A DYNAMIC AND RELIABLE LOCATION TRACKING APPROACH FOR MOBILE ENVIRONMENTS

by

Pavan K. Nallamothu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

ARIZONA STATE UNIVERSITY

August 2004

A DYNAMIC AND RELIABLE LOCATION TRACKING APPROACH FOR MOBILE ENVIRONMENTS

by

Pavan K. Nallamothu

has been approved

August 2004

APPROVED:

_____, Chair

_____

_____

Supervisory Committee

ACCEPTED:

_____

Department Chair

_____

Dean, Division of Graduate Studies

ABSTRACT

Localization is a key feature for any context aware application. Most of the existing localization approaches use a fixed infrastructure and don't support ad-hoc architecture. The need for localization for a mobile environment which is ad-hoc, reliable, and cost efficient simultaneously has led to the idea of mobile localization approach. This approach consists first of an approximate localization, and second of a precise localization, to ensure accurate location information even in environments in which nodes cannot triangulate because of a lack of infrastructure. With the ad-hoc nature of mobile localization, nodes dynamically identify neighbors and triangulate their own node position using their neighbors. The proposed protocol is robust against signal collision and multi-path effects. It is also energy efficient to prolong battery lifetime. Mobile localization is implemented in a static environment using TinyOS, and the obtained data is used to simulate a mobile environment. The performance of the protocol is evaluated with respect to node mobility, radio range, the number of nodes in the environment, and the time interval for refreshing location information.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Professor Dr Sandeep KS Gupta, for his guidance and support during my M.S study and research. It has been very rewarding to have him as my advisor. I would also like to thank my committee members, Dr. Karam S. Chatha and Dr. Chaitali Chakrabarti, for their support.

I would like to thank my research colleague in IMPACT Laboratory: Valliappan Annamalai for helping me with implementation, and John Quintero for helping me in statistical analysis and documentation.

Lastly, I am forever indebted to my parents for their understanding, endless patience and encouragement when it was most required.

Pavan K Nallamothu.

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

In today's world, our environment is being made context aware to serve mankind more efficiently [1,2]. The need for a user to be aware of his own location is very important. Researchers are trying to find newer and more efficient [21] types of localization techniques [3, 6, 11, 12, 13, 14, 15] every year. This is an ongoing process, as localization method for a particular scenario [4, 6, 20] may not work for other scenarios. Imagine a forest fire in an inhospitable terrain where a group of monitoring nodes have been dropped at random. These nodes should be able to localize themselves from their neighbors positions as they move. For these kinds of scenarios, existing localization approaches like Active badge [5], RADAR [12], and Cricket [11, 13] cannot be used, as they depend on a fixed infrastructure. GPS [19, 8] cannot be used, because of the difficulty in monitoring a large number of nodes through satellite. Therefore, the need to accommodate node mobility has led to a localization approach in which mobile nodes are able to localize themselves with respect to their neighbors. This localization technique is dynamic, reliable, robust and doesn't need a fixed infrastructure.

The basic assumptions of this architecture are as follows: All nodes present in the environment have computation and communicational capabilities. They each have unique ID's, know their start point and move at a standard pace with direction information. In the proposed mobile localization approach, there are two kinds of localization: approximate localization and actual localization. During approximate localization, nodes have an approximate idea of their coordinates. In actual localization, nodes update their approximate location information from neighboring dependent nodes. The subject node uses time-of-flight of an acoustic signal for distance estimation, using an RF signal for communication.

Localization accounting for mobility adds even greater complexity to existing localization approaches. The mobile nodes should continuously be able to determine their position. Assuming no fixed infrastructure nodes are forced to localize themselves from neighboring nodes that are mobile, too. Nodes should be sequenced [16] properly to obtain localization, as two nodes cannot localize at the same time because of the problem of sharing the same neighbors or each other. The problem of signal collision, corruption [9], and the difficulty in sequencing for localization increases with the number of nodes. Also, as the protocol relies on neighbors, the converge-cast [28] problem from dependent to subject occurs. The other challenging aspect is that the protocol should be made energy efficient to help ensure a long lifetime of the nodes.

This mobile localization approach also applies to real world situations, like office environments, shopping malls etc, where people will be moving and some static objects are available [4, 5, 6, 7, 8]. The existence of static objects capable of

communicating with other nodes improves the performance of the protocol further. They help quantify location more reliably compared to mobile nodes. Chapter three presents the algorithm for the proposed localization approach, a description of the kinds of problems arising from it and ways to tackle them. The implementation and simulation results and aspects of real world problems are explained in chapters four and five. The reliability and performance of the protocol with respect to parameters such as the number of mobile nodes, the location refresh interval, the radio range and power level of the nodes and the mobility of the environment are carefully evaluated in chapter six. In chapter seven, comparison with a similar localization approach is done. The thesis concludes with a presentation of the advantages of the proposed localization approach and future research.

CHAPTER 2

RELATED RESEARCH

Localization is an issue researchers have been working on for a long time. This chapter briefly outlines existing localization approaches. These can be broadly classified as either tagged or un-tagged location tracking methods. In tagged methods, the user carries a smart device which helps in location identification. Factors like environmental noise and users line-of-sight determine the reliability of the protocols. Un-tagged methods, although providing precise location information, are costly to implement. Different kinds of tagged and un-tagged methods and their corresponding advantages are described in sections 2.1 to 2.5.

2.1. Electro Motive Force Method

Electromagnetic [8] sensing is a classic position-tracking method for fine location tracking to an accuracy of 1mm and 0.1 degree orientation. A Motion Star DC magnetic tracker generates axial DC magnetic-field pulses from a transmitting antenna in a fixed location. The system computes the position and orientation of the receiving antennas by measuring the response in three orthogonal axes to the trans-

mitted field pulse, combined with the constant effect of the earth's magnetic field. This method has the disadvantages of high implementation cost, complex setup and poor performance in the presence of metallic objects.

