INDOOR LOCATION SENSING USING WLAN

by

SATHIS KUMAR MOORTHY

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

JUNE 2007

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

INDOOR LOCATION SENSING USING WLAN

by

SATHIS KUMAR MOORTHY

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirements for the
Degree Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved by,

Pn. Norashikin Yahya
(Project Supervisor)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JUNE 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

SATHIS KUMAR MOORTHY

ABSTRACT

The purpose of this project was to produce an indoor location system using WLAN IEEE 802.11b in Access Point (AP) infrastructure setup without additional external hardware. The system was developed in MATLAB platform that comprises of location sensing algorithm (Nearest Neighbor), signal strength database, client-server interface and graphical user interface (GUI). The basic idea was to build a database of signal strength information for various predetermined locations. Using the Nearest Neighbor algorithm, this database can be used to determine a particular location provided that a set of signal strength is given. As for the location sensing system's performance, it had a precision in terms of average distance error of 5 m and an accuracy of 72.2 %. The justification of these results was that noise, interference, multi-path effect, indoor propagation loss and random movement in the environment affected RF signals.

ACKNOWLEDGEMENTS

Special acknowledgement and sincere gratitude to Ms. Norashikin Yahya the project supervisor for her assistance on the project, advice, guidance and motivation for the author towards achieving the goal of completing the project with all the objectives met.

The author would also like to express his fullest appreciation to his family for their moral support and absolute confidence in the author throughout the course of the project. Here, the author also refers a sincere gratitude towards to the technical staff namely Mr Musa and Miss Siti Hawa, who were certainly provided good technical advice and service such as lending and providing necessary equipment for experimenting purposes and preparing the workstation needed to undergo the project.

Last but not least, thanks to everyone involved directly and indirectly towards the success of this project. To sum it up, Rome was not built in a day or by only a person for that matter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

A wireless local area network (WLAN) uses radio communication to link two or more computers by accomplishing the same features as wired LAN [1] .WLAN utilizes spread spectrum technology based on radio waves to enable communication between devices in a limited area. This area is referred to as basic service set (BSS). A complete wireless LAN architecture is shown in Figure 1.1 [1]. The popularity of WLAN compared to wired-LAN is due to array of benefits ranging from cost efficiency to seamless integration with other networks. A wireless LAN offers users:

1. Convenience, affordability, and mobility
2. Flexibility, speed and scalability in installation
3. Regions without or with limited wired infrastructure can easily establish wireless communication
4. Wireless networks have a better chance of surviving disasters
5. 802.11 wireless LANs, WiMAX and 3G+ cellular networks promise high bandwidths, global mobility, quality of service and seamless integration with one another.

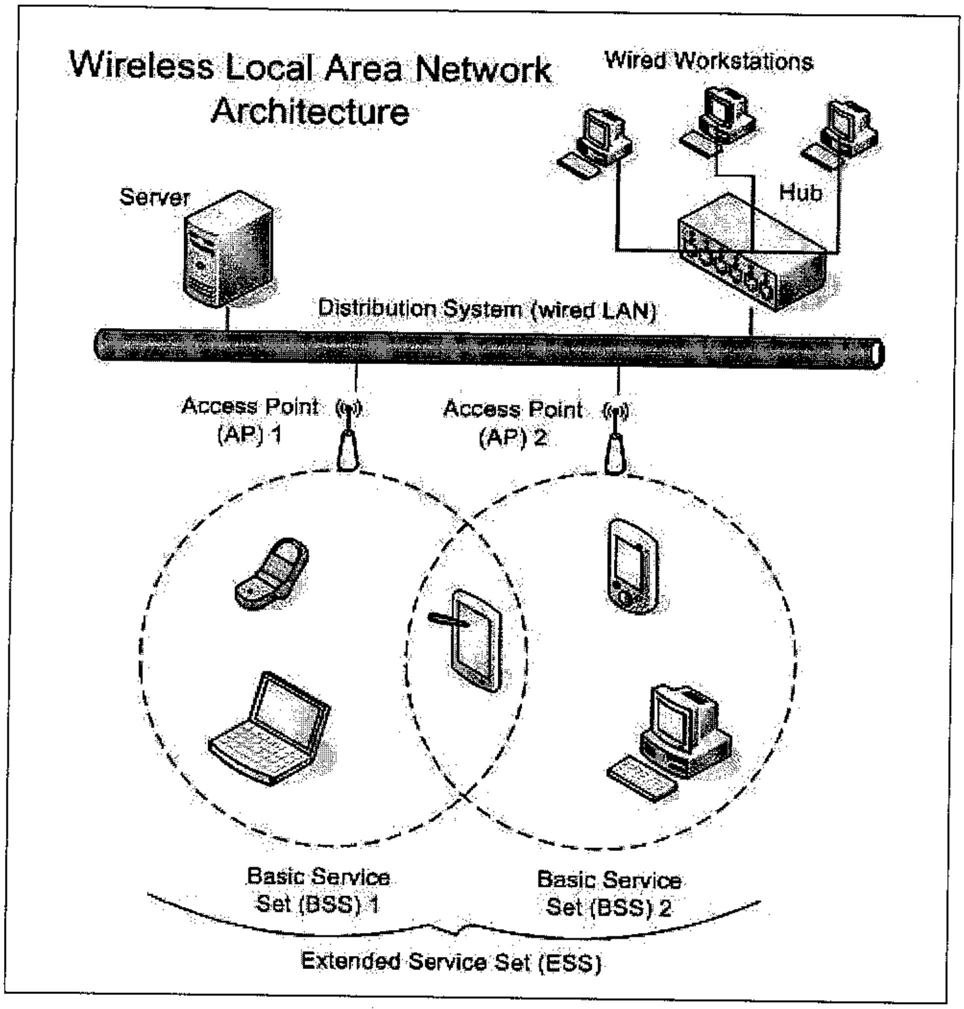

Wireless Local Area Network Architecture
Wired Workstations
Server
Hub
Distribution System (wired LAN)
Access Point (AP) 1
Access Point (AP) 2
Basic Service Set (BSS) 1
Basic Service Set (BSS) 2
Extended Service Set (ESS)

Figure 1.1: WLAN Architecture using an Infrastructure BSS

There are two types of wireless local area networks namely 'peer-to-peer' or 'Ad-Hoc' network and Access Point or Infrastructure network.

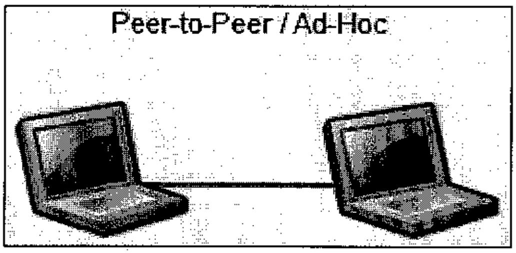


Figure 1.2: Ad-Hoc WLAN Network

An Ad-Hoc network, shown in Figure 1.2 [1], functions such that direct communication among wireless devices is allowed as long as these devices are within the range of each other. This eliminates the need for central access point, so two or more devices can connect to each other to form a network. However the disadvantage of this setup is that the received signal strength from all the Ad-Hoc devices could not be determined accurately. This is because the received signal strength refers to the strongest signal which is from the closest device in the network.

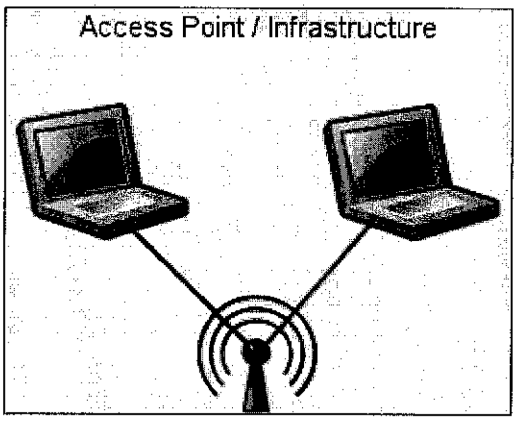


Figure 1.3: Infrastructure Wireless Local Area Network

In an infrastructure WLAN, shown in Figure 1.3 [1], network connection is established when a wireless enabled device (client) connects through an access point. An access point, often hub or router, has a built-in antenna that allows transmission or reception of radio frequency and bridges a wireless network to wired Ethernet network. A typical home network uses one access point that is directly connected to the internet to provide network with internet access. Meanwhile, larger networks applications utilize multiple access points at strategic locations.

