 3870 with Bluetooth [4]. Figure 3.7 shows a picture of the device. The Bluetooth tranceiver is located in the black section at the top. When the Bluetooth radio is enabled a blue light emitting diode (LED) will flash indicating its operation to the user. The devices consists of a 206 Mhz Intel Strongarm processor, 64MB of random access memory (RAM), 64,000 color display, integrated secure digital (SD) card support, and builtin microphone. The device ran Microsoft's Windows CE PocketPC 2002 operating system [7] as well as Bluetooth manager software.

3.2.6 Summary

Here a summary of some of the main features of the Bluetooth technology are listed. This is not an exhaustive, but highlights some of the most important to get an introduction to the technology. The following list summarizes some of the key features of the Bluetooth technology:

- Low-cost, low power radio tranceiver chip (0.5 square inches).

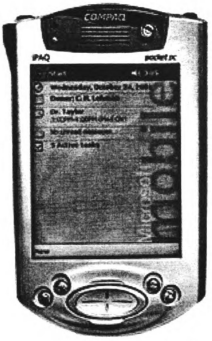

Figure 3.7: Compaq Ipaq 3870 with Bluetooth [4]

- Price of a Bluetooth module will be approximately $15-20 in the near future and the goal is to approach $5 or less.
- A low nominal range of Bluetooth radio (10 meters) for saving battery power.
- Extended range with external power amplifier (100 meters)
- Operation in the globally available and unlicensed 2.4 GHz band (global standard).
- One master controls piconet.
- Seven active slaves per piconet.
- Device can be a slave in multiple piconets concurrently.

- Up to 200 devices can be registed with a master in parked mode at a time.
- Speeds up to 1Mbps are possible.
- Up to 3 concurrent, synchronous voice channels.

4 The WeBOTS System

This section will describe, in detail, the various parts of the WeBOTS System. It will talk about the data that is involved with the system, the flow of that data through the system, and the various components of the system. Before going into detail about all the various components of the system, an overview of the system will be presented.

Figure 4.8 shows a highlevel view of WeBOTS. Around the outside of the diagram is a line that represents an Ethernet backbone where most of the communication takes place. Technically the communication is done over TCP/IP, so the various devices do not have to all be on the same Ethernet backbone, but for purposes of illustration, this figure suffices.

The next important features of the diagram are the tagd Server, RFID Client, and Bluetooth Clients. The RFID Clients get data from from the TCLI machines and send the data on to the tagd Server. TCLI stands for TagTracker Concentrator LI and is discussed in Chapter 3, but in short it is a piece of software provided by RFCode Inc [11] that handles all the communication with the RFID Readers. The Bluetooth Clients will scan for Bluetooth devices in the area and send the information to the tagd Server.

At the bottom of the figure there is an Internet cloud and Web User. A Web User requests the location of a particular object or person over the world wide web. The tagd Server then sends the response back over the Internet. All of these items will be discussed in more detail in the following sections.

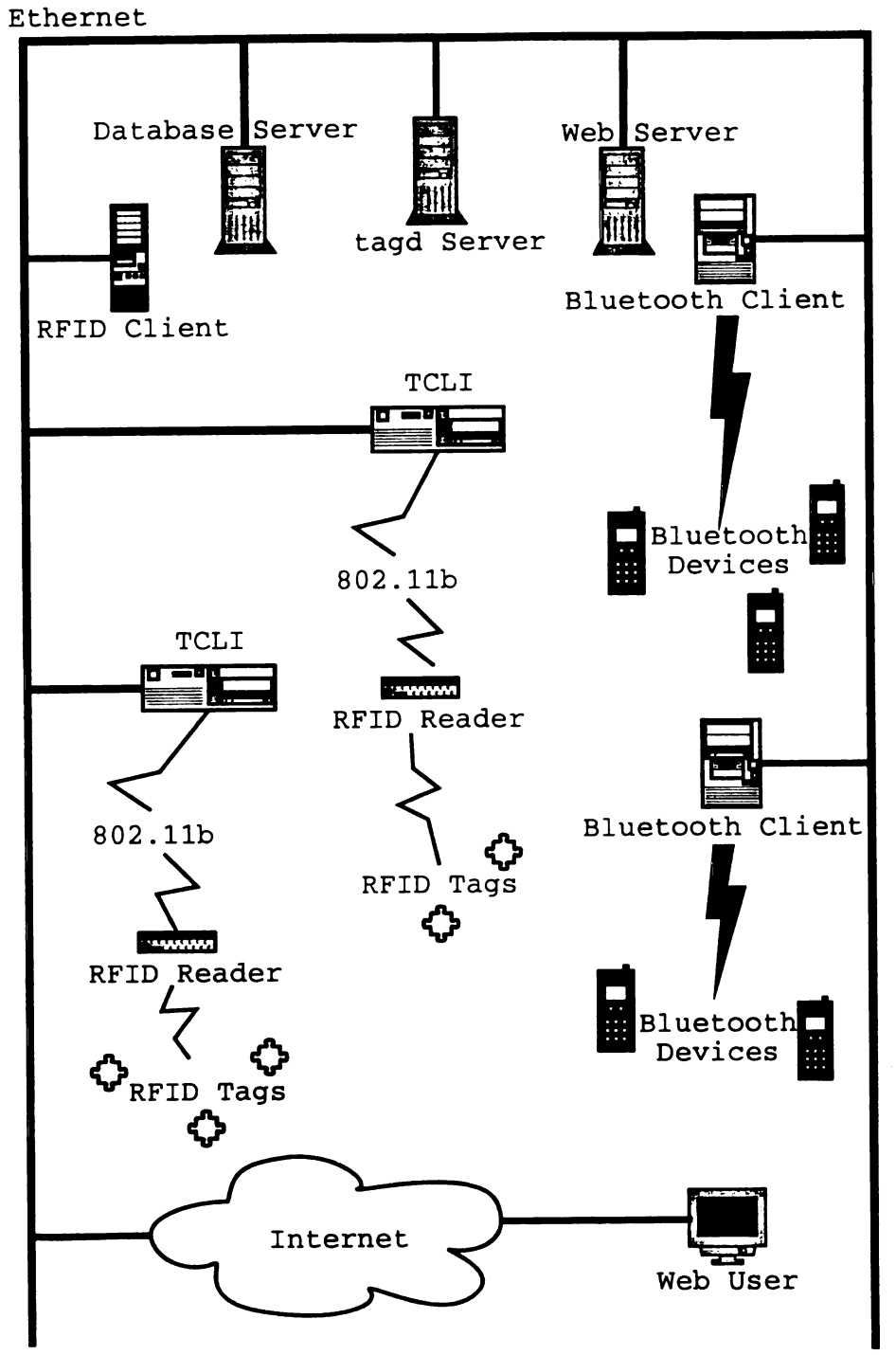

Figure 4.8: The WeBOTS Infrastructure

4.1 Data Model

The driving force of the implementation of many applications is the data that needs to be stored and the data structures that are used to provide that storage. In the WeBOTS system, data is stored in two databases that each contain the same three tables. One of the databases is a real-time database and the other is a database of archived entries. A larger discussion of the databases will come in Chapter 4.5. For now we will discuss the contents of the three tables contained in the databases: Event, Reader, and Tag.

Name	Description
tag_id	The unique key identifing the tag. References the Tag table.
reader_id	The unique key identifing the reader. References the Reader Table.
btime	The time the tag entered the region.
etime	The time the tag left the region.
power	The power level the reader was set at.

Table 4.6: Event Table Description

The discussion of the data model will start with the Event table. Table 4.6 shows the columns of the Event table. The tag_id is used as a foriegn key into the Tag table and the reader_id is a foreign key into the Reader table. The two time columns, btime and etime, represent the beginning time and ending time of the event described by the row. The btime, tag_id, and reader_id together form the primary key of the table. The last column is unused and is something for future work. It allows the event to know the power level the reader was set to. The power level denotes how much power is used by the reader's antenna to find tags. The higher the power level, the farther away the tags can be. This measure helps to know the accuracy of the

particular reader when the tag was found.