2.2. GPS System

GPS uses satellites to track user location. This method is not feasible for indoor environment, as signals from satellites degrade in indoor environments. It is also costly and requires a very fine granularity in the system clock. Inspite of all these disadvantages, there were some implementation of GPS [6] in indoor environments which provide an accuracy of about 5 meters.

2.3. Laser Range Finder

This method is developed by the MIT media laboratory [31]. The users have smart receivers placed in their shoes. The shoe position is found by a laser range finder working on the principle of SONAR. This method provides a good precision but is costly to implement and needs a large infrastructure deployment.

2.4. Active Badge

Active Badge is a smart device capable of transmitting and receiving beacons used for location tracking. Users in the environment are assumed to wear active badges. There are 3 kinds of technologies used in active badge tracking.

2.4.1. Infrared Tracking

Active Badge, developed by AT&T laboratories [5], is capable of radiating infrared signals every 10 s. These infrared pulses are captured by infrared sensors, and the data is sent to a centralized server to calculate position. The system has poor performance in the presence of fluorescent lighting and sunlight. It is also sensitive to reflected infrared signals.

2.4.2. Lateration by Signals Strength

The RADAR [12] project, developed by Microsoft Research Group, tracks 2D user position within a building based on IEEE 802.11 wireless networking technology. The signal strength and signal-to-noise ratio of user's wireless devices are measured at the base station. This data is used to compute the location of the user. It provides an accuracy of 4 to 5 m, but requires multiple base stations. Also, the signal strength measurement is not consistent because of signal loss in the indoor environment.

2.4.3. Lateration by Time of Flight

AT&T researchers came up with an Active Bat location system [30], which uses the time-of-fight of ultrasonic pulse for lateration. The nodes to be localized send ultrasonic pulses. These pulses are received by wall-mounted receivers, and the distance is computed at a centralized server system.

MIT laboratories proposed a decentralized location system called Cricket [11, 13]. In this system, the wall-mounted transmitters transmit active beacon message

around a building. The Mobile receivers calibrate their distance from the transmitters time of flight difference between RF and ultrasonic pulses. A series of these distance values from different beacon sources are used to estimate position. The advantage of this method is user privacy the disadvantage is a lack of centralized management.

2.5. AD-HOC Localization

In this kind of localization, nodes dynamically discover neighboring nodes and triangulate their location from them. Multi-lateration [22, 23, 24] is performed to acquire fine-grained location information. This method requires the availability of fixed nodes for localizing ad-hoc nodes. The ad-hoc nodes loose location information if they cannot find fixed nodes in range with which to triangulate.

The other kind of tracking method, classified as the untagged method, has accurate location determination, no attenuation losses in indoor environments, no complex calculation, and the user need not be tagged.

2.6. Video Recognition Method

In this method [32], images are captured at different instances of time. The difference in the successive images with respect to a particular object to be tracked is used to estimate location. Although this method is robust, it requires a complicated setup. It is expensive to implement, and the object to be tracked should always be in the systems line of sight.

2.7. Active Floor Method

In this approach smart pressure sensors [33] capture footfalls, and the system uses this data for position tracking. Location tracking using this method is accurate, but user identification is a complicated problem in this approach.

All the above-mentioned research is reliable for static environments.Though the user keeps moving, he get localized by a fixed infrastructure.

Mobile localization is a step ahead of these existing localization techniques. It is infrastructure free, cost efficient, dynamic, and robust to signal collision, corruption and multi-path effects.

CHAPTER 3

MOBILE LOCALIZATION ALGORITHM

Mobile localization algorithm is designed for localizing mobile users in an infrastructure-free dynamic mobile environment. The algorithm can typically be used in scenarios such as localizing mobile monitoring nodes in hazardous regions like forest fires, nuclear dumps, and scenarios like underwater aquatic monitoring. The algorithm can also be extended to localizing mobile users in a typical office environment, shopping malls etc. The assumptions for mobile localization algorithm are that nodes have computational and communicational capabilities. They have a unique id, are time synchronized and time multiplexed. Nodes know their initial location as they enter the environment, and their mobility is constant. With all these conditions seemingly attained, every node updates it location during its multiplexed slot. When a node gets its turn to update its location information, it becomes the Subject, and the nodes it chooses to triangulate its position with are Dependant1 and Dependant2, based on the order of participating in location estimation. Subject and Dependant1, Dependant2 each have a unique protocol. The algorithms they follow, state diagram of the algorithm (figure 1), and message beacons for communicating

are described below.