1.2 Problem Statements

The influx in mobile computing devices and wireless LANs has prompted the significance of determining the location of a device at any point in time. Location information can be used for many applications. One application conceived would be to initiate the first few steps towards building context sensitive smart devices [2]. For example, a museum can use such a technology to build location-aware hand-held devices. Consider a visitor walks into a room, holding such device, information about exhibits in that particular room flashes on the screen of the device. Location sensing can be used in security applications [3]. Consider an intruder connects through a WLAN and begins transmitting packets (spam etc). The drawback in using WLAN is that the rogue machine could not be physically located and disconnected as in wired-LAN. In addition, the only information that can be obtained is that the rogue machine is associating with given base station. This information does not provide the physical location of the rogue machine. Regardless of whether the network is using common forms of wireless encryption, the wireless network such as 802.11b are subject to additional class of attacks by the rogue machine. Due this security reasons, location sensing or determination is a priority for the network administrator.

1.3 Objectives and Scope of Study

1.3.1 Objectives of the Project

The objective for this project is to produce an indoor WLAN location sensing system that can be used in any WLAN infrastructure environment. The project goals include but are not limited to the following:

- To be able to determine the location of a static device communicating in WLAN environment given needed information which is received signal strength data
- Utilizing the features in WLAN IEEE 802.11g without the need for additional hardware

1.3.2 Scope of Study

Basically, the study needed for this project will focus on the integral parts of a location sensing system as the following:

1. The location sensing system will involve interaction between client and local location server, so an efficient system (client-server) model and the communication interface to be used will be studied upon.
2. Commercially, signal strength information shown in communicating devices is through various interfaces that are in qualitative representation. However, this project requires a numerical value of signal strength. Therefore the study will focus on finding a reliable method of obtaining numerical value of signal strength.
3. There are many location sensing methods and algorithms that could be used. However, a reliable location sensing method or algorithm that yields results which has small degree of error is required. Therefore the study will focus on location sensing methods and algorithms.

4. The implementation of this system requires the usage of database that will be used to store received signal strength data. Currently there are database applications such as ‘Microsoft Excel’ and ‘Access’, ‘MySQL’ and others. The focal study point would be to choose the software that can be linked to ‘MATLAB’ software without complications.

CHAPTER 2
LITERATURE REVIEW

2.1 Ad-Hoc Network

One of the feasible papers by S. Li et al in [4] is a set up over an 802.11b Ad-hoc Network. The method used is based on triangulation utilizing empirical radio propagation. Through this method, there is no Access Points required and they relied wholly on signal strength data. The advantage of this approach is that the location sensing system can be utilized in a temporary mode. However, this approach has its drawbacks where the approach can only be used for outdoor location sensing and storing of huge training data and classification is infeasible. Since this project will focus on location sensing in indoor environment, the method used in Ad-Hoc setting will not applicable.

2.2 Access Point (AP) Infrastructure Network

Basically, all the current papers on indoor location sensing manipulate the fact that signal strength (power) received at different points of distance has different values. To be precise, as log distance of communicating device increase, the received signal strength (power) decreases.

S. Saha et al in [2] built a location sensing system in AP infrastructure in indoor environment. The approach method in this paper is by utilizing 3 Access Points via non-overlapping frequency channels 1, 6, and 11. Then, a database of signal strength is built by capturing the signal strength data at the access points (while in communicating with the device). The data is acquired using 'Sniffer' utility software. Then, they tested 2 location sensing algorithms namely 'Nearest Neighbor Classifier' and 'Back Propagation Neural Network'.

a) Nearest Neighbor Classifier

Take into consideration of a sample containing signal strength information of 3 APs at a given time interval. Denote these information as c_1, c_2, and c_3 (average of first, second and third channels). Then, the mean signal strength for each channel, M_j (c_{j1}, c_{j2}, c_{j3}) at each location is calculated. This constitutes the signal strength profile of a given location. Now given a test sample, the mean of the three channels, M' (c'_1, c'_2, c'_3). Then the classifier would determine that the test data belongs to location j if:

$$dist(M_j, M') \leq dist(M_k, M'),\ if\, k \neq j \tag{2.1}$$

To be precise, the classifier chooses the class whose center is nearest in feature space to mean of the current data. The distance used is Euclidean distance.

b) Back propagation neural network

A neural network is a collection of neurons arranged in a particular manner to form a network. A back propagation neural network is a multi-layer feed forward neural network. This network typically consists of several layers that consist of a set of source nodes comprising the input layer, one or more hidden layers and an output layer of neurons. Training of such a network is done by adjusting the synaptic weights such that given a particular input; the network provides a particular output. For a back-propagation network, training is done by using an iterative-algorithm called the error back-propagation algorithm.

The back-propagation neural network has 3 inputs, one for the mean of signal strength of each channel. It has N+1 hidden nodes and an output layer consisting of

N nodes, where N is the number of training points. The output of this network is an n-dimensional vector, which gives the probability of the input data belonging to each class. The neural network is trained by signal strength samples from each of these N training, after preprocessing the data to remove the outliers. Each vector used to train the neural network is the average signal strengths obtained from each access points. For testing process, the average of signal strengths obtained from different access points are used to construct the input vector. Then, these data is fed into the neural network to obtain the location.

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Procedure Identification

In order to ensure smooth and steady progress for the completion of this project, the work is conducted in a flow of steps as illustrated in Figure 3.1. The preliminary research and literature review done was discussed in the previous section. For this project, the tools used are as follows:

- 3 Access Points that uses IEEE 802.11b Protocol
- A laptop
- Wireless network adapter 802.11b (client). The specification of the wireless adapter is included in Appendix I.
- Communication interface softwares ('mIRC v6.14' and 'Unreal IRCD 3.2')
- Signal strength extracting softwares (Netstumbler and Cisco Aironet Site Survey Utility)
- Desktop PC (server)
- MATLAB v6.5.1 software

The other tasks involving the project are discussed in the following sections.