Each row in the table represents the time (btime) a tag, represented by tag_id, entered a readers region, represented by reader_id, and what time (etime) that tag left the readers region. The meanings of the columns are exactly the same between the real-time and archive databases.

Name	Description
id	The unique key identifing the reader.
power	The power level the reader is set at.
location	Textual description describing the readers location.

Table 4.7: Reader Table Description

The Reader table, shown in Table 4.7, is quite simple. The id column is a unique identifier for the reader and is the primary key for the table. The power column, just as for the event table, is unused. The purpose of the column is to store the power level the reader runs at. The last column, location, is a textual description of the location of the reader. An example of a location description would be placing the reader in a room, and setting the location description to that room number.

Name	Description
id	The unique key identifing the tag.
name	Textual description describing the tagged object.

Table 4.8: Tag Table Description

The Tag table, shown in Table 4.8, is even more straight forward than the Reader table. It has two columns, id and name, that are a primary key for the table and a textual description of the tag respectively. An example of a textual name would

be the user's name. The location of that user can be found later by selecting based on that textual name.

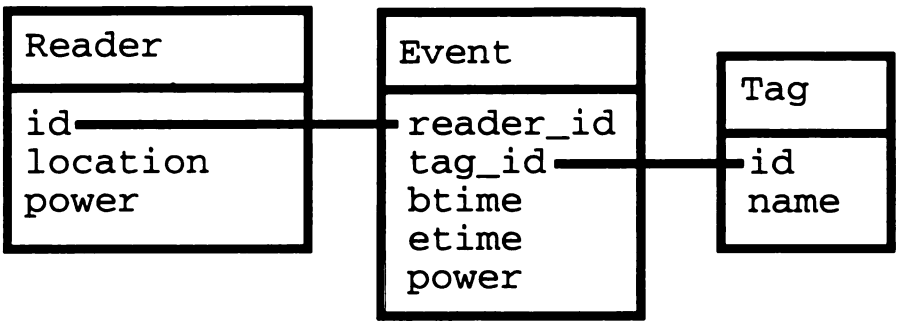

Figure 4.9: Overall Database Relationships

Figure 4.9 shows the interrelationships between the three database tables. Each box represents one of the three tables: Reader, Event, and Tag. The lines connecting the tables together represent the relationships between the tables. For instance, the Event table is related to the Reader table via the id in the Reader table and the reader_id in the Event table.

4.2 Data Flow

There are three main concurrent data flows through the system at all times. Each one will be described in this Chapter. The discussion will start with the simplest flow model, the archive data flow.

The basic flow for the archive data can be seen in Figure 4.10. Data is pulled out of the Location Database through the tagd Server and finally into the Archive Database. The tagd Server is a program that is always running for the WeBOTS system. It described in detail in Chapter 4.5.

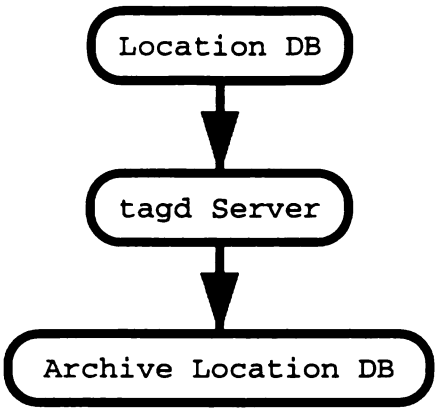

Figure 4.10: The flow of archived data

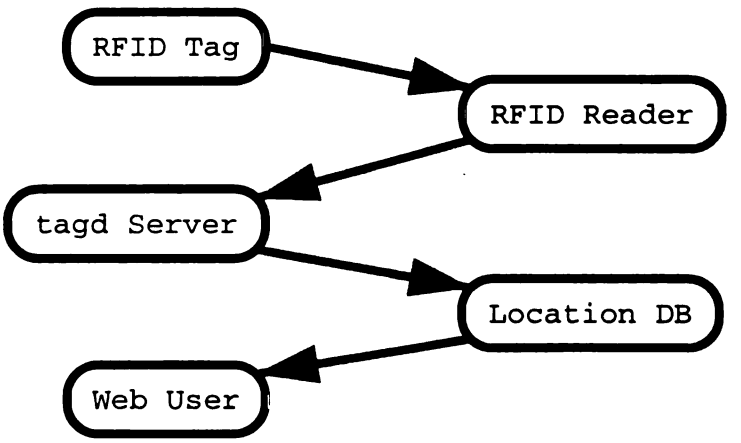

Figure 4.11: The flow of the data for the RFID implementation

The flow for the RFID portion of the system is a little more complicated than the archive data flow. A diagram of the flow can be seen in Figure 4.11. A RFID tag broadcasts its identification to all RFID Readers in the area. An RFID Reader will receive the broadcast, and then send the event to the tagd Server over an 802.11b network. The tagd Server will then update the information for the tag in the Location Database. Later, when a Web User requests information on that tag, the data will be sent to them.

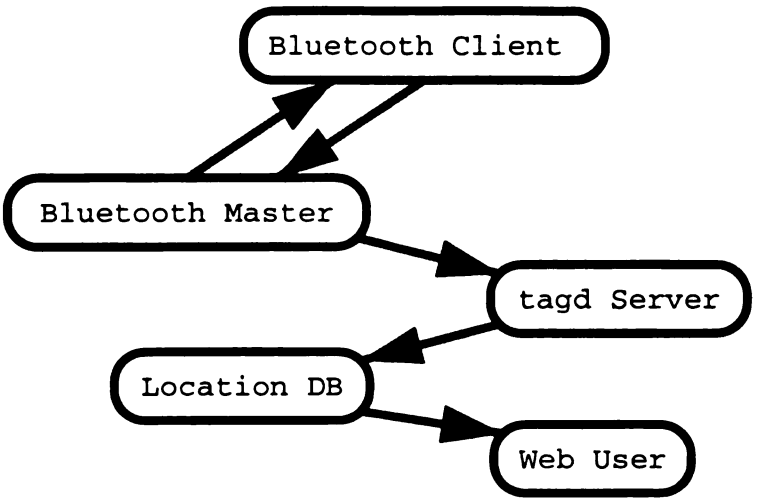

Figure 4.12: The flow of data the for the Bluetooth implementation

The flow for the Bluetooth portion of the system, as seen in Figure 4.12, is very similar to that of the RFID portion. The main difference between the two is the bi-directional flow of data between the Bluetooth Client and the Bluetooth Master. The Bluetooh Master first sends out a request to all clients in the area. The Bluetooth Client will then respond to the Bluetooth Master letting it know that the client is in the area. From there, the Bluetooth Master will send this event on to the tagd Server, which will add it to the Location Database, and finally get passed on to a Web User when requested.

4.3 RFID Client

The RFID portion of the program utilizes the TagTracker Connector LI (TCLI) program from RFCode's Spider III RFID system [11]. The TCLI connects to the RFID readers and interprets the data received from them. The TCLI will then listen on a TCP port for incoming connections.

A program, called rfid_reader, will need to connect to the TCLI software. This program looks at each response received and decides what to do. If a new RFID tag enters a RFID reader's area, the program sends a new RFID tag event to the tagd server. For those devices that are still in the area, i.e., those that it does not see as new to the area, the program does nothing. If the program notices that a device has not responded in a reasonable amount of time, it considers this RFID tag out of its region, and sends a RFID tag left the region event to the tagd Server.

Each RFID client can connect to one TCLI. The TCLI can connect to up to 10 RFID Readers. Each RFID Reader can find up to 500 tags every 7 seconds according to RFCode's specifications. Therefore, each RFID client can support up to 5000 tags in any given 7 seconds of time.