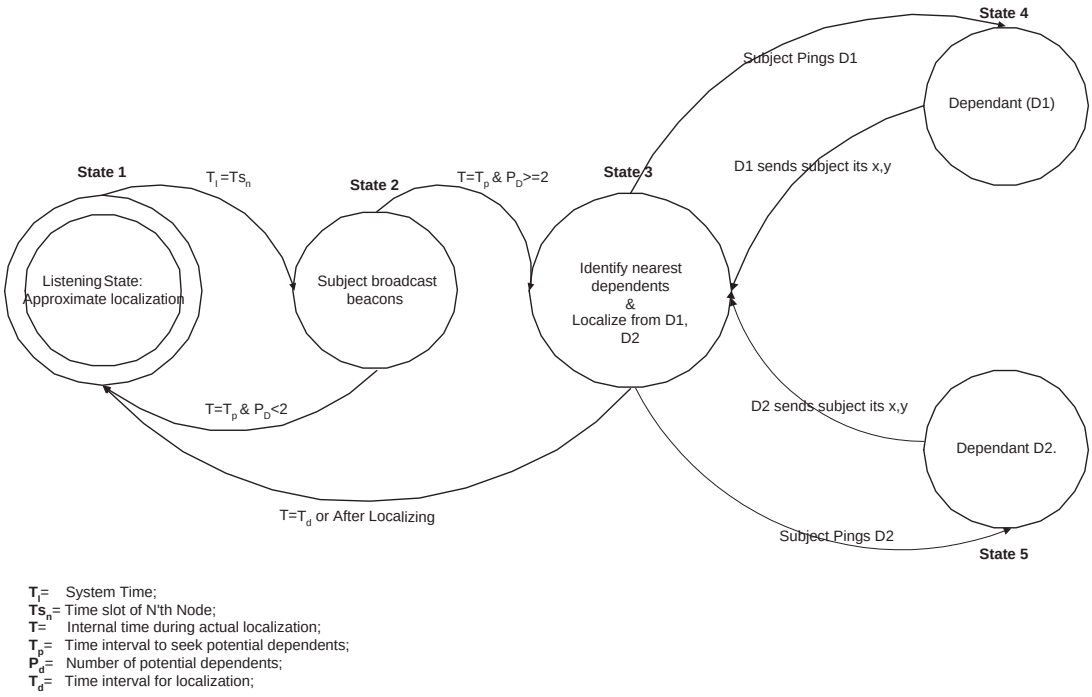

Figure 1. Mobile localization state diagram

3.1. Mobile Localization State Diagram

Mobile localization algorithm can be explained by the five states of the mobile localization state diagram. The state value and condition for state transition are as described.

State 1: Nodes are initially in the listening mode during which they continuously update their approximate location information as they move. They use their previous location coordinates, their mobility (speed), time difference, and direction

movement node to calculate their approximate location. When the node gets its turn to update its location, it becomes Subject.

Approximate Location estimate =

$$previous(x, y) + ((\mu_s * \triangle T)cos(D_s), (\mu_s * \triangle T)sin(D_s)); \tag{3.1}$$

Where,

μ_s= Mobility of the node

$\triangle T$= Time difference between previous location calibration and present time

D_s= Direction of motion of mobile node

State 2: The Subject, on getting its chance to update its location, broadcasts for the nodes (mobile or static) around it to participate in its location calculation (figure 2). The broadcast message is:

Subject id	1	Time out value

where 1 signifies that the subject is broadcasting for potential dependants;

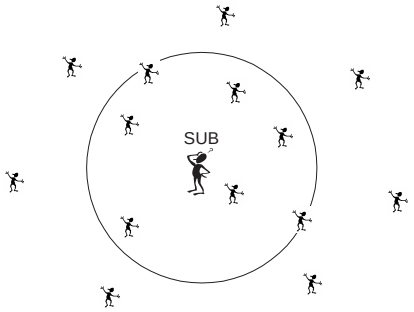

Figure 2. Subject beaconing to dependents

After the neighboring nodes receive these beacons, they respond (following a collision avoidance principle) with a beacon message to the subject:

Subject id	Potential dependants id	distance	Time out value

State 3: The Subject stacks all potential dependants and selects two reliable dependants based on the criteria of either being a nearest neighbor or being a static node. The subject then sends a uni-cast beacon to each of its reliable dependents (Dependant1 and Dependant2), one after the other, for their own location coordinates:

Best Dependants Id(D1)	Subject Id	Time out value

State 4 and 5: Dependants (D1/D2) when receiving this beacon message, send their corresponding (x, y) coordinates:

Subid	D1Id	Xco	Yco	Time out

The Subject calculates its position from the D1 and D2 distance and location coordinates and its own previous location information. The subject then returns to the listening mode. In case the Subject did not hear from any or both of the Dependents, the Subject relies on its approximate location information.

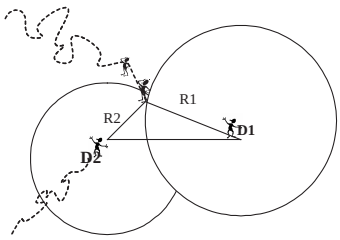

Figure 3. Location calculation of subject

Subject location calculation:

$$(X_2 - X_1)^2 + (Y_2 - Y_1)^2 = R^2; \tag{3.2}$$

From the data provided by the Dependents, the Subject can potentially be in two locations, A or B as shown in figure 3. The exact location is identified from previous location information stored in memory.

The algorithm is made efficient for further location updates by the use of previously-existing Dependants. The Subject in this case need not broadcast for dependants, it only has to check if it could use one or both of the previous dependants. This scenarios is outlined in figure 4.

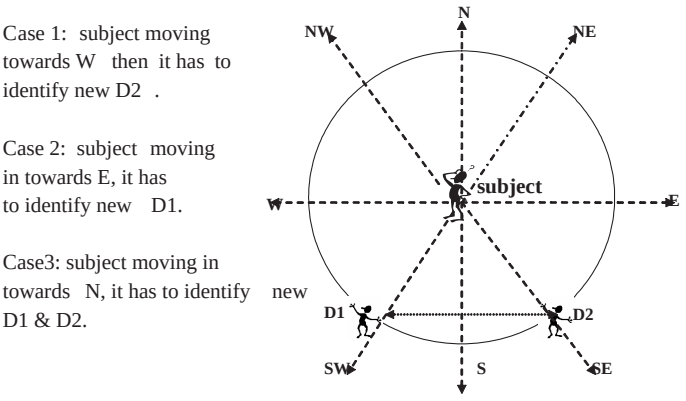

Figure 4. Possible mobility of the subject

3.2. Mobile Localization Problems

3.2.1. Signal Collision

Since there are multiple mobile nodes which have to be localized, there arises the problem of signal collision when multiple nodes transmit beacons for location information at the same time. This problem can be solved through a time multiplexing

method, in which each and every mobile node is given a time slot to update their location information. The time slot for each mobile node is calculated as:

$$T_s = T_{Rsn} + T_{Rdn} + T_{D1n} + T_{D2n} + M_G; \tag{3.3}$$

Where,

T_s= Time slot in which individual mobile node calibrate location.

T_{Rsn}= Time taken for subject to broadcast beacon to all nodes within its range asking them to participate in location tracking.

T_{Rdn}= Time taken for all neighboring nodes (which are potential dependants) to respond.

T_{D1n}= Time taken for subject to receive distance, location information of D1.

T_{D2n}= Time taken for subject to receive distance, location information of D2.

M_G= Margin for error to avoid signal overlapping.

The time after which each node gets a chance to update its location information depends on the number of mobile nodes. This is calculated as:

$$T_I = T_S * N_M; \tag{3.4}$$

where,

T_I= Time interval after which a mobile node gets its chance to update its location information.