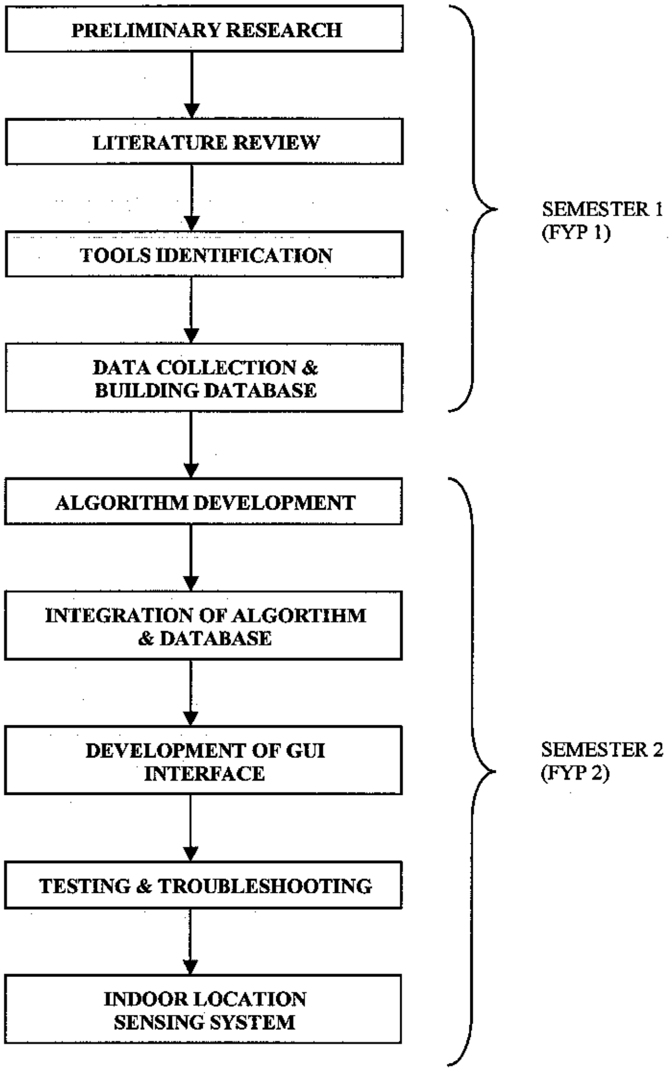

PRELIMINARY RESEARCH
LITERATURE REVIEW
TOOLS IDENTIFICATION
DATA COLLECTION & BUILDING DATABASE
ALGORITHM DEVELOPMENT
INTEGRATION OF ALGORTIHM & DATABASE
DEVELOPMENT OF GUI INTERFACE
TESTING & TROUBLESHOOTING
INDOOR LOCATION SENSING SYSTEM
SEMESTER 1 (FYP 1)
SEMESTER 2 (FYP 2)

Figure 3.1: Project Methodology Flowchart

3.2 Project Approach

The indoor WLAN location sensing system will be implemented in the Access Point (AP) infrastructure environment. A configuration or setup as shown in Figure 3.2 is deployed on the 2nd floor of Building 23 (EE).

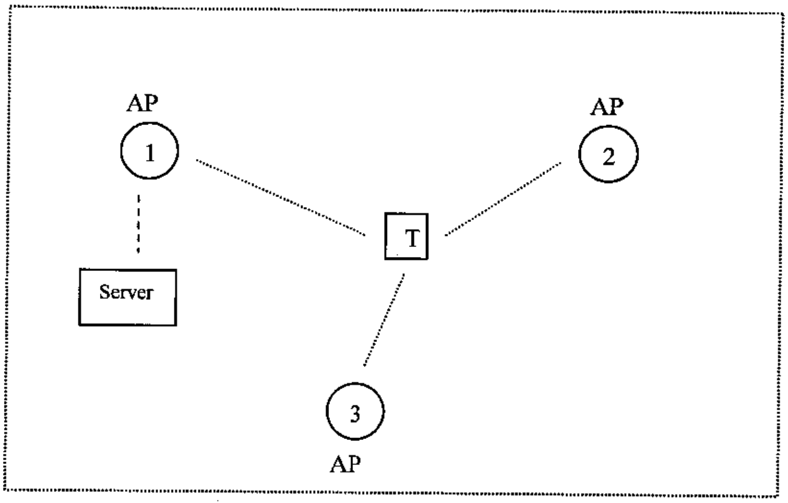


Figure 3.2: WLAN location sensing setup

Figure 3.2 show WLAN location sensing setup where three APs operating on non-overlapping frequency channels (1, 6 and 11) is being setup in a triangular pattern so that optimal working condition can be achieved. Here, the target device (T) is the client that will be communicating through this WLAN setup. The location sensing system functions such that the target device will query the server (system) about its location as needed by providing the signal strength profile at its current location. Then, the server will receive this query and process the signal strength profile received from the target device. The data processing is done by using location sensing algorithm that makes comparisons with the signal strength database. The output of this algorithm will be the estimated location or probable location of the target device. Next, the server will send this location information to the target device.

3.3 System Model

As mentioned previously, the WLAN location system will be built based on client-server approach. There are two such approaches namely active client-server approach and passive client-server approach [6]. The passive client approach will not be discussed since this project will utilize the active client approach.

a) Active Client-Server Approach:

In this approach, the client-server interaction will be a two way route whereby each of them has predetermined tasks and responsibilities. Figure 3.3 shows the active client-server approach with basic tasks of the client and server stated. Meanwhile, Table 3.1 [6] describes their tasks and responsibilities in detail.

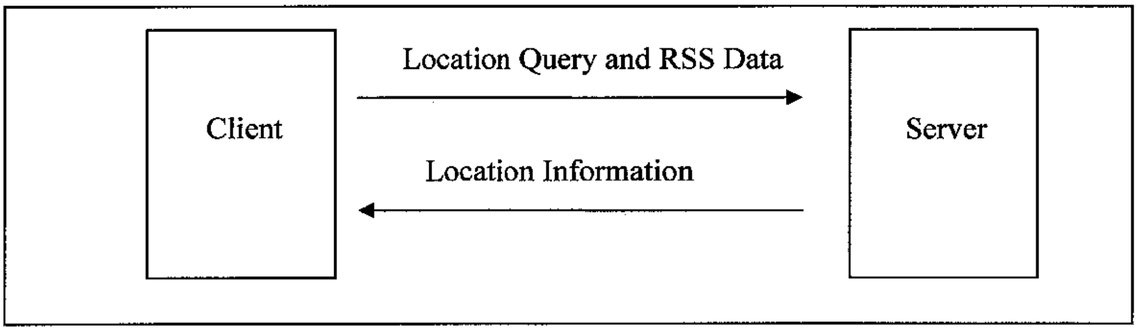


Figure 3.3: Active Client Server Model

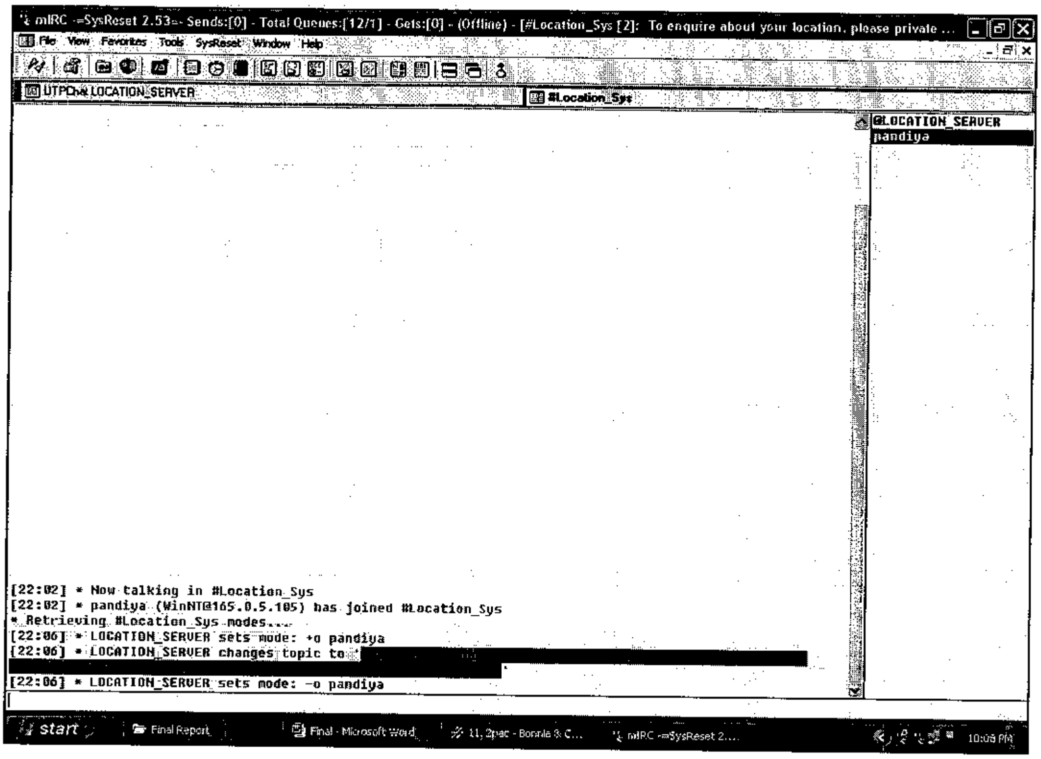
Table 3.1: Client and server responsibilities in the active client approach