4.4 Bluetooth Client

As mentioned previously, location tracking with Bluetooth has not been explored much before because it was not one of the intended applications of Bluetooth at its inception. This section describes the method that was used to over come this limitation in the design of Bluetooth. There is a basic command that Bluetooth devices contain called HCI Inquiry. The command sends out a request to all devices in a Bluetooth master's piconet for them to respond. All devices (active and parked) in the area will send a response to the request. The results of the inquiry leave the master machine with a list of the Bluetooth devices in its area. Using this information we now have a proximity based method of tracking object locations.

A program, called bluetooth_reader, is needed for the Bluetooth system. When the program runs it makes a connection to the tagd Server, and begins repeatedly scanning for devices. If a new device enters the area, the program sends a new tag event to the tagd server. For those devices that are still in the area, i.e. those that it does not see as new to the area, the program does nothing. If the program notices that a device has not responded in a resonable amount of time, it considers this tag out of its region, and sends a tag left the region event to the tagd Server.

The number of devices a single Bluetooth client can support is based on the specifications of the Bluetooth technology. The specification states that a Bluetooth device can see up to 200 devices in parked mode. Therefore, there is a limitation on the number of devices the Bluetooth client can support.

4.5 tagd Server

The tagd Server is a program written in such a way as to allow for differnt types of location techniques concurrently. It allows multiple clients to connect to it. The clients collect location data and send it to the tagd Server in real-time. The tagd Server will then process the data and load it into the Location Database and does not get bogged down.

To illustrate the servers ability to work with many location tracking techniques, two location information gathering techniques were used; RFID and Bluetooth. Many clients can connect to the server at the same time. The server uses a standard interface for clients to connect to. A new client can then be created using a new location

tracking technology, and be able to plugin to the system without the server going down.

Since the server supports multiple clients at a time, the number of total devices scales up very quickly. However, since much of the processing is offloaded to the clients, the server acts mainly as a go between for the clients and the Location database.

The other important duty of the tagd Server is the archiving of data. As the number of devices (tags and readers) increases, the real-time Location database can grow very quickly. In order to keep the database small and quick a method of archiving the data was added to the server. Periodically the server scans the real-time database for entries that it considers "old". These "old" entries are removed from the real-time database and then added to an Archive Location database.

4.6 Database Server

The data for the system is stored in a database. The database server can be any type that supports communication over TCP/IP. All the data that is discussed in the above sections is stored in the database server. The database server could be split into two servers, one for the real-time database and one for the archive database.

4.7 Web Server

The web server is a very important component of the system in that it will be the medium through which users of the system can get to the information contained

within. The type of web server is largely unimportant as long as it supports server side programming of some kind to all connection to the Database Server. Requests from the Web User are sent to the Web Server. The Web Server and Web Frontend perfrom the neccessary processing and send the results back to the Web User.

4.8 Web Frontend

The web front end puts a nice face on the database allowing users to make inquiries about where objects or people are located. There is a basic interface to the real-time system allowing the user to click on an object or person's name which will then go to a new page showing the last known locations. There is also a search section where the user of the system can search through the archive database. For instance, a search for who was at location X between times Y and Z is possible.

5 Experimentation with WeBOTS

5.1 Experiment Environment

Two different locations served as the environments for all development and experimentation for WeBOTS. The RFID development was performed in the Experimental Laboratory for Advanced Networks and Systems (ELANS) at Michigan State University. ELANS is a very dynamic environment with students and professors coming and going all throughout the day. Figure 5.14 shows the layout of ELANS and the locations where the various tests were performed.

The Bluetooth experiments took place in an apartment with a few rooms. This environment was chosen because testing could take place in multiple rooms, with doors open or closed to see the affects of the lack of line of sight. Figure 5.14 shows the layout of the apartment and the locations where the various tests were performed.

5.1.1 Hardware

This section lists the hardware and software ran on the development and experiment machines. Three main machines were used for the entire thesis. Machine number one was a Dell desktop machine. The development of the RFID portion of the system was done completely on this machine. The specifications for it follow.

- Manufacturer: Dell
- Model: Dimension

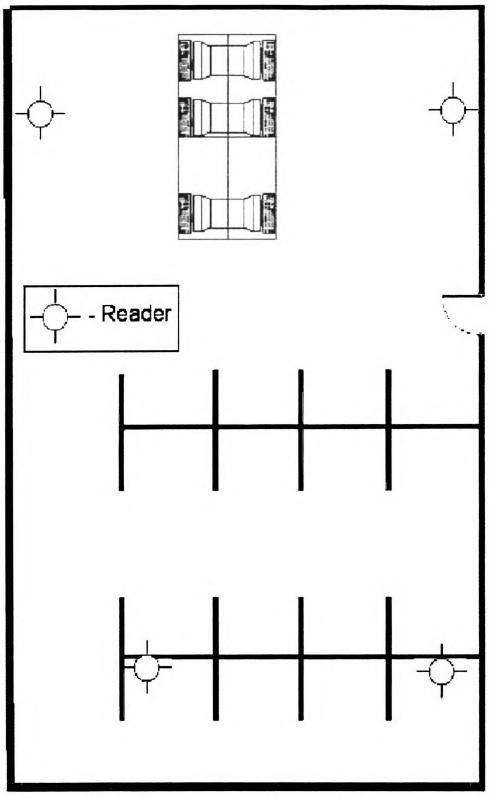

Figure 5.13: RFID Test Environment

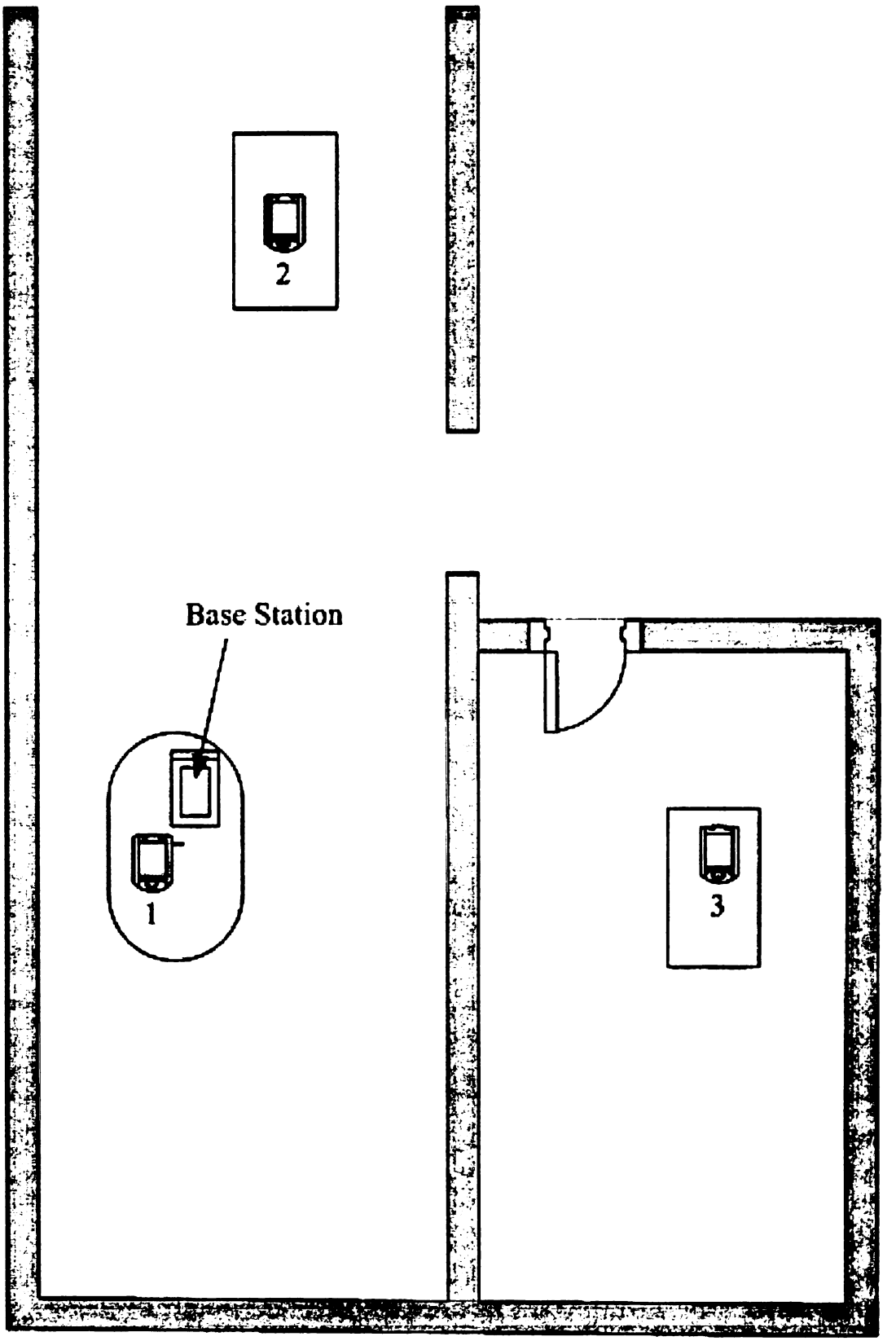

Figure 5.14: Bluetooth Test Environment

- Processor: P4 GHz
- System Memory: 512MB
- Storage: 40GB Western Digital 7200RPM HD
- OS: Redhat Linux 7.2
- Network: 100Mbit Ethernet