N_M= Number of mobile nodes present in the environment.

The time interval for location refreshing is directly proportional to the number of mobile nodes; it should not be too long, because a mobile node may lose its

location information as it is constantly moving. This can be solved by partitioning the environment in to different regions, each region operating at a different frequency.

$$T_{I_m} = T_I / N_R; \tag{3.5}$$

where,

T_{I_m}= Time interval improved because of partitioning the environment into different regions

N_R= Number of regions in the environment

3.2.2. Vast Beacon Messages Spread in the Environment

As the number of mobile nodes increases, the number of message beacons increases, which increases the possibility of jeopardizing beacon signals. This problem can be addressed by giving a timeout value for each beacon, after which beacons die down.

3.2.3. Wrong Location Estimate

Each node calibrates its location at an instance of time and then waits for its turn to update its location information. In the mean time, there might arise a situation in which all its neighbors have been mobile and don't know their own new locations. This leads to a wrong location estimate. This can be rectified by updating node location information periodically by calculating distance using speed, time and direction information.

Approximate location estimate =

$$previous(x, y) + ((\mu_s * \triangle T)cos(D_s), (\mu_s * \triangle T)sin(D_s)); \tag{3.6}$$

3.2.4. Multi-path and Fading Effects of Signals

As the beacon signals travels in air, they undergo reflection, refraction, and diffraction because of different obstacles present in the signal path. The signals therefore undergo multi-path and fading effects. This issue is overcome by ignoring duplicates of the same signal, as beacons have unique ids. Also, TOF differences between Ultrasonic and RF signals reduce the error caused because of multi-path and fading effects [10].

CHAPTER 4

IMPLEMENTATION

For the implementation of a mobile localization protocol, each and every node in the environment should have communication and calculation capability. Typical node hardware consists of an ultrasonic transceiver, an RF transceiver, a magnetic compass, an internal clock of fine granularity, a microprocessor, and memory to store temporary data.

Implementation of the protocol is done in Tinyos [29]. Tinyos is component based runtime environment developed by the University of California, Berkeley. The TinyOS system, libraries and applications are written in nesC [26], a language for programming structured component based applications. It has C-like syntax. Nesc supports the tinyos concurrency model, structuring and naming. For any application in tinyos, a component and an implementation module have to be defined. Implementation modules have tasks, hardware event interrupts, standard commands and events to drive hardware nodes called motes.

4.1. Motes Hardware Overview

Standard motes, usually called mica motes (mica2), are second-generation mote modules used for research and development of low power wireless sensor networks. Motes are equipped with ATmega 128L, a low power microcontroller which runs tinyos from its internal flash memory. It uses a CC1000 ISM band radio transceiver module for wireless communication. It has 128kb of onboard flash memory, operates at 4MHz speed, and has UART serial communication with a 10 bit ADC. The CC1000 radio operates at 916 MHz radio frequency and is capable of 100 foot radio range. The mote has a 51 pin external connector and draws 0.75 mW when fully operational. It provides a node interface through 3 onboard LEDs and can be driven through a 3V external power supply or by 2 AA batteries. A typical mote block diagram is shown in figure 5.

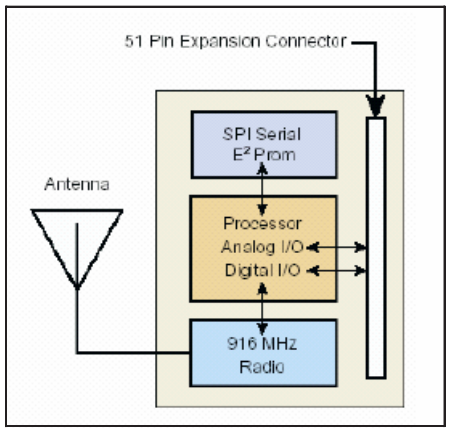

Figure 5. Typical Mica Mote block diagram

Motes have detachable sensor boards mounted through the 51 pin connector.

The sensor board supports analog inputs, I2C, SPI, UART, and a multiplexed address/data bus. Sensor boards have a photo diode, thermistor, microphone, sounder, and a magnetic and acceleration sensor. The block diagram of a mica mote with sensor board and direction signaling is shown in figure 6.

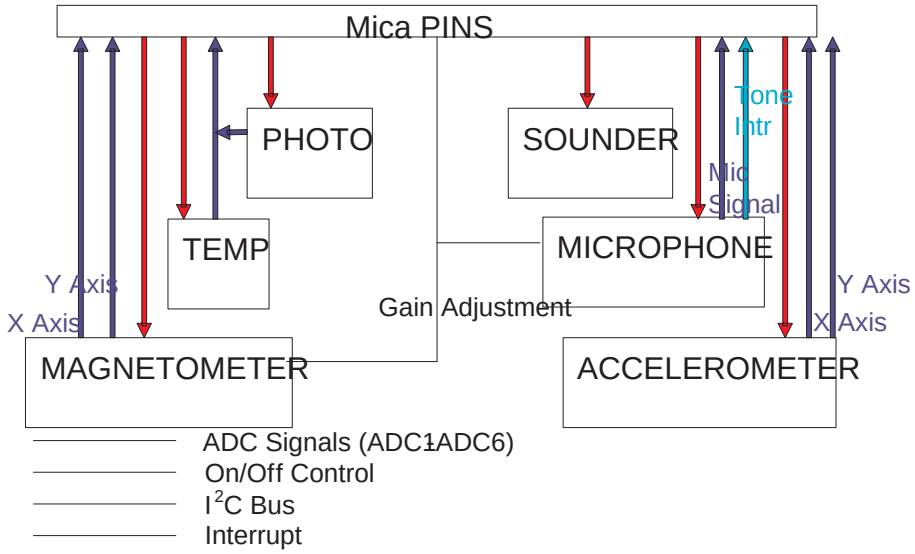

Figure 6. Typical Sensor board block diagram

These mica motes are programmed via a mica interface board, which has a parallel port for programming motes and a serial port to read data from motes. Interface boards are powered by 3 V external power supply or by the 2 AA batteries of the mote, when plugged on top of the interface board. A typical mica sensor board and programming board are shown in figure 7.