Client Tasks	Server Tasks
Collect the signal strength	Accept incoming queries
Converge or average the signal strength values	Compare the observation vector with training database using algorithm
Construct signal strength profile and send a query to the server	Arrive at a probable location
Receive the location information from server	Return a location estimate or the probable location

In addition to those above, the client device is expected to run an application that is capable of collecting signal strength values. The primary advantage of this approach is that the system can be built completely from free software and inexpensive commodity hardware.

b) Communication Interface

Since the system involves interaction between client and server, a communication interface is required. For this purpose, 'mIRC v6.14' and 'Unreal IRCD 3.2' softwares that are commercially available freeware softwares are used. Basically, the 'mIRC v6.14' software, shown in Figure 3.4, is used as the communication medium whereby the client device will connect through using a provided server address. Then the client will join a channel named 'Location_Sys' by using */j #Location_Sys* command. From the right column (nickname column) in Figure 3.4, it is seen that are two nicknames namely "LOCATION_SERVER" and "pandiya" which refers to the server and client respectively. The instructions to enquire for location information are provided in the channel's topic.


mIRC -=SysReset 2.53=- Sends:[0] - Total Queues:[12/1] - Gets:[0] - (Offline) - [#Location_Sys [2]: To enquire about your location, please private ...
File View Favorites Tools SysReset Window Help
UTPChat LOCATION_SERVER
#Location_Sys
@LOCATION_SERVER
pandiya
start
Final Report
Final - Microsoft Word
11, 2pac - Bonnie & C...
mIRC -=SysReset 2....
10:08 PM

```
[22:02] * Now talking in #Location_Sys
[22:02] * pandiya (WinNT@165.0.5.105) has joined #Location_Sys
* Retrieving #Location_Sys modes....
[22:06] * LOCATION_SERVER sets mode: +o pandiya
[22:06] * LOCATION_SERVER changes topic to '[redacted]
[22:06] * LOCATION_SERVER sets mode: -o pandiya
```

Figure 3.4: 'mIRC v6.14' software

Meanwhile, the 'Unreal IRCD 3.2' software is used to set up the 'mIRC' server functionality for the server computer. This software functions such that the IP address of the server computer is automatically used as the server address. The layout of this software is shown in Figure 3.5.

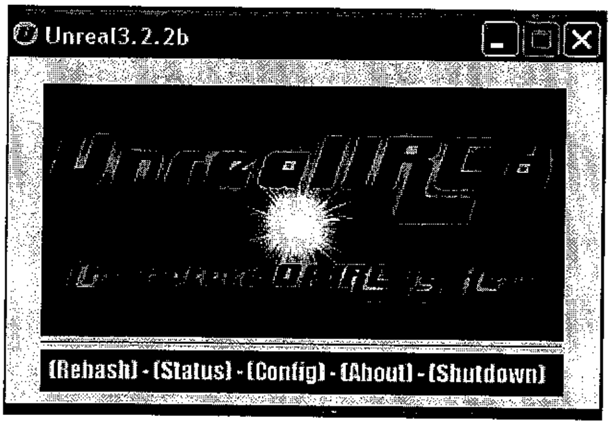
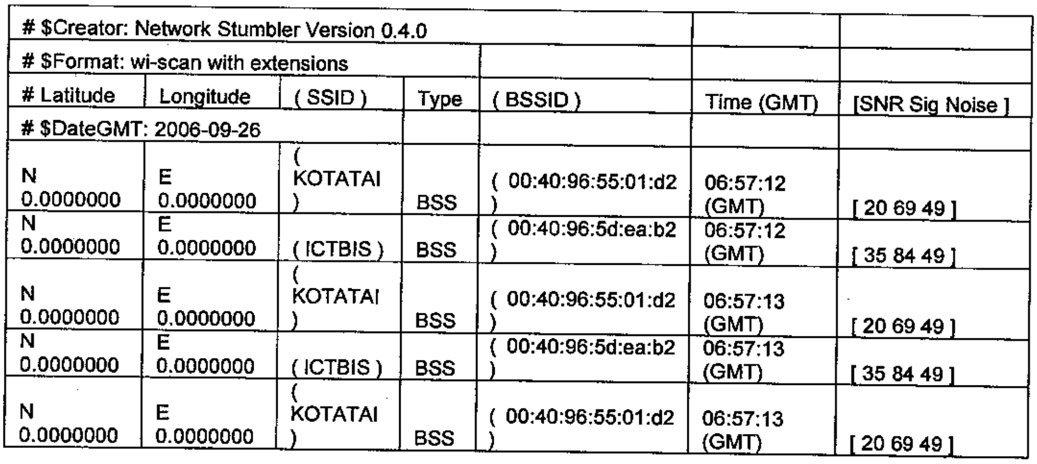


Figure 3.5: 'Unreal IRCD 3.2' software

3.4 Data Acquiring

a) Netstumbler

The data collection using 'Netstumbler' is done by opening the software and initiating the wireless scan. After a predetermined period, the scanning is stopped and the wi-scan file is exported. Then this wi-scan file is opened using Microsoft Excel. Figure 3.6 shows the wi-scan file obtained using 'Netstumbler' software. The column service set identifier (SSID) indicates the identification of access points present and the basic service set identifier (BSSID) column shows the media access control (MAC) address of the access point. The last column shows the SNR, signal value, and noise value at the certain point of measurement. The signal strength value is in dBm.



# $Creator: Network Stumbler Version 0.4.0						
# $Format: wi-scan with extensions						
# Latitude	Longitude	(SSID)	Type	(BSSID)	Time (GMT)	[SNR Sig Noise]
# $DateGMT: 2006-09-26						
N 0.0000000	E 0.0000000	(KOTATAI)	BSS	(00:40:96:55:01:d2)	06:57:12 (GMT)	[20 69 49]
N 0.0000000	E 0.0000000	(ICTBIS)	BSS	(00:40:96:5d:ea:b2)	06:57:12 (GMT)	[35 84 49]
N 0.0000000	E 0.0000000	(KOTATAI)	BSS	(00:40:96:55:01:d2)	06:57:13 (GMT)	[20 69 49]
N 0.0000000	E 0.0000000	(ICTBIS)	BSS	(00:40:96:5d:ea:b2)	06:57:13 (GMT)	[35 84 49]
N 0.0000000	E 0.0000000	(KOTATAI)	BSS	(00:40:96:55:01:d2)	06:57:13 (GMT)	[20 69 49]

Figure 3.6: Netstumbler wi-scan file

b) Cisco Aironet Site Survey Utility

The data collection is done by opening this software and clicking on the 'AP Log Snapshot' button. This step can be repeated several times or by setting automatic logging in the software's option. Once the snapshot button is clicked, the software will create a text file in the software's directory. Figure 3.7 shows AP log text file obtained using 'Cisco Aironet Site Survey Utility' software. In every line, the details provided are the time stamp of data collected, AP SSID, MAC address, received signal strength in dBm, connection type, AP frequency channel and the data rate.

```
2007-04-05 15:25:29 ,23-02-03_Tech Room,00:15:E9:AF:1C:2E,-87,Secure,G,6 (2437),54,,,,""
2007-04-05 15:25:29 ,belkin54g,00:17:3F:0E:0E:C8,-87,Open,G,11 (2462),54,,,,""
2007-04-05 15:25:29 ,datacomm_lab02_belkin54g,00:17:3F:0E:0C:86,-87,Secure,G,1 (2412),54,,,,""
2007-04-05 15:25:59 ,23-02-03_Tech Room,00:15:E9:AF:1C:2E,-65,Secure,G,6 (2437),54,,,,""
2007-04-05 15:25:59 ,belkin54g,00:17:3F:0E:0E:C8,-93,Open,G,11 (2462),54,,,,""
2007-04-05 15:25:59 ,datacomm_lab02_belkin54g,00:17:3F:0E:0C:86,-59,Secure,G,1 (2412),54,,,,""
```

Figure 3.7: AP log snapshot text file

3.5 Signal Strength Database

The signal strength database is an integral part of the server functionality. In order to start building signal strength database, three APs are setup in Building 23 2nd Floor which has a dimension of 75 x 28.4 meters. Initially, the radio propagation mapping was to be done by dividing up the floor into 5 x 5 meters of grid. However, the width of the floor did not permit the usage of this preplanned grid. So, an approximated 5 x 5 meter grid is implemented that divided the second floor into 71 locations (subsections). Next, the radio mapping (data collection) is started. The data collection is done by remaining static at one of these sections or locations mentioned above. Then, data is collected using 'Netstumbler' software and 'Cisco Aironet Site Survey Utility' software. After collecting data for 1 to 2 minutes in one location, these steps are repeated for the next section or location. The setup of the radio mapping is shown in Figure 3.8 below. The triangles represent the access points that are setup on that floor.