The second machine to metion was a Dell laptop. All testing and development of the Bluetooth portion of the system was performed with the laptop. The specifications for it follow.

- Manufacturer: Dell
- Model: Dell Inspiron 8100
- Processor: Intel Pentium III 1.0Ghz
- System Memory: 512MB Ram
- Storage: 30GB IBM TravelStar HD
- OS: Gentoo Linux 1.2
- Network: 100Mbit Ethernet

The third machine to mention is the Microsoft Windows based machine that runs the TagTracker LI software for the RFID system. The specifications of this machine follow.

- Manufacturer: NEC
- Model: PowerMate ES Slimeline
- Processor: Intel Pentium III 1.0Ghz
- System Memory: 128MB Ram
- Storage: Quantum Fireball 20GB
- OS: Windows 2000
- Network: 100Mbit Ethernet

5.1.2 Software

There are three C++ programs that comprise most of the system. They are tagd, rfid_reader, and bluetooth_reader. The tagd server runs on the same machine as the database server and web server. The database used was MySQL and the web server used was Apache with PHP. The tagd program successfully runs on both the Dell desktop machine and the Dell laptop. The protocol stack used to communicate with the Bluetooth device was the Bluez protocol stack. It is considered the official Bluetooth protocol stack for GNU/Linux operating system. [2], [8], [6], [3], [9]

5.2 RFID Testing

This section contains a brief discussion of the RFID system's capabilities. Since RFID has been around for a while and many studies have been performed on the technology,

the focus of the experiments will be on the Bluetooth portion of the system. Here a few points and observations of the RFID system are discussed.

Before going into the details of the RFID testing Figures 5.15, 5.16, and 5.17 show screenshots of the web pages at various times for a tag. Figure 5.15 shows the location information for the tag before it enters the southwest corner of ELANS. Figure 5.16 shows the location information for the tag after it enters the southwest corner of ELANS. Finally Figure 5.17 shows the location information for the tag after the leaves the southwest corner of ELANS.