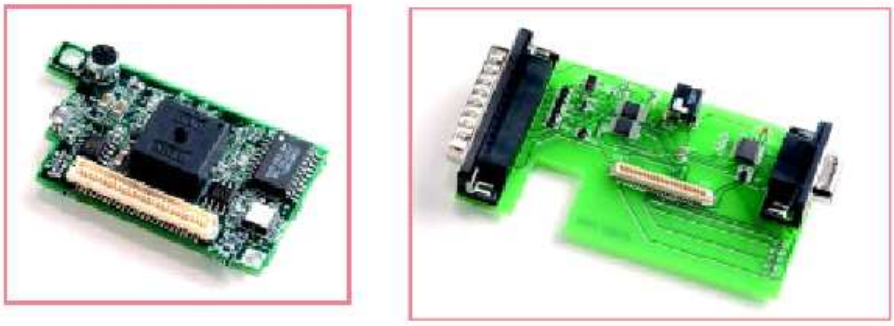

Figure 7. Sensor board micasb and interface board.

4.2. Tinyos Programming Overview

An application in nesC consists of one or more components, which are linked together to provide and use interfaces. An interface consists of a set of commands that the provider must implement and a set of events that the node must provide. A single component may provide or use multiple interfaces. For any application in nesC, there are two types of components: modules and configurations. Modules provide application code, implementation and interfaces, while configuration are used to assemble other components, and to connect interfaces to other interfaces.

Since tinyos provides a concurrency model, there are tasks and hardware event handlers running in parallel (figure 8). Tasks are like functions. They should be extremely light weight. The hardware event handlers are executed in response to hardware interrupts, and are run to completion. Commands and events that are part of hardware event handlers are declared with 'Async'. There are atomic statements to perform small operations and 'norace' keyword used to avoid race condition.

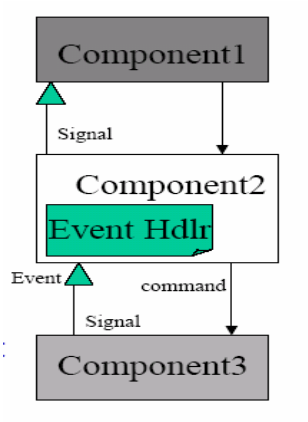

Figure 8. Tinyos signaling model.

4.3. Implementation Idea

The proposed protocol can be implemented with a group of 10 motes equipped with sensor boards. The Mobile localization algorithm is programmed into every node, which are placed at random on the floor. The acoustic ranging method described in next section is used for distance estimation. The localization values of each node are updated to the PC through the UART.

4.4. Acoustic Ranging

This is an acoustic distance approximation method [27] provided by tinyos and developed by the Institute for Software Integration Systems, Vanderbilt University. It uses standard mica motes and sensor boards to obtain promising results with a 1 percent distance error. In acoustic ranging, a mote emits symultanious RF signals and a series of acoustic chirps. All other motes on receiving the RF pulse, sample

chirps one by one and process them as a single sampled signal. The typical plot at the sender and receiver end is shown in figure 9.

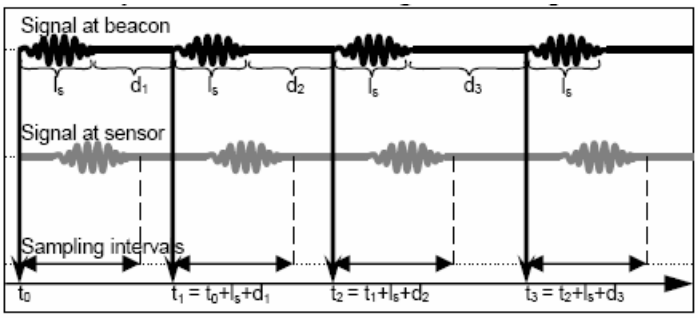

Figure 9. Multiple signal Sampling. Ls- Length of signals, (d1, d2,, dN) - delays between the consecutive chirps (known to the sensor).

With both the length of signals and the delay between chirps known, the start times of the sampling intervals can be easily computed. A digital band pass filter is employed to improve SNR. Since environmental disturbances are Gaussian in nature and are independent of the chirp, the useful signal content would be identical. By adding signal samples together, SNR improves by 10log(N) dB. Therefore for 16 chirps, SNR improves by 12 dB. Ranging is calibrated from the power of local maximas of filtered samples at receiver. The error plot over the range of 10 m and the histogram acoustic of the range error are shown in figure 10, figure 11.

The only problem with using this system is that it is only reliable outdoors. Multi-path effects disrupt the signal indoors

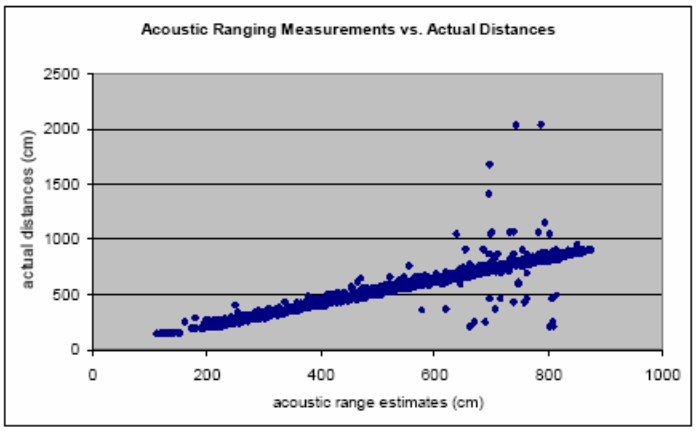

Figure 10. Range Error Plot.

4.5. Implementation Design of Mobile Localization

Implementation of mobile localization specifically involves 2 modules - the subject module and the dependent module. Both subject and dependent modules are present in the motes. They are activated upon specific requirements. Care should be taken that no two motes become subject at the same time. Subject and dependent state machines programmed in motes share an idle state as the common starting point. The state machines are Moore state machines, i.e., the output of a state depends only on the present state. The start of the state machine is the idle state, when it checks if other nodes, are a subject. Then it tries to be a potential dependant and assists the Subject in its localization. Otherwise it would itself become the Subject.