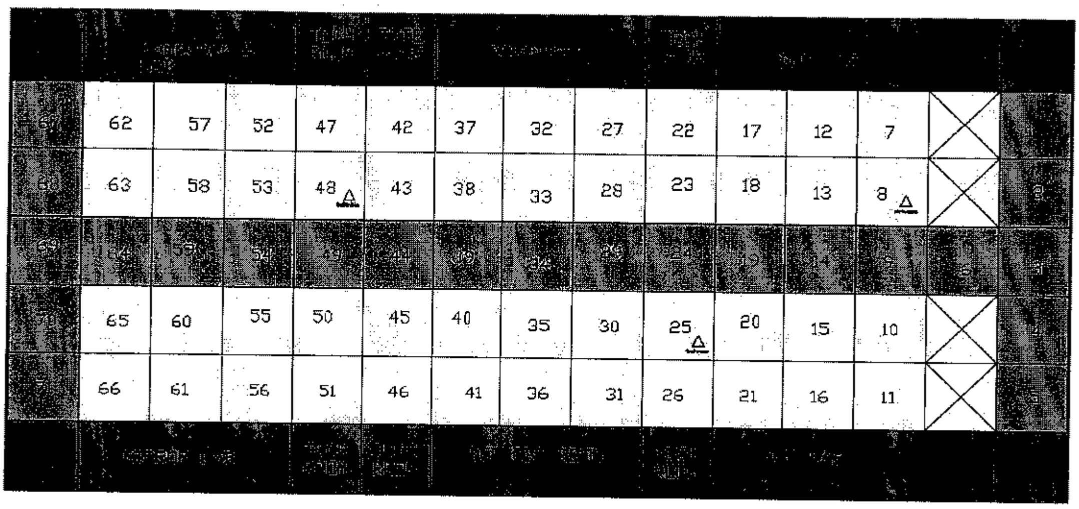
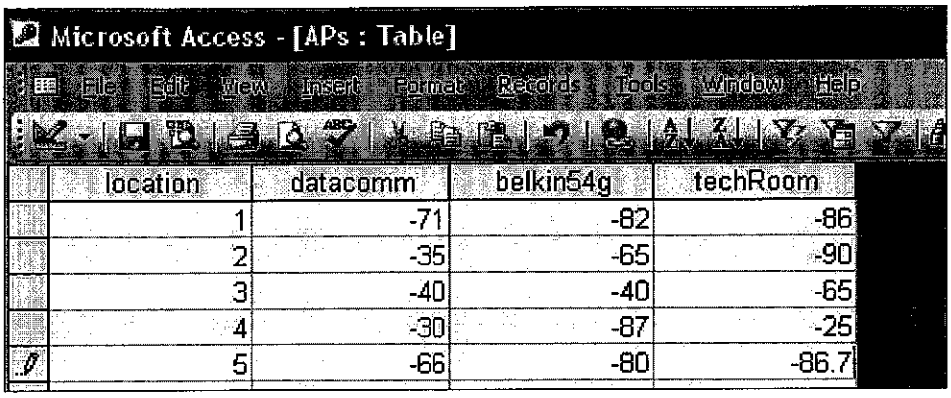


Figure 3.8: Radio Mapping Layout of Building 23 2nd Floor

The shaded portion (locations) in Figure 3.8 represents the hallway in that floor that consists of exits and stairways. During the radio mapping session, data collection could not be done in a few of the locations. This is because those locations are limited access rooms or locations (authorized person only). Those locations are shown in Figure 3.8 by the crossed locations that comprises of locations 6, 7, 9, and 10.

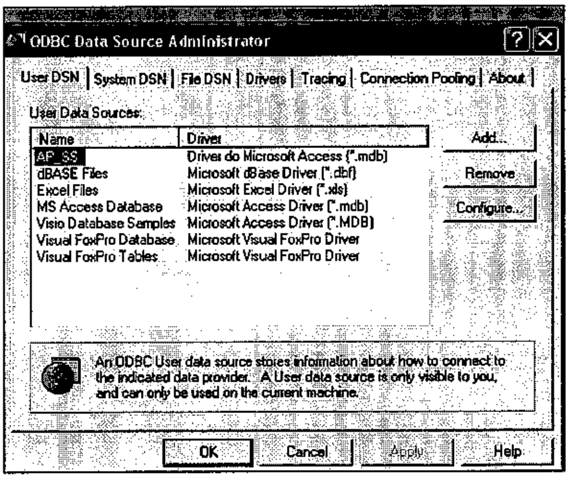
After the radio propagation is completed, the data collected is processed by averaging the recorded received signal strength values. Next, these processed data is compiled into a database employing a table structure in Microsoft Access. Figure 3.9 below illustrates a portion of this database. The first column shows the location as depicted in figure 3.8 and the remaining columns contains received signal strength from the three APs.



location	datacomm	belkin54g	techRoom
1	-71	-82	-86
2	-35	-65	-90
3	-40	-40	-65
4	-30	-87	-25
5	-66	-80	-86.7

Figure 3.9: Signal Strength Database using Microsoft Access

Since the system will be developed in the 'MATLAB 6.5.1' platform, the signal strength database has to be added as a data source using 'ODBC Data Source Administrator'. This is so that the 'Database Toolbox' will be able to recognize the signal strength database and the linking process with 'MATLAB 6.5.1' will be successful. The ODBC Data Source Administrator dialog box is shown in Figure 3.10.



User Data Sources:

Name	Driver
AP_SS	Driver do Microsoft Access (*.mdb)
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
MS Access Database	Microsoft Access Driver (*.mdb)
Visio Database Samples	Microsoft Access Driver (*.MDB)
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Figure 3.10: ODBC Data Source Administrator