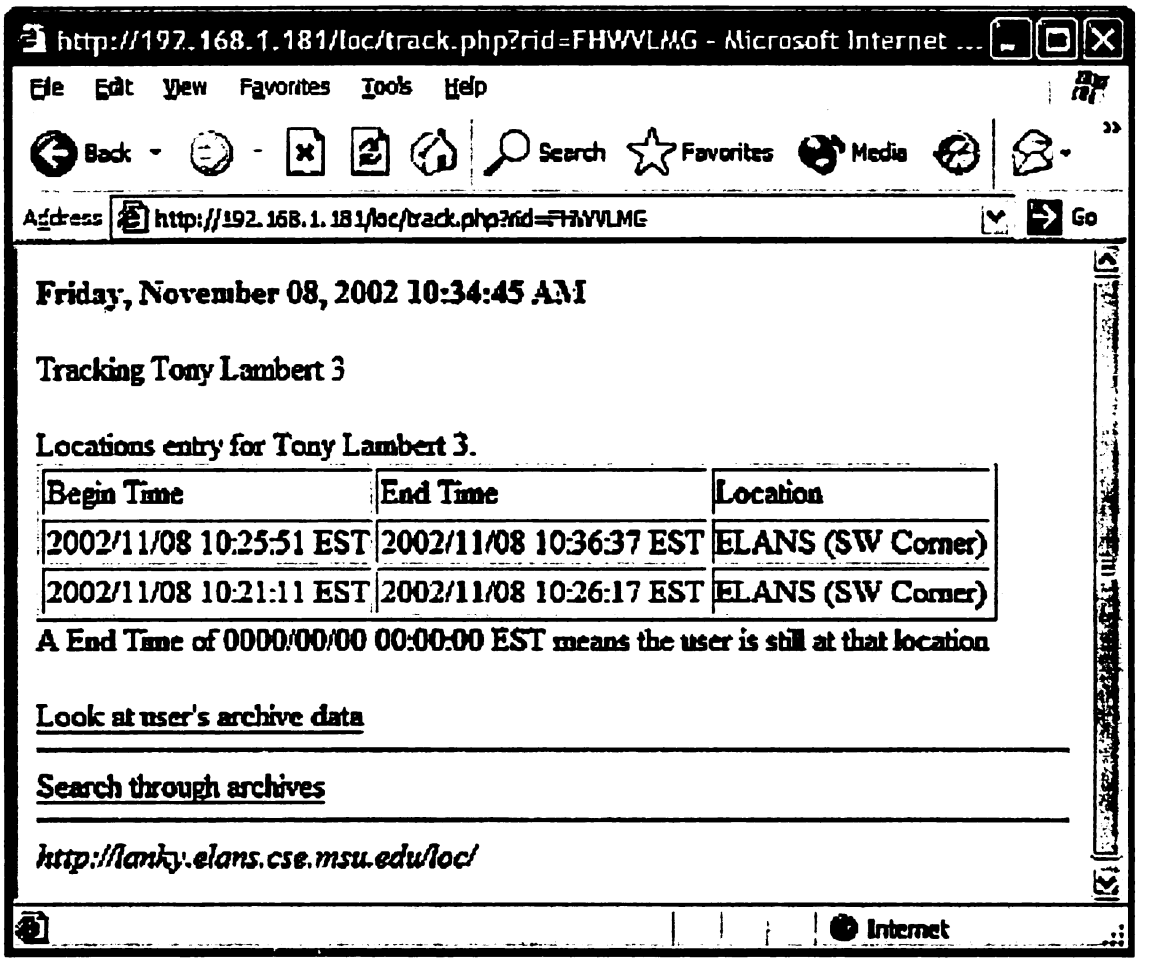

Figure 5.15: RFID Screenshot Before Tag Entered Region

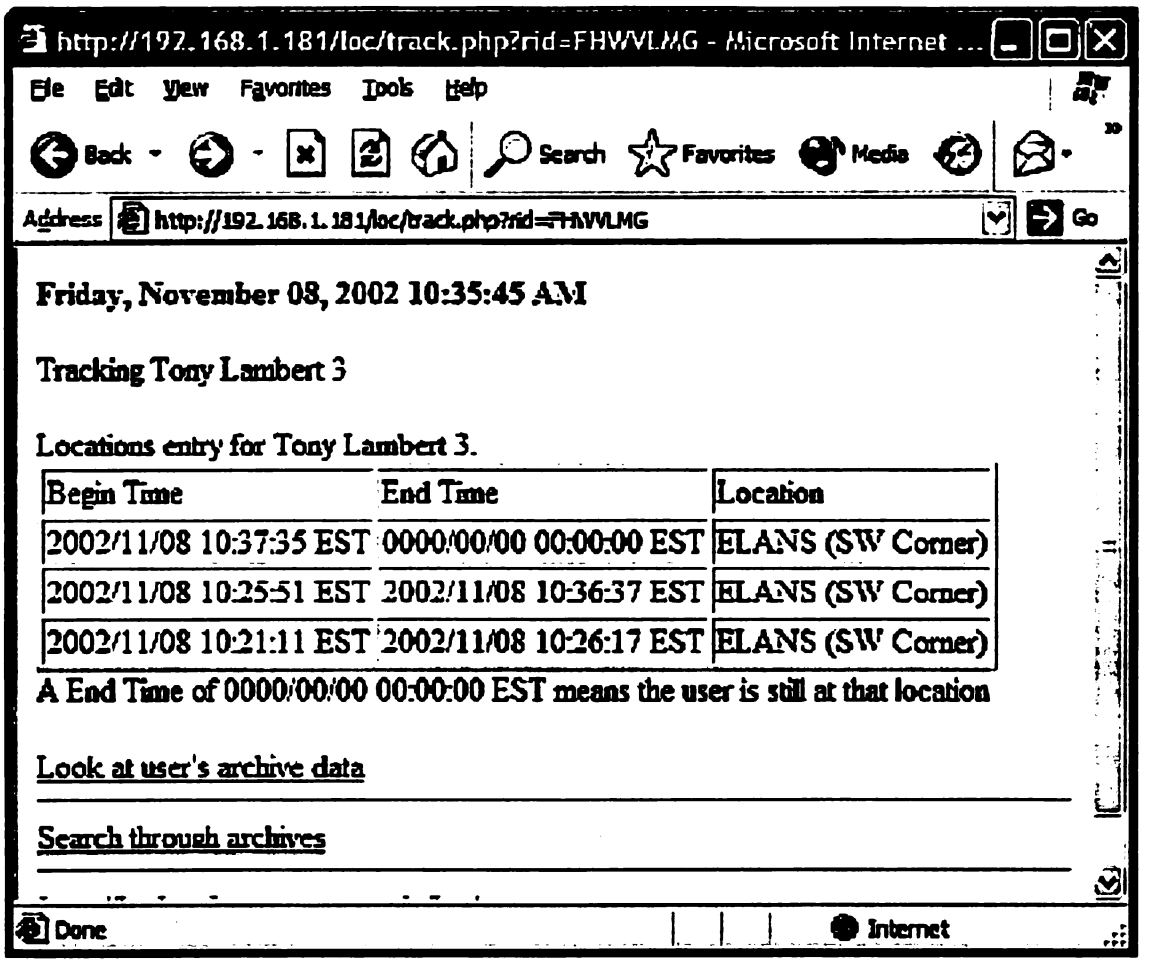

Figure 5.16: RFID Screenshot After Tag Entered Region

5.2.1 System Latency

The maximum length of time it takes a RFID tag to be sensed by the system is 7 seconds. The length of time it takes for the data to be transferred to the tagd Server is very small next to the 7 seconds provided they are part of the same intranet, so the latency of the RFID system is nominally within 7 seconds. If there are a large number of tags in the system, however, this 7 seconds could be increased due to interference if two tags try to transmit their ids at the same time. The RFID tags employ a random

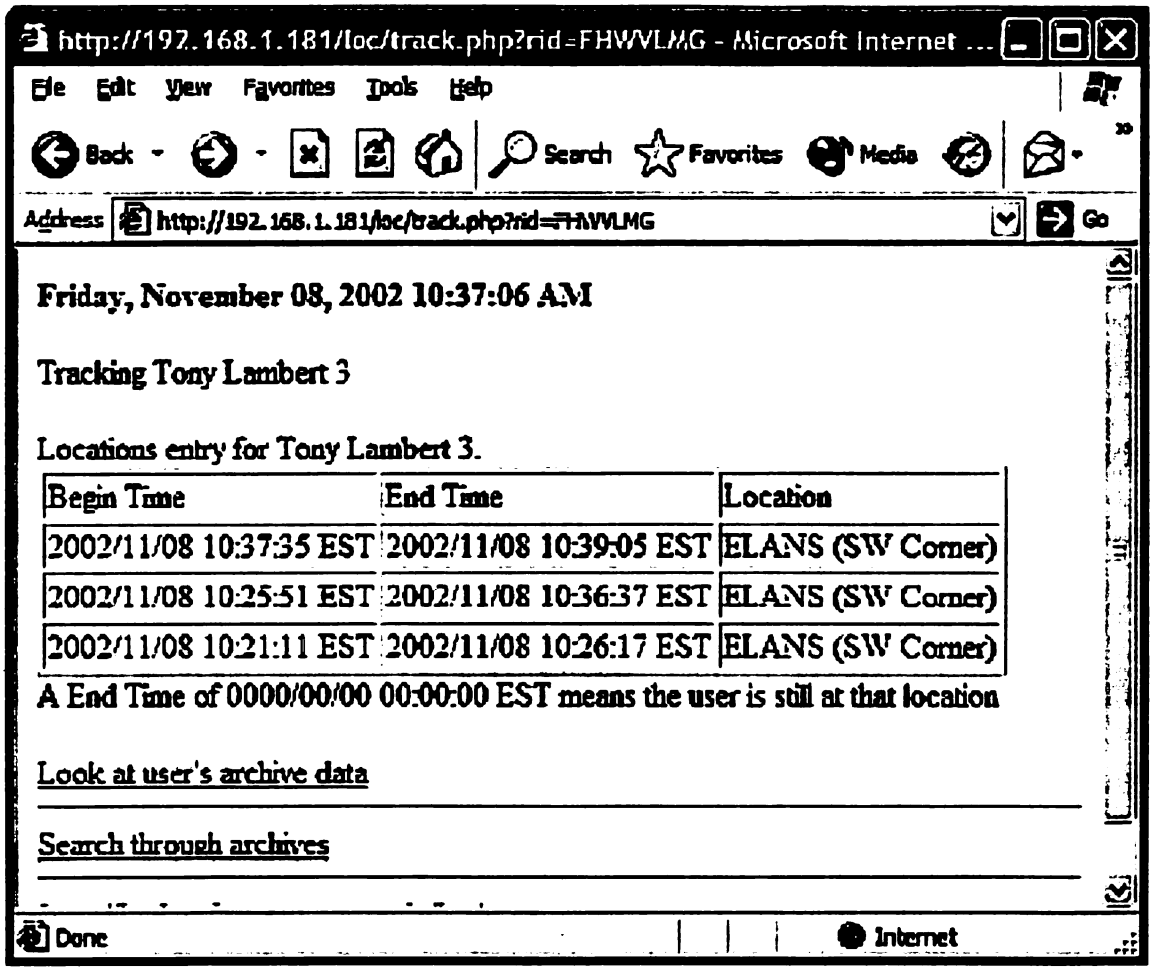

Figure 5.17: RFID Screenshot Tag Left the Region

frequency change for the next 7 second period, so they should be able to send their unique id to the reader in the next 7 second period giving us a likely maximum of 14 seconds, with a median under the 7 second mark.

For devices leaving a region, however, the latency is worse. In order to decide that a device has left a region the client would need some signal sent by the RFID tag to the reader that it left. However, the RFID tags only broadcast their unique id, so this is not possible. Therefore, if a RFID client does not receive a signal from a tag that was previously within the region for 30 seconds, the client assumes the tag

left the region. Therefore, the latency for a tag leaving a region is 30 seconds.

The 30 seconds that was decided upon was an arbitrary choice. It was used because it allows for a few collusions to occur when the tags attempt to transmit their id's, while still giving a relatively small amount of latency for when the system knows that a tag leaves a region.

5.2.2 Number of Devices

The number of devices the system can support is quite large and hard to calculate. The specification of the RFCode Inc.'s Spider III system states that a reader can sense 500 different tags in 7 seconds. Now, WeBOTS will determine that a tag leaves a region if that tag has not been sensed after a specific amount of time. With this being the case, how many tags can get through enough times to allow them to be considered still in the region for a 30 second time frame? For instance, 1000 tags might be able to get their id to the reader in a 30 second period. If it was known how long a tag took to broadcast its message, than a theoretical maximum could be calculated such that each tag would get a specific timeslot that it could fit in without collusions.