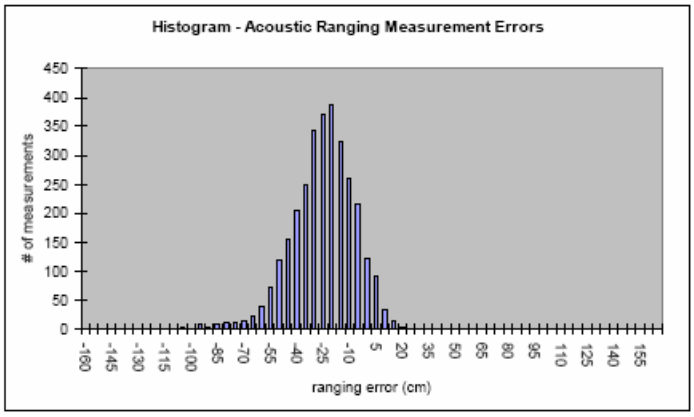

Figure 11. Histogram of Acoustic ranging error.

4.5.1. Subject State Diagram

The subject, depending on the tasks it should perform, is divided into 5 states, and makes transitions from one state to the other upon corresponding events. The five states, their state values and their transitions are shown in figure 12. The onboard leds are configured to display the state of subject and dependent motes. The subject state diagram is described below:

- STATE IDLE: This is a transition state to become either a Subject or a Dependent. In this state, all leds are turned off. The node remains in the idle state till Buzy Backoff turns to zero, during which it checks to see if it could become a potential dependent. If not, it becomes a subject. When Buzy Backoff becomes zero, the state becomes STATE ACUATING.

- STATE ACTUATING: In this state, the mote transmits simultaneous RF and acoustic pulses as described in the acoustic ranging method (section 4.4). In this

state red led is turned on. The node remains in this state till it has transmitted all 16 acoustic pulses, after which it moves to STATE BCASTRCV.

- STATE BCASTRCV: After the subject node broadcasts acoustic pulses seeking potential dependents, it waits to hear from them, changing its state to either STATE SENDD1 or to STATE IDLE. On timeout, it resets back to STATE IDLE when it does not hear from any the potential dependents, In this case there are no potential dependents in range of the subject. It would move to STATE SENDD1 when it hears from at least two potential dependents. The format of packets received is [receive(TID, RID, Distance)] with message id of 114.

- STATE SENDD1: After the subject hears from its potential dependents, it stacks ups the dependents, selects the best dependents (the nodes closest to it or a static node), and sends a message to two dependents (D1, D2). The message format is: [send(TID, RID)] with a message id of 115. The subject moves to STATE RCVD1D2 on send done.

- STATE RCVD1D2: In this state, the subject waits to hear from dependents D1 and D2. The x, y coordinates that both these dependents sent and the distance information from the previously stored stack is used to calculate the current coordinates of the subject. The message that the subject would hear from the two dependents is [receive(TID, RID, XCO, YCO)] with message id of 117. After hearing from two dependents, the subject turns the red led off and moves to STATE IDLE.

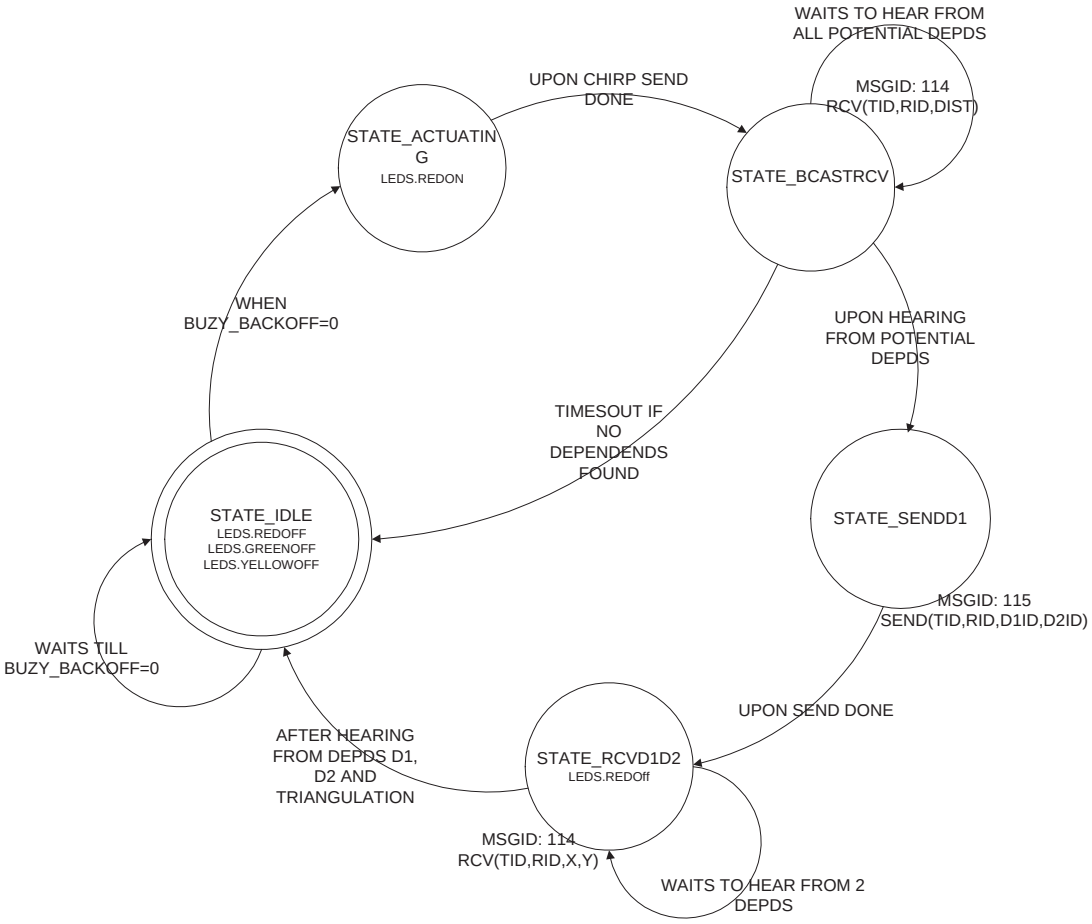

Figure 12. Subject State Diagram.