3.6 Location Sensing Algorithm

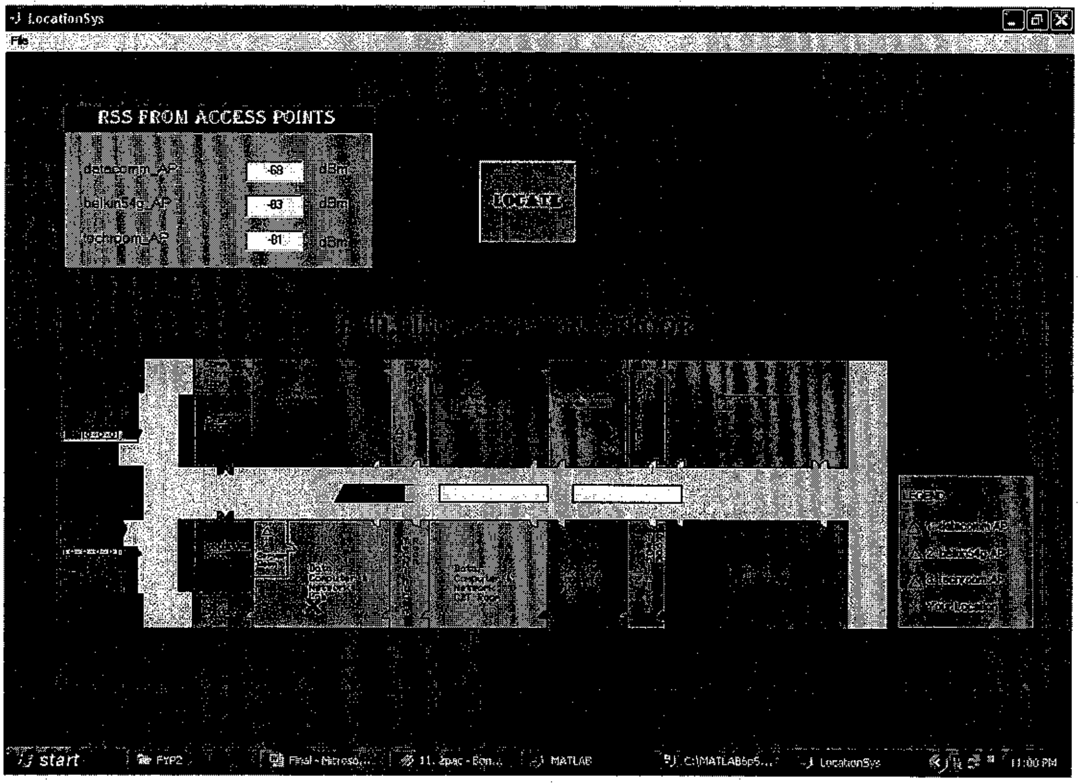
Location sensing algorithm is another vital component in the system. The algorithm functions to accept observation data (query) and compare it according to the signal strength database and determine the estimated or probable location. The Nearest Neighbor algorithm is used as the location sensing algorithm. The working principle of this algorithm is that the received signal strengths from APs for an unknown location is taken and inputted to this algorithm. The signal strengths are taken as the coordinates of a point in feature space that represent this unknown location. In this feature space, there are other points that represent all the locations in the database. The distance between the unknown location point and all database location points are calculated. The distance used here is Euclidean distance. The database location point which has the minimum distance value is chosen as the approximated location of this unknown location. To be precise, the algorithm chooses the class whose center is nearest in feature space to the unknown location data. The pseudocode of this algorithm is shown in Figure 3.11 and the MATLAB coding for this algorithm is included in Appendix II.

```
get a list of all the locations in the training database
    for each of these locations, get the list of access points and
    their associated signal strength values
        for each of these access points, find the difference and
        square this difference value
    sum the squared  difference value and find the square root
using the minimum value, the estimated location is projected
```

Figure 3.11: Nearest Neighbor Algorithm pseudocode

3.7 Graphical User Interface (GUI)

In order to instill user friendly functionality, graphical user interface (GUI) is developed. This feature is included so that, the server administrator has an easier task of conducting location sensing. The layout of the GUI is shown in Figure 3.10 and the MATLAB coding of the GUI and map plotting coding is included in Appendix III and IV respectively.



RSS FROM ACCESS POINTS

datacomm_AP -68 dBm

belkin54g_AP -83 dBm

techroom_AP -81 dBm

Figure 3.12: Graphical User Interface (GUI)

Referring to Figure 3.12, the GUI consists of three main sections that are the input section labeled 'RSS FROM ACCESS POINTS', activation section via pushbutton labeled 'LOCATE' and the map labeled 'BUILDING 23 SECOND FLOOR'. In addition, there is a pull down menu section labeled 'File' where two functions are available namely 'Save Location' and 'Close'. The GUI is operated by inputting received signal strength (RSS) from the access points. Then, the pushbutton is pushed to start the location sensing algorithm. After a few seconds, the queried location will be projected on the map section with 'X' marking the location. By selecting 'Save Location' in the pull down menu, the GUI saves the map as a jpeg file named 'locate.jpg'. Meanwhile, selecting 'Close' will close the GUI figure.

CHAPTER 4
RESULTS & DISCUSSION

4.1 Signal Strength Database

Since the building of database involved wireless propagation, a study is carried out as a confirmation that the practical results of received signal strength measured adheres to wireless propagation theory. The theory underlying this study is discussed in the following arguments.

$$P_r(dBm) = P_t(dBm) - L_p(dB) \qquad (4.1)$$

Equation 4.1 shows the basic modeling equation for wireless signal propagation. P_r is the power received (receiver) at a certain point of consideration measured in dBm. P_t is the signal power transmitted from the transmitter measured in dBm and L_p is the propagation path loss in wireless communication. As for this study, P_t corresponds to the transmitting power of the access point and P_r is the signal strength received by the client device. However the indoor propagation path loss is not simplistic in nature and can be subdivided into path loss components as shown in equation 4.2 [8] below.

$$L_p(dB) = \beta(dB) + 10\log_{10}\left(\frac{d}{d_0}\right)^n + \sum_{p=1}^{P} WAF(p) + \sum_{q=1}^{Q} FAF(q) \qquad (4.2)$$

Here, β represents the measured path loss at the reference distance d_0, typically 1 meter. In the absence of other information, β is taken to be free-space path loss at distance 1 meter from transmitter. The path loss exponent is denoted as *n,* which values typically ranging from 2 to 5. *WAF (p)* is the wall attenuation factor and FAF *(q)* is the floor attenuation factor. *P* and *Q* are the number of walls and floors separating the transmitter and receiver. Assuming β, *WAF (p)*, and *FAF (q)* are constant and substituting equation 4.2 into equation 4.1, the simplified wireless propagation modeling equation 4.3 is obtained.

$$P(d)(dBm) = P(d_0)(dBm) - 10\log_{10}\left(\frac{d}{d_0}\right)^n \qquad (4.3)$$

Figures 4.1, 4.2 and 4.3 depict the results of this study for all the three APs utilized on the 2nd floor of Building 23.