Well, more than 500 would most likely cause enough interference that it would cause 500 tags to be the maximum even though the question really is how many different tags can be reliably sensed every 30 seconds. There were not enough RFID tags to perform the test, but a guess would be 500 tags is most likely the maximum that the system could support because of collusions. Furthermore, the amount of space 500 people would comfortably take up would most likely be more space than a

single reader could cover, making 500 tags a number that would never be reached.

5.3 Bluetooth Testing

This portion of the experiment section focuses on the Bluetooth portion of the system. A few important questions of the Bluetooth system are posed and answered. Before the description of the Bluetooth experimentation, the following screenshots are shown of the Bluetooth portion of the system in action. Figures 5.18, 5.18, and 5.18 show the state of a Bluetooth device at various times. The figures show the system before the tag enters, after the tag is in the region, and after the tag leaves the area around the laptop respectively.

5.3.1 Scan Length and System Latency

The latency of the Bluetooth client depends solely on how long it takes the Bluetooth base station to perform its scan of the area. The scan length is adjustable. The Bluetooth HCI Inquiry command discussed in Chapter 4 scans for any Bluetooth devices in the area. Scans sent out by the Bluetooth device will be some positive integer multiple of 1.28 seconds in length. The important questions for Bluetooth scans are shown in the following list.

- How long should the scans be to find all the devices in the area?
- How long should the scans be to find the devices often enough to never consider them as having left the region while they are still in the region?

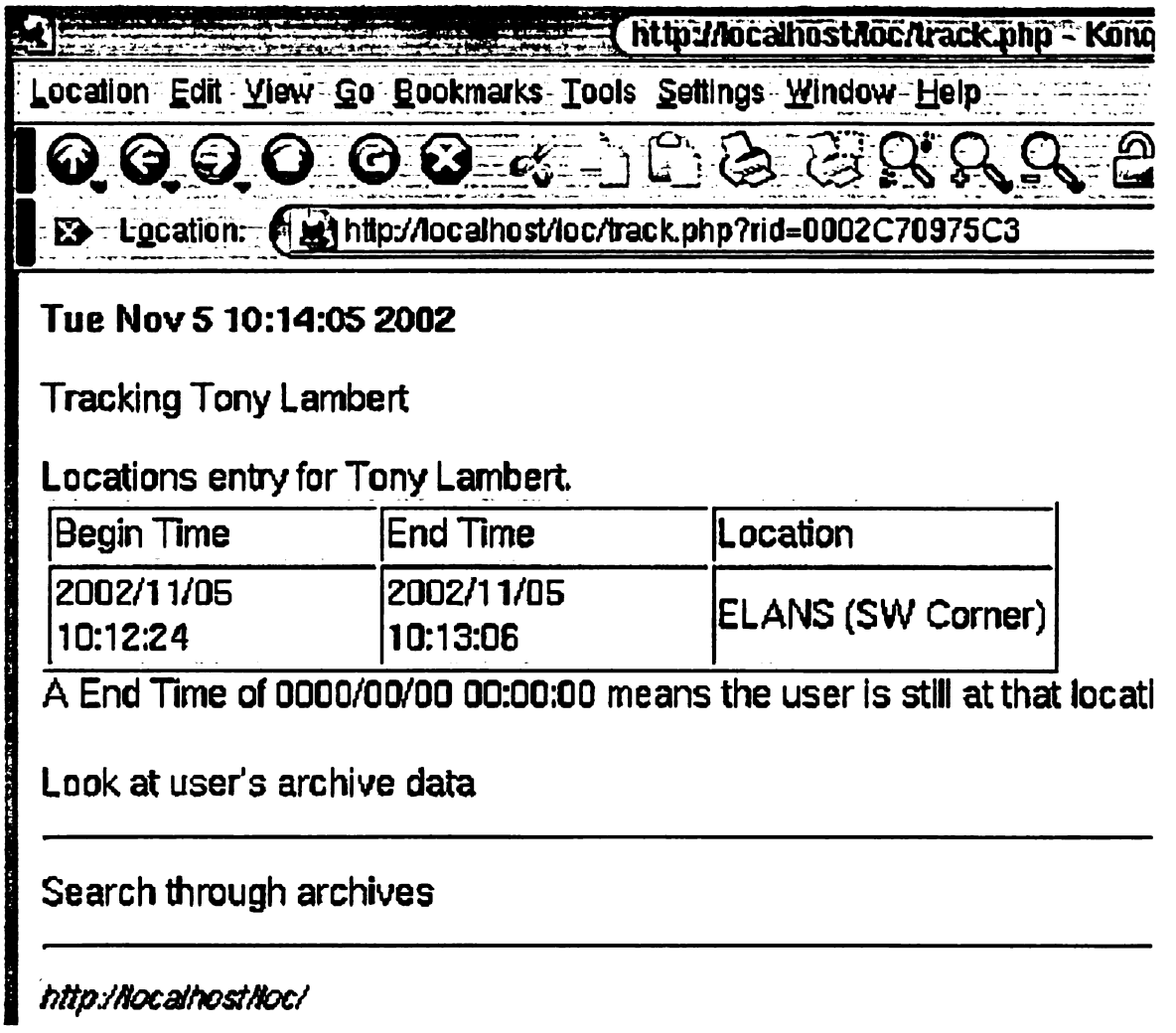

Figure 5.18: Bluetooth Screenshot Before Tag Entered Region

- Does the number of devices in the area mean the scan length needs to be increased?
- Does distance and object interference play a big role in the requirements of the scan length?

An important thing to note is that a Bluetooth device that responds to a scan at some given time will not neccessarily respond to a scan at a different given time. It can vary quite a lot actually. Table 5.9, provides some experimental data showing

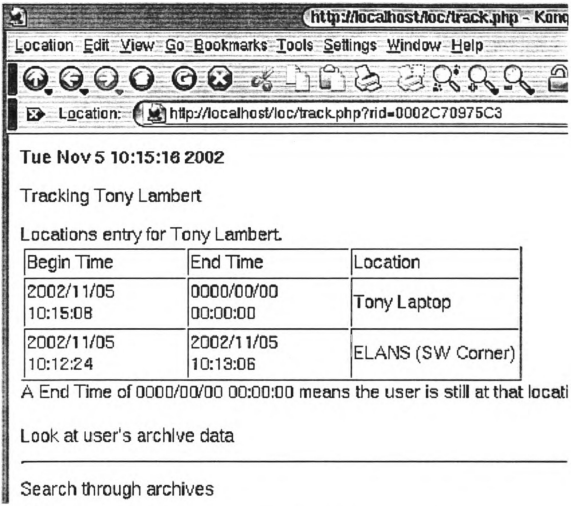

Figure 5.19: Bluetooth Screenshot After Tag Entered Region

this phenomenon.

Table 5.9 is broken up into three main sections, Next to Base Station, 10 Feet Away, and 10 Feet Away with Wall Blocking. These three positions can be seen in Figure 5.14 as the little Ipaq icons. The locations of the experiments correspond to locations 1, 2, and 3 in the figure respectively. The base station is denoted by the box in the center of the diagram. Each experiment took place independently of one another.

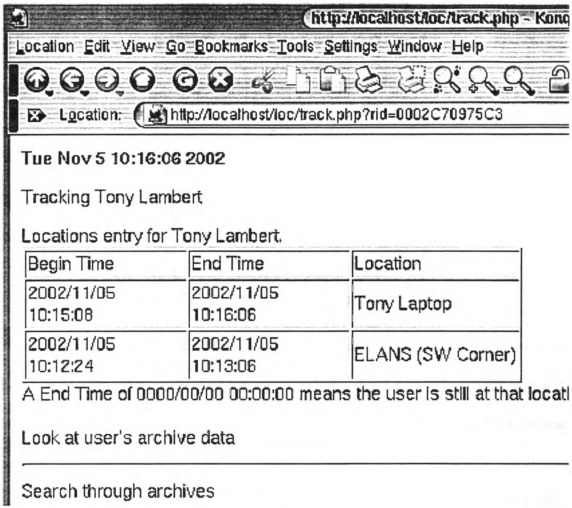

Figure 5.20: Bluetooth Screenshot Tag Left the Region

For each test, the Bluetooth base station repeatedly sent out scans of the length N * 1.28 seconds to find devices in the area. In every test there were two Bluetooth devices in the area placed next to each other at the aforementioned locations. So, for instance, the third row in Table 5.9 represents if both devices, one device, or neither device responded to a 5.12 second scan. Each test used 520 total scans with the first and last 10 scans thrown out.