4.5.2. Dependent State Diagram

The dependent has 5 states. During transitions nodes become dependents and assist the subject in localization. The dependent state diagram is shown in figure 13. The description of each state and its transition are shown below:

- STATE IDLE: This is the start state - in this state the nodes wait for RF pulses

from the subject. Once the pulse is received, it changes to STATE SENSING. In the STATE IDLE all leds are turned off.

- STATE SENSING: In this state, the potential dependents receive a series of chirps sent by the subject, filters out noise, runs a peak detection algorithm, and calibrates its distance from the subject node to itself as described in acoustic ranging method. When the dependent is in this state, it would turn on the green led. After obtaining the distance information, the dependent changes its state to STATE INTIMATESUB.

- STATE INTIMATESUB: The calculated distance information is stored and is sent to the subject in the form of message: [send(TID, RID, DIST)] with a message id of 114. The important aspect that has to be taken care of in this state is that the message sent to the subject should not be lost because of collision. Potential dependents process chirps at approximately the same time and the send distance information to the subject. There is a great possibility that the message is lost because of collision. There fore, each and every potential dependent should follow a collision avoidance principle. This is achieved by setting a small time delay, unique to each dependent, before they contact subject. After the message send to the subject is done, the dependent moves to STATE RCVD1 state.

- STATE RCVD1: In this state the dependent waits for the message from the subject of the form [receive(TID, RID, D2ID, D3ID)] with a message id of 115. On receiving the message, the dependent turns the red led on and moves to state

SEND SUB. In case the potential dependent did not hear from the Subject, it resets back to STATE IDLE.

- STATE SENDSUB: In this state, dependents D1 and D2 send messages to the subject in the format: [send(TID, RID, Xco, Yco)] with an id of 114. Again, there is a potential danger of signal collision. This is overcome by careful queuing of dependents. At the send done event, the dependent would be reset to STATE IDLE.

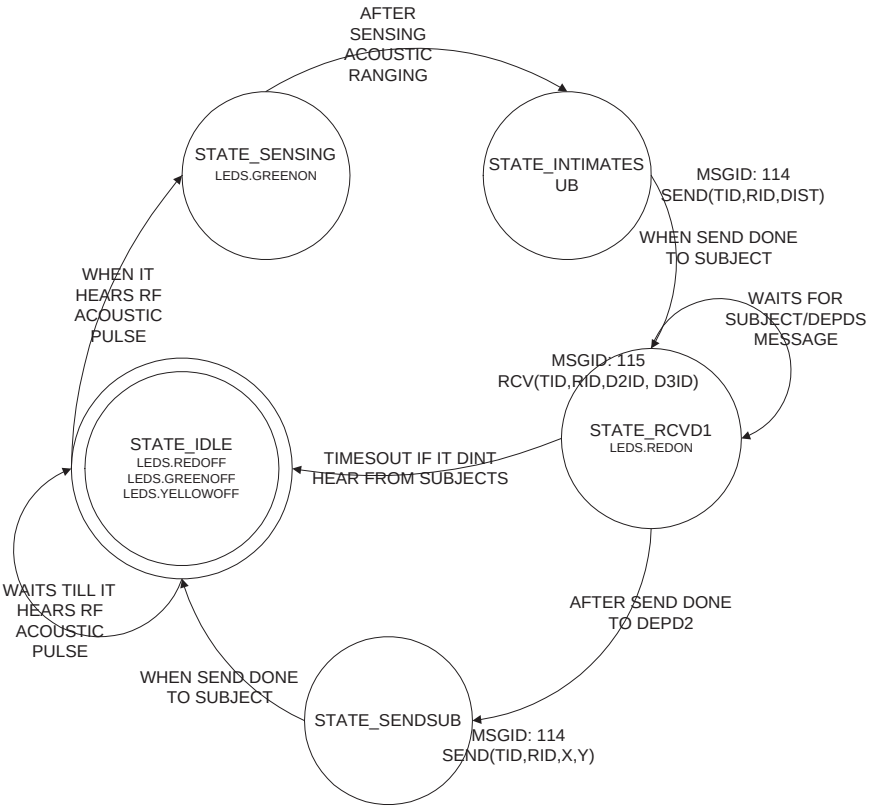

Figure 13. Dependents State Diagram.

Typically at an instance of mobile localization, when the subject is triangulating its position from dependents, the led's configuration is as shown in figure 14.

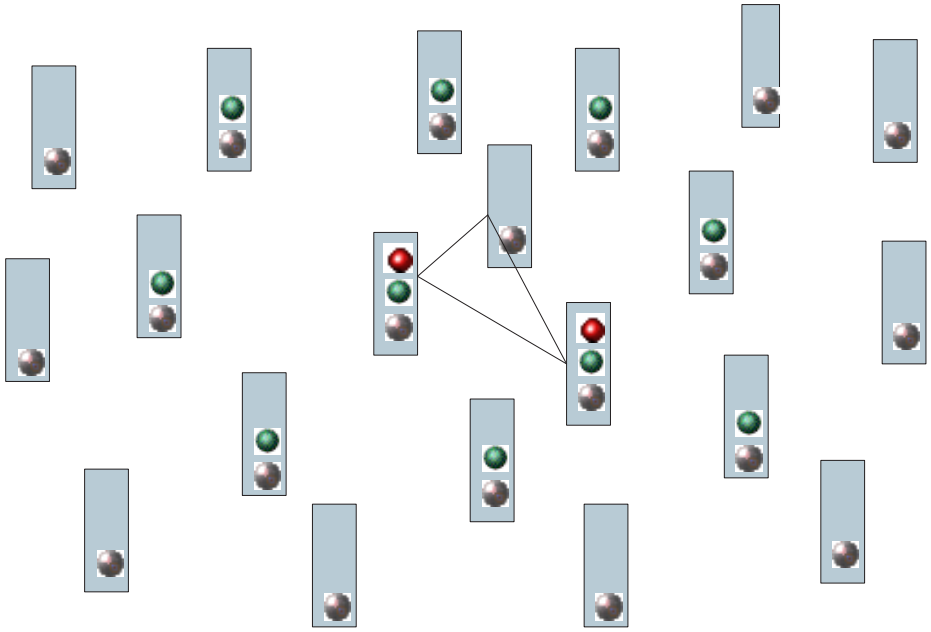

Figure 14. Leds configuration in an instance of mobile localization.