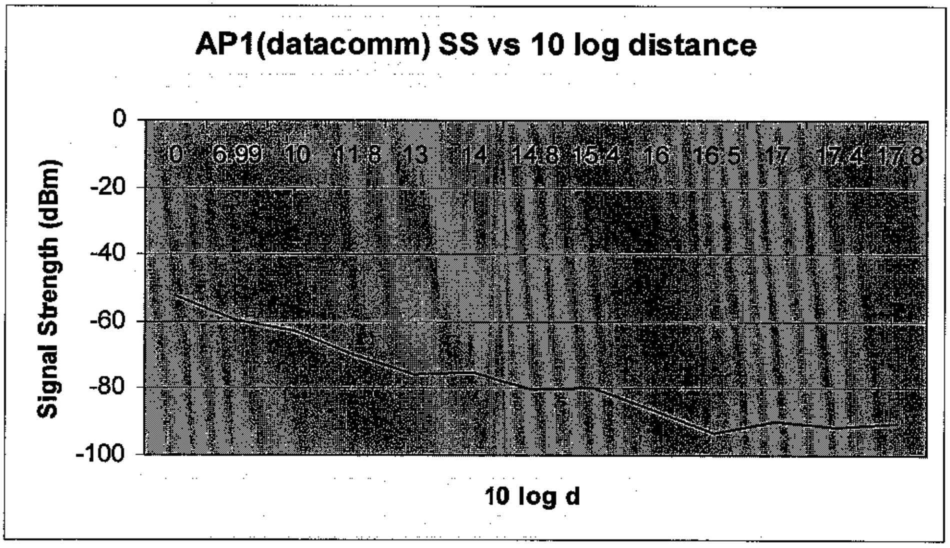


Figure 4.1: AP1 (datacomm) SS vs 10 log distance

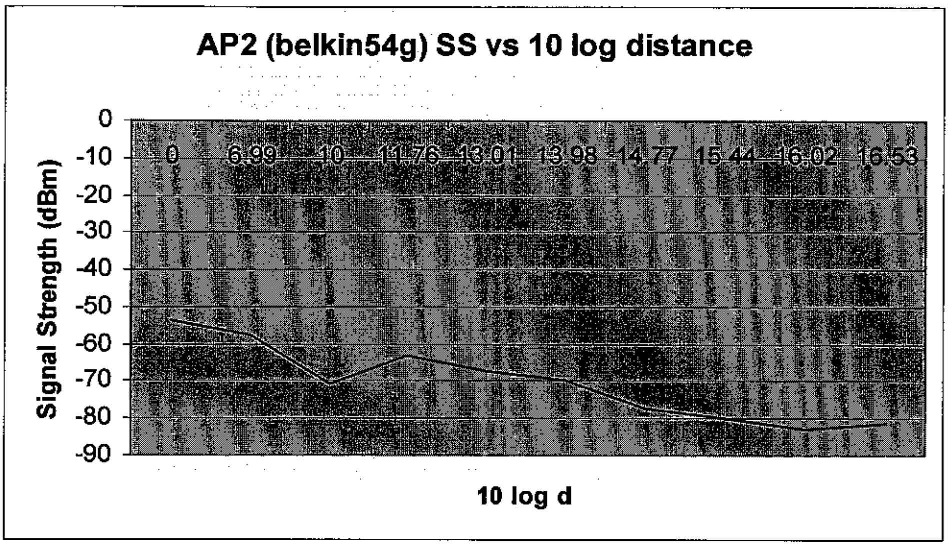


Figure 4.2: AP2 (belkin54g) SS vs 10 log distance

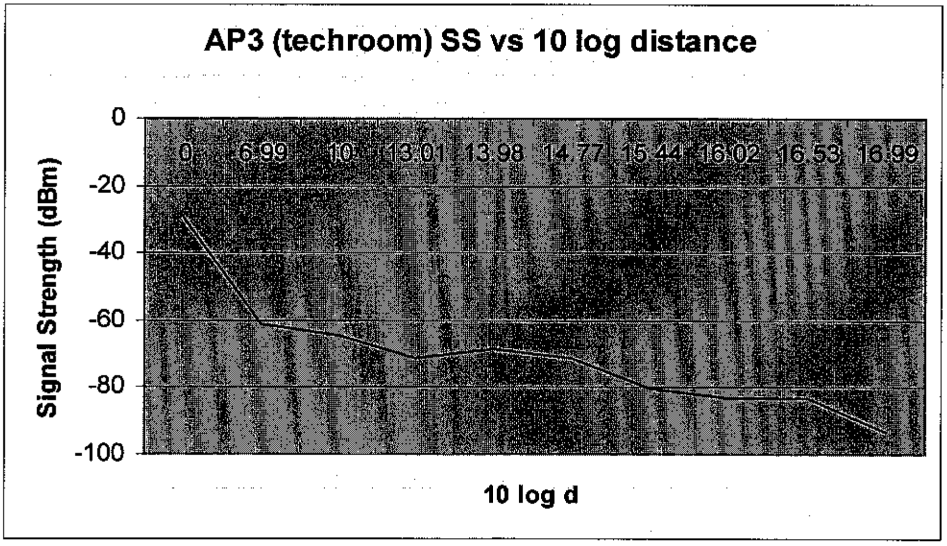


Figure 4.3: AP3 (techroom) SS vs 10 log distance

Observations on the results of this study indicate that received signal strength from access points (AP) decreases with the increase in distance from the AP. However, the received signal strength seems to be fluctuating due to obstructions, interference, reflection and diffraction. This result is expected since the Building 23 2nd floor consists of separation walls for all the laboratory and rooms. In addition to that, the signal strength was also affected by computers and equipments placed in the laboratories. Even so, it is seen that the results are following the indoor propagation model marginally which is adequate enough for this project.

4.2 Location Sensing Algorithm

A testing session is conducted to check the Nearest Neighbor algorithm functionality and location sensing capability. The testing procedure is done by collecting received signal strength data at random locations on the Building 23 2nd floor. This is done as sort of simulating real time event whereby a client is querying for the location on the floor. However, for testing purpose, these random locations are actually known to the tester, so that qualities such as average error distance in terms of precision and accuracy of the algorithm can be evaluated. The results of this testing session are shown in Table 4.1.

Table 4.1: Nearest Algorithm Testing Results

Trial (N_i)	Average Received Signal Strength (dBm)	Real Location	Estimated Location	Error Distance (m)
1	(-68, -83, -81)	12	12	0
2	(-73, -84.75, -82.67)	3	1	10
3	(-60.3, -88.5, -65)	20	19	5
4	(-80.4, -71.6, -78.2)	33	38	5
5	(-88.5, -58.5, -79.5)	43	42	5
6	(-88.4, -76, -64.4)	35	31	5
7	(-73.3, -73.5, -68)	39	39	0
8	(-68.25, -79.25, -81)	17	12	5
9	(-93.8, -78.4, -84.8)	56	61	5
10	(-89, -81, -88.75)	61	68	OOR
11	(-89.25, -80.75, -76.67)	45	40	5
12	(-52.9, -85.4, -79.2)	8	8	0
13	(-74.8, -71, -78.6)	23	27	5
14	(-81, -72.6, -75.8)	27	38	10
15	(-80.5, -46, -86.5)	47	47	0
16	(-95, -60, -84)	53	42	10
17	(-94, -65.2, -92.8)	62	57	5
18	(-97, -71, -89.75)	65	63	10

In Table 4.1, the second column refers to the averaged received signal strength whereby the values in the bracket (RSS_1, RSS_2, RSS_3) contains the average received signal strength from access points 'datacomm', 'belkin54g' and 'techroom' respectively. The location numeric are based upon the setup as in Figure 3.8 discussed previously. Meanwhile, in the last column OOR (out of range) refers to error distance more than 10 m.

Average Error Distance, A_e is a qualifying method that is used to evaluate the precision of the algorithm in accordance to distance. Let N_T be the number of trials conducted and N_{OOR} be the number of trials where the algorithm yields an out of range (ORR) location. Then the Average Error Distance is defined as in equation 4.4 [2].

$$A_e = \frac{\sum_{i=1}^{N_T - N_{ORR}} dist(l_i, y_i)}{N_T - N_{OOR}} \tag{4.4}$$

Here, l_i refers to the real location of the test data and y_i the estimated location for the i'th trial and *dist* (l_i, yi) refers to the separation distance between these locations. As for the testing results discussed earlier, there were 4 trials that yielded error distance of 0 m, 9 trials with 5 m error distance, 4 trial with 10 m error distance and 1 trial with ORR (out of range) error distance. After performing calculations, the Average Error Distance, A_e was found to be 5 m.

Another evaluation method for the algorithm used was by calculating the accuracy using equation 4.5. In equation 4.5, N_{TAe} refers to the number of trials for which the error distance is equal or less than the calculated Average Error Distance, A_e. For this calculation, N_{TAe} is found to be 13.

$$Accuracy = \frac{N_{TAe}}{N_T} \times 100\% \tag{4.5}$$

Calculations done revealed that the algorithm performs at an accuracy of 72.2%. The justification of these results is that the performance of the algorithm relies totally on the accuracy of signal strength measuring software and also the performance of the wireless adapters.

CHAPTER 5
CONCLUSION & RECOMMENDATION

In this project, the indoor location sensing system is implemented via WLAN IEEE 802.11b AP infrastructure environment without the usage of additional external hardware. The location sensing system used the Nearest Neighbor algorithm that performed its location determination task at an average error distance of 5 m and an accuracy of 72.2%. However, the ability to use a WLAN IEEE 802.11g for indoor location sensing has been demonstrated provided that received signal strength data is available.

There are certainly issues that need to be further investigated with regards to precision and accuracy. In terms of precision, reducing average distance error, a recommendation would be to increase the current number of APs so that the there will be more signal strength data available for each location. Hence, each location will become more unique leading to a more accurate location determination. Next recommendation is to increase accuracy of the system by implementing another location sensing algorithm. One such algorithm is the back-propagation neural network algorithm that was discussed in Literature Review section.