N * 1.28s	Both	One	Neither
Next to Base Station			
4	444	55	1
3	376	92	32
2	0	79	421
10 Feet Away			
4	477	23	0
3	418	80	2
2	0	5	495
10 Feet Away with Wall Blocking			
4	466	33	1
3	403	92	5
2	0	0	500

Table 5.9: Variability of Bluetooth Scan Response

A few interesting results can be seen from this test. For ease of discussion the 2 * 1.28 second scan length will be referred to as the 2 unit scan, likewise the 3 * 1.28 second and 4 * 1.28 second scans will be referred to as the 3 and 4 unit scans respectively. The first is the difference between the 2 unit and the 3 and 4 unit scans lengths. The 2 unit scan is not reliable at all and cannot be used for object tracking. It was never able to find both devices and rarely found even one. The 3 and 4 unit scans seemed to be equally as good, so it would be logical to pick the 3 unit scan as it would decrease the system latency for a new device entering a region.

However, it could be the case that once 100 devices enter a region, a lot more collusions take place and the scan length of 4 units provides more reliable data. The default HCI Inquiry scan is 7 units in length which is just shy of 9 seconds. This might work better for the case when large numbers of devices are in the area, but due to the great expense of devices that are Bluetooth enabled, this was not able to

be investigated further.

The latency for a device leaving an area is subject to the same drawbacks as the RFID system. If a base station does not hear from a device within 30 seconds of the devices last response, then the base station considers that device as having left the region. It forwards this response on to the tagd Server. Therefore there is a maximum of a 30 second latency when a device leaves a region until the Bluetooth system is aware of it.

The question of object interference is also important to look at. Object interference does affect the Bluetooth scans to some degree, and can be seen by the fall in numbers in the table when the Bluetooth devices were put behind the wall. The wall in the way was not that detrimental to the results for the 3 and 4 unit scans, however, and still had good enough response during the time to consider both devices in the region at all times. Based on these results it is apparent the system would perform quite well in an area with other things in the way such as cubicles of an office or medical carts in a hospital room.

5.3.2 Number of Devices

The number of devices the Bluetooth client can support is 200. This is a restriction put in place by the Bluetooth specification and is discussed in Chapter 3. A Bluetooth master can only have 200 devices registered with it at any one time. Right now ELANS has a limited number of Bluetooth devices with which to do testing, therefore this theoritical limit can not explicitly be tested.

5.4 Database experiments

This section focuses on the results of the various techniques used to lessen the load on the database and server.

5.4.1 Database Entry Reduction

This section of the system went through two iterations. Initially, the plan was that every time a base station (either RFID or Bluetooth) sensed a device, it sent a message to the tagd Server. This message would then be inserted into the database. That means that for the RFID system where the reader can sense a tag every 7 seconds, there would be a new entry in the database every 7 seconds. Over a 30 minute period, that is more than 250 entries for one tag.

For the sake of argument, assume there is a room that four people work in. Each person works from 8:00 AM to 5:00 PM with a 30 minute lunch and two 15 minute breaks. Therefore each person is potentially in the room for eight hours. This would create about 16,400 new entries in the database for just one room of a building. Extending this to a building with multiple rooms each with multiple people in them and it easily becomes apparent that this growth rate is just too fast to be feasible.

The system was then rebuilt as it now stands. The client programs will note the first time a new tag enters the base station's region. At this point a "tag entered the region" message is sent to the server. This event get added to the database with the current time as the beginning time. Then every time a client sees that tag still in its region, it does not send a new message to the server, but updates and internal

timestamp associated with the tag. If the client has not sensed the tag again in a given amount of time, then the client will send a "tag has left the region" message to the server. The server then updates the entry in the database.

Going back to the previous example of the four people in the from 8:00 AM to 5:00 PM, the amount of entries in the database will be very, very minimal compared with that for the original setup. Each person will have only four entries in the database, with 8 total database updates during the day assuming they only leave for their breaks and lunch. That means, in total, 16 new entries to the database and 32 total database updates. This scales up a lot better than the 16,400 entries seen previously.

5.4.2 Archive Database

The tagd Server has a thread that wakes up every hour and cleans out all entries in the real-time database that are older than 8 hours. The age that entries get archived is configurable with a compile time setting, but 8 hours was chosen as the default. 8 hours was the default with the thought that a person's locations for the current entire work day could be quickly accessable without going through the large and slow archive database for the information.

Archiving the data in this fashion helps location lookups for the current day run really fast, but does not completely lose all the old data. The ability to search through the archive data is still available through the Web Frontend.

5.5 Scalability

The software for the WeBOTS system was written entirely for the GNU/Linux operating system. The tagd Server spawns a thread to handle each client that connects to it. The number of threads in a GNU/Linux system is configurable by using the command "ulimit". The "ulimit" command can tell the Linux kernel to increase the number of threads allowed per process. So, the scalability of the system comes down to the processing power of the tagd Server and how many concurrent threads it can spawn and still have the performance be acceptable. This will vary greatly from system to system and is therefore difficult to quantify.

6 Conclusion and Future Work

This section wraps up the thesis. It discusses the outcomes of the objectives attempted for the RFID, Bluetooth, the database, and WeBOTS as a whole. Finally it concludes with a discussion of future work on the system.

6.1 RFID

The RFID technology performed as expected. When tags entered the region they were sensed and when tags left the region, this was also noted by the system. The exception mode that the TCLI software can run in did not seem to provide accurate results, so the continuous mode had to be used.

The RFID system used had 8 different levels of power output. This was good because it allows the RFID readers to be configured more accurately for different sized rooms. In a room that is 10ft x 10ft the reader can be set to operate in a small range. However, in a room that is 100ft x 100ft the reader could be configured to sense a much larger range, or perhaps the room could be covered with many readers using a small range. The RFID system is very configurable this way.

The RFID system works quite well, but its major drawback is the cost of the RFID readers. Every room in the building would need a RFID reader. This would increase the cost of the system very quickly. The good thing however, is once the readers are placed in the rooms, the cost of new RFID tags is very minimal in comparison. Therefore the cost of setting up an RFID only system would be very steep up front, but would be pretty cheap after that because only new tags would

have to be purchased.

Another drawback to the RFID system is the reliance on an intermediate machine between the RFID reader and the tagd Server. This intermediate machine, the machine that runs the TCLI software, causes problems. For every 10 RFID readers in the system another machine has to be running because the TCLI software only supports up to 10 readers at a time.

6.2 Bluetooth

One of the objectives this paper set out to investigate was the use of Bluetooth as a method of obtaining location information. Bluetooth turned out to be a very capable location tracking tool, even more reliable for the purpose than RFID. Bluetooth was built to make sure that data was transferred while the RFID system makes no such claims as to reliability. For instance, in a Bluetooth system, it is imperative that all data of a document gets completely copied from one device to another, but it is okay if a RFID reader misses a tag's beacon because it will be picked up the next time a beacon is sent.

Furthermore, the Bluetooth system seemed more stable than that the RFID system, mainly because the TCLI software was slow and buggy. The Bluetooth system was very stable obtained good and quick responses from devices in the area.

Bluetooth's reliability and ability to penetrate through objects can actually be seen as a detriment, however. The Bluetooth system works really well through walls, which means that an object will be seen as being in multiple rooms at the same time.