Another recommendation is that, passive client-server approach is used. By utilizing this approach, the location sensing system will be based on intelligent access points (AP) whereby the AP scans for clients. Then, these APs will send the signal strength information of the clients to the server to be updated. Hence, any queries made by the clients will no require the client to provide signal strength profile data. However, for this functionality, highly customizable APs would be used where third party software can be installed to perform client scanning and information sending.

REFERENCES

[1] http://en.wikipedia.org/wiki/Wireless_LAN, 10 August 2006

[2] S. Saha, K. Chaudhuri, D. Singh, and P. Bhagwat ,"Location Determination of a Mobile Device Using IEEE 802.11b Access Point Signal", Wireless Communications and Networking, 2003, Volume 3, 16-20 March 2003 Page(s):1987-1992 vol.3

[3] P. Tao, A. Rudys, A.M. Ladd, and D.S. Wallach, "Wireless LAN Location-Sensing for Security Application"

[4] S. Li, G. Zhao and L. Liao, "User Location Service over an 802.11 Ad-Hoc Network"

[5] A. Hatami and K. Pahlavan, "In-Building Intruder Detection for WLAN Access"

[6] A. Harder, L. Song, and Y. Wang, "Towards an Indoor Location System Using RF Signal Strength in IEEE 802.11b Networks"

[7] Cisco Aironet 350 Series Access Points Data Sheet

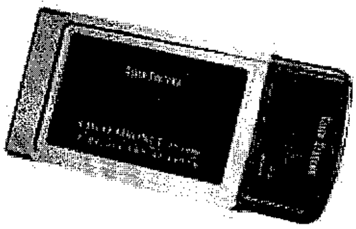
[8] Simon Haykin and Micheal Moher, *Modern Wireless Communications*, Pearson Education Inc, 2005

APPENDIX

APPENDIX I: Wireless Adapter Specification

<u>Cisco Air-PCM 352</u>

Technical Details



- **Model**: AIR-PCM352
- **Standard:** 2.4GHz/802.11b wireless standard.
- **Hardware Platform**: PC, Mac, Unix
- **Networking Feature**: Network adapter
- **Form Factor**: Plug-in module
- **Network Interface Description**: DSSS Type II PC Card
- **Connectivity Technology**: Wireless
- **Security:** Up to 128-bit WEP encryption
- **Data Link Protocol**: IEEE 802.11b
- **Network Data Transfer Rate**: 11 Megabits Per Second
- **Maximum Range Indoors**: 350 feet (1Mbps) , 130 feet (11 Mbps)
- **Minimum system requirements:** Microsoft DOS, Microsoft Windows 95/98, Linux, Microsoft Windows CE 2.11, Microsoft Windows CE 3.0, Microsoft Windows 2000 / NT4.0, Microsoft Windows Millennium Edition, Apple MacOS 9.x, Apple MacOS X, Microsoft Windows XP
- **Width**: 4.4 inches
- **Depth**: 2.1 inches
- **Height**: 0.1 inches
- **Weight**: 1.6 Ounces
- **Warranty**: Limited lifetime warranty

```
      %sum all the squared difference values in 'diff' array for the
      %'i'th row and columns 1,2 & 3, then find the square root of
      % this sum. Next store in array 'distT'

    distT(i,1)=sqrt(dist(i,1)+dist(i,2)+dist(i,3))

end

[Y,l]=min(distT)     %find the minimum value (Y) and the row location
                     %of this value (I) from array 'distT'

% Close the cursor and the connection.
```

APPENDIX III: GUI MATLAB Coding

LocationSys.m

```
function varargout = LocationSys(varargin)
% LOCATIONSYS M-file for LocationSys.fig
%LOCATIONSYS, by itself, creates a new LOCATIONSYS or raises the
existing singleton*.
%H = LOCATIONSYS returns the handle to a new LOCATIONSYS or the
%handle to the existing singleton*.
%
%LOCATIONSYS('CALLBACK',hObject,eventData,handles,...)   calls   the
%local function named CALLBACK in LOCATIONSYS.M with the given input
%arguments.LOCATIONSYS('Property','Value',...)   creates   a   new
%LOCATIONSYS or raises the existing singleton*.
%Starting from the left, property value pairs are
%applied to the GUI before LocationSys_OpeningFunction gets called.
%An unrecognized property name or invalid value makes property
%application stop.
%All inputs are passed to LocationSys_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help LocationSys
% Last Modified by GUIDE v2.5 29-Apr-2007 02:44:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @LocationSys_OpeningFcn, ...
                   'gui_OutputFcn',  @LocationSys_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before LocationSys is made visible.
function    LocationSys_OpeningFcn(hObject,    eventdata,    handles,
varargin)
```

```
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to LocationSys (see VARARGIN)

% Read the image file banner.jpg
handles.B23_edit = imread('layout23.bmp');
% Determine the size of the image file
info = imfinfo('layout23.bmp');
position = get(hObject, 'Position');
axes(handles.axes1);
image(handles.B23_edit)
set(handles.axes1, ...
    'Visible', 'off', ...
    'Units', 'pixels', ...
    'Position', [50 50 info.Width info.Height]);

% Choose default command line output for LocationSys
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes LocationSys wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function   varargout   =   LocationSys_OutputFcn(hObject,   eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes during object creation, after setting all properties.
function datacomm_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to datacomm_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor
'));
```

```
end

function datacomm_input_Callback(hObject, eventdata, handles)
% hObject    handle to datacomm_input (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of datacomm_input as
text
%str2double(get(hObject,'String'))      returns      contents      of
datacomm_input as a double


% --- Executes during object creation, after setting all properties.
function belkin_input_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor
'));
end

function belkin_input_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function techroom_input_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor
'));
end

function techroom_input_Callback(hObject, eventdata, handles)


% --- Executes on button press in locate_pushbutton.
function locate_pushbutton_Callback(hObject, eventdata, handles)

a = str2double(get(handles.datacomm_input,'String'));
b = str2double(get(handles.belkin_input,'String'));
c = str2double(get(handles.techroom_input,'String'));

I=NearestN(a,b,c)

set(hObject, 'Units', 'pixels');
% Read the image file banner.jpg
handles.B23_edit = imread('layout23.bmp');
% Determine the size of the image file
info = imfinfo('layout23.bmp');
```

```
axes(handles.axes1);
image(handles.B23_edit)
set(handles.axes1, ...
     'Visible', 'off', ...
     'Units', 'pixels', ...
     'Position', [50 50 info.Width info.Height]);

plotcoor(I)


% ------------------------------------------------------------------
--
function file_menu_Callback(hObject, eventdata, handles)


% ------------------------------------------------------------------
--
function     menu_file_savelocation_Callback(hObject,     eventdata,
handles)

f = figure;
temp = copyobj(handles.axes1, f);
%set(temp,'Units',368,'Position',748)
set(temp,'Units','default','Position','default')
set(f, 'Visible', 'off')

saveas(f,'locate.jpg','jpg')
delete(f)


% ------------------------------------------------------------------
--
function menu_file_close_Callback(hObject, eventdata, handles)

close(handles.figure1)
```

APPENDIX IV: Map Plotting MATLAB Coding

plotcoor.m

```
function[p]=plotcoor(1)

clear p

timeout=logintimeout(5)

% Connect to a database.

conn=database('FYP_AP','','')

% Check the database status.

ping(conn)


% Open cursor and execute SQL statement.

curs=exec(conn,'select * from plot_coordinate');

% Fetch tall the data in database APs.

curs=fetch(curs)

% Display the data.
columnnames(curs)

[m,n]=size(curs.Data)    %find the row and column sizes in the
                         %database

x=curs.Data(1,2)

y=curs.Data(1,3)

p=text(([x{:}]),([y{:}]),'x','Color','r','FontName',...
'ComicSans Ms','FontSize',20,'FontWeight', 'bold')

close(curs)
close(conn)
```