Furthermore, if it can work through walls, it can work through floors and ceilings. If a device is seen on multiple rooms on the same floor, a human could find them easily enough, but if a human has to check three different floors, the system would not be very useful.

Some of Bluetooth's features make it a really powerful system for location tracking, while others can be a hinderance. Chapter 6.4 talks about some future work to get around these problems. Some of these problems could be handled more easily if the Bluetooth technology was modifed to make it more applicable for location tracking.

The first important feature that is missing from the Bluetooth technology that would be useful in obtaining location information is signal strength information. Using signal strength data, a user's location can be pinpointed to much greater accuracy and therefore be able to tell exactly where and in which room an they are in.

Another feature that would allow more configurability in a Bluetooth tracking system is more power output levels. The Bluetooth specification only allows for three different levels of power output, with the smallest level already having enough power to cover an area with a diameter as large as 10 meters, does not allow for the base station to be setup to work in a lot of different sized rooms.

Lastly, cheap Bluetooth devices could be manufactured. These devices would bascially just have a Bluetooth id that could be embeded in a name tag much like RFID tags. These Bluetooth tags could then only respond when a HCI Inquiry scan is sent out. Using these devices and methodolgies such as those employed in the LANDMARC system discussed in Chapter 2 should make for a very stable and

accurate location tracking paradigm.

6.3 Database

The methods used to limit database interactions and size were definitely successful. Pushing the logic to test if a device has entered or left a region out to the client applications really reduces the strain on the database. The database only gets notified if something interesting happens, so it no longer gets probed or modified everytime a tag is sensed.

Furthermore, the archiving of the data out of the real-time database keeps the real-time database small and efficient. Having less data in the real-time database to go through when selecting data out of it makes it much more responsive for the web site.

6.4 Future Work

It is possible that a single object in the system could have multiple tags associated with it. For instance, a user could have a name badge with builtin RFID, a Bluetooth enabled cellphone, and a Bluetooth enabled PDA. Currently there are no provisions for this in the database. A new table would need to be created that allowed a mapping between many tag ids and one actual object/person.

Currently the system is limited by the processing power of the machine the tagd Server is running on. With a few changes to the tagd Server code, multiple servers could run concurrently on different machines and connect to the some database

sitting on a remote machine. If, at any time, the tagd Server starts to get bogged down by the amount of traffic going through it, another tagd Server could be added to take over some of the load. This would further increase the scalability of the system as a whole, while also moving away from a single point of failure situation.

Presently, if the tagd Server goes down, then the entire system comes to a halt. If multiple tagd Servers are running and one of them server goes down, only the portion of the system that was connected to that server would stop functioning. Therefore one of two design topologies could be implemented for the system. Every other room could be associated with a different tagd Server. So, if one server goes down, then the entire area can still able to be monitored, just not with as great of accuracy. Alternatively, a series of zones could be setup. For example, one tagd Server could cover clients that ran in all the rooms on floors 1 through 4, while another could cover floors 4 through 8.

Another addition to the system that would be nice in the event of tagd Server failure, would be some predictive and learning methods which could be used to guess where objects are. Also, if the tagd Server is running perfectly fine but there is no current location information for a particular object, predictive methods could be used to give the user of the system an idea of where the object might be. For instance, if an object moves to the corner of a large room and the base station cannot pick up the device anymore, the system will think that the device has no location associated with it. If none of the other rooms surrounding that room show that the device has moved through, the system could correctly guess that the device is still in the original room just out of range of the base station.

Furthermore, after a while the archive database would contain a large amount of data about the objects that are tracked. This information could be used to calculate statistics about the flow of objects through the environment. For instance, statistics about the location of users during the course of the day could be used to make guesses about where they are when the system is not able sense them. Many other statistics could be calculated as well such as percentage of time spent in certain rooms.

Appendix

Required Software

Software that must be installed and running properly is listed here. Version numbers are indicated that the system is known to work with. Most other versions should work, but are untested.

- GNU/Linux operating system (Kernel 2.4)
- Bluez Bluetooth Protocols Stack
- Apache (1.3.26)
- MySQL (3.23)
- PHP (4.0)
- MySQL++ (1.7.9)

Setup Procedure

After all the above software is installed and setup to work together, the next step is to create the database users "locadmin" and "locuser". The "locadmin" user will need full control of the two databases that will be created; locationdb and arlocationdb. The "locuser" user can only select data from the databases. Next, it is necessary to create the databases themselves and the tables in those databases. There is are two mysql formated dump files that can be used to construct the database entries. After the databases are setup, entries need to be added to the database for tags that will

be tracked and readers that are in the system. The mysql dump files contain example data, so there will already be some in the databases after adding them.

Now, installation of the web files is necessary. The php files that make up the web system need to be placed in a web accessible directory. Also, it is important to note that the password in the conf.php file must be set correctly in order for the php files to correctly connect to the database.

Then compilation of the three programs in the system is necessary. Makefiles are provided for each of the programs. They may need to be modified slightly for your particular system however. Once the programs are compiled, the system can be started by running the tagd program. The tagd program will be sitting there waiting for clients to connect to it.

Important Files

tagd - This is the executable is for the process that waits for connections from clients. This process receives data from the clients and loads it into the location database. This process also is responsible for handling the archiving of data.

rfid_reader - This executable is for the process that connects to the TCLI software and sends data for when a tag enters or leaves a region to the tagd process.

bluetooth_reader - This executable is for the process that scans the area for Bluetooth devices and sends data for when a device enters or leaves a region to the tagd process.

conf.php - Configuration for the parameters to the WeBOTS system web site.

func.php - A set of utility funtions for connecting to and querying the databases for the system.

refresh.php - An abstract set of javascript that can be included into any page that needs to periodically refresh the page.

index.php - Home page of the WeBOTS system. Lists the tags that are being monitored by the system, and offers a link to search the archives of the system.

track.php - Page that tracks a particular tag. It will refresh every so often looking to see if the tag's location changed. Also, offers a link to get all the archived data for the current tag being tracked.

archive.php - Page that shows all the archived entries for a particular tag or all the tags in the system.

search.php - Page that shows allows the user to search through all the entries in the archive database for specific information.

References

[1] 3com home page. http://www.3com.com.

[2] The apache http server project. http://httpd.apache.org.

[3] Bluez - official linux bluetooth protocol stack. http://bluez.sourceforge.net.

[4] Compaq home page. http://www.compaq.com.

[5] Iso - international organization for standardization. http://www.iso.org.

[6] The linux home page. http://www.linux.org.

[7] Microsoft corporation. http://www.microsoft.com.

[8] Mysql. http://www.mysql.org.

[9] Php: Hypertext preprocessor. http://www.php.net.

[10] Radio frequency identification (rfid) home page. http://www.aimglobal.org/technologies/rfid.

[11] Rfcode inc. home page. http://www.rfcode.com.

[12] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[13] Bluetooth Special Interest Group. *Specification of the Bluetooth System*, 2001.

[14] J. Haartsen, M. Naghshineh, J. Inouye, O. Joeressen, and W. Allen. Bluetooth: Vision, goals, and architecture, 1998.

[15] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[16] J. Hightower, C. Vakili, G. Borriello, and R. Want. Design and calibration of the spoton ad-hoc location sensing system, 2001.

[17] Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location systems for ubiquitous computing.

[18] James Kurose and Keith Ross. *Computer Networking: A Top-Down Approach Featuring the Internet.* Addison Wesley, 2001.

[19] Yiu Cho Lau. Landmarc: Location identification based on dynamic active rfid calibration, 2002.

[20] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: Indoor location sensing using active rfid, 2002.

[21] Abhishek Pramod Patil. Performance of bluetooth technologies and their applications to location sensing, 2002.

[22] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Mobile Computing and Networking*, pages 32–43, 2000.

[23] Simonas Saltenis and Christian S. Jensen. Indexing of moving objects for location-based services. In *ICDE*, 2002.

[24] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. pages 1–10.

[25] Ouri Wolfson, A. Prasad Sistla, Sam Chamberlain, and Yelena Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases*, 7(3):257–387, 1999.