The complete state diagram running on each node is as shown in figure 15.

Since it is difficult to implement mobility and simultaneous localization, mobile localization is simulated using a Visual C++. Application GUI interface is provided to simultaneously change the number of nodes, the subject position, mobility, time refresh interval, and radio range and thereby to study the reliability of the protocol.

4.5.3. Acoustic Ranging Statistical Analysis for Indoor Environment.

In order to simulate close to an implementation of mobile localization, the obtained distance error and package loss error are statistically analyzed and incorporated into simulation. The reliability of motes and the acoustic ranging method is evaluated by conducting a series of experiments.The plot of distance error against number of errors occurred for acoustic ranging when separated by 100 cm, 200 cm, 300 cm and 400 cm is shown in figure 16, 71, 18, 19. The gaussian curve of the distance error helps in determining the confidence level of the acoustic ranging method. As observed from the figures, for a 90 percent confidence interval, the error varies from 5 to 25 cms for 100 cm; -5 to 15 cms for 200 cms; -1 to 20 cms for 300 cms; and -5 to 15 cms for 400 cms.

The reliability of motes is also evaluated for a test separation of 100 cms. The standard deviation of 3.873 cm is observed for 30 motes.

The statistical data described above is incorporated into the simulation environment to make it more closely approximate realistic values. The simulation environment and its features are as described in chapter 5.

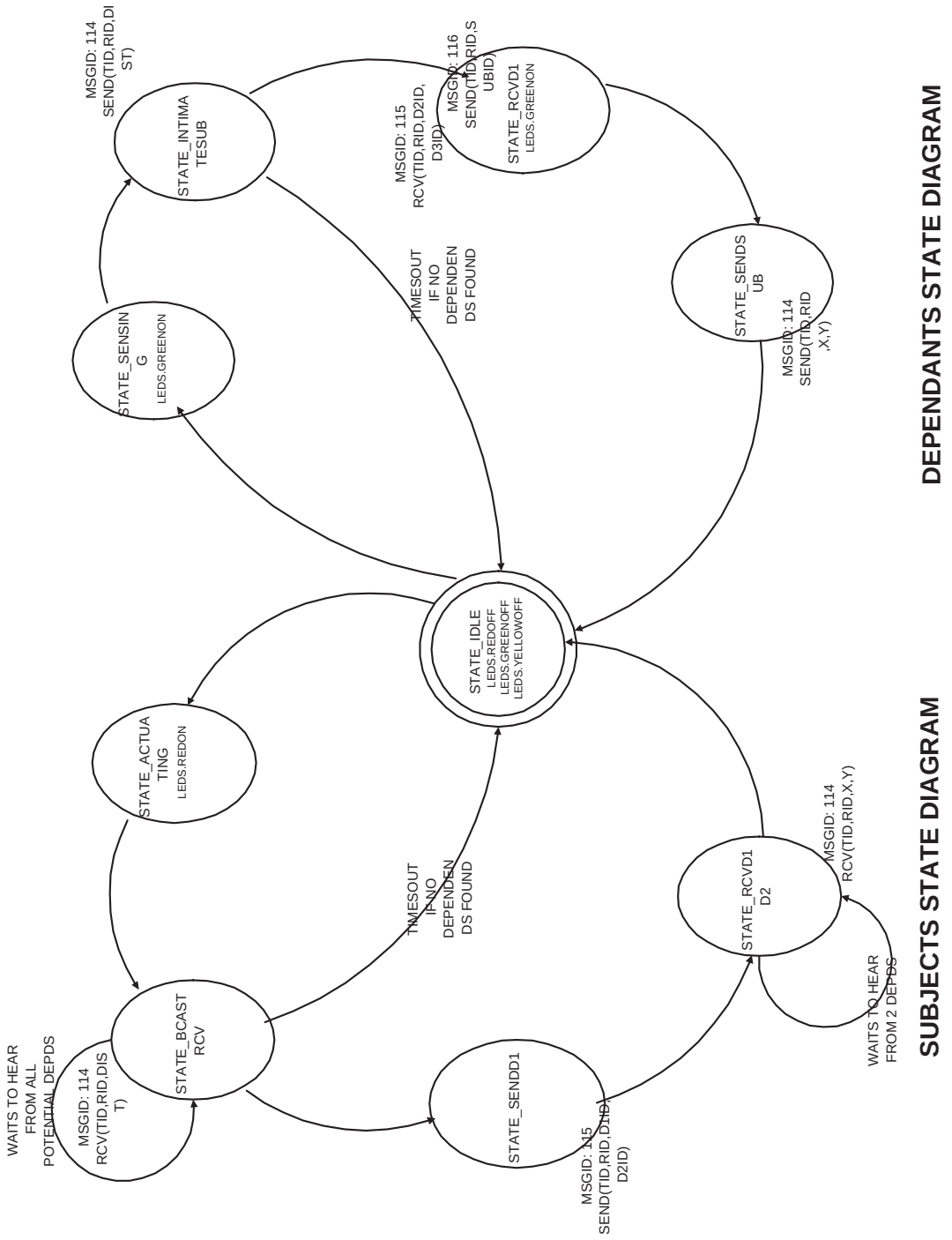

Figure 15. Complete State Diagram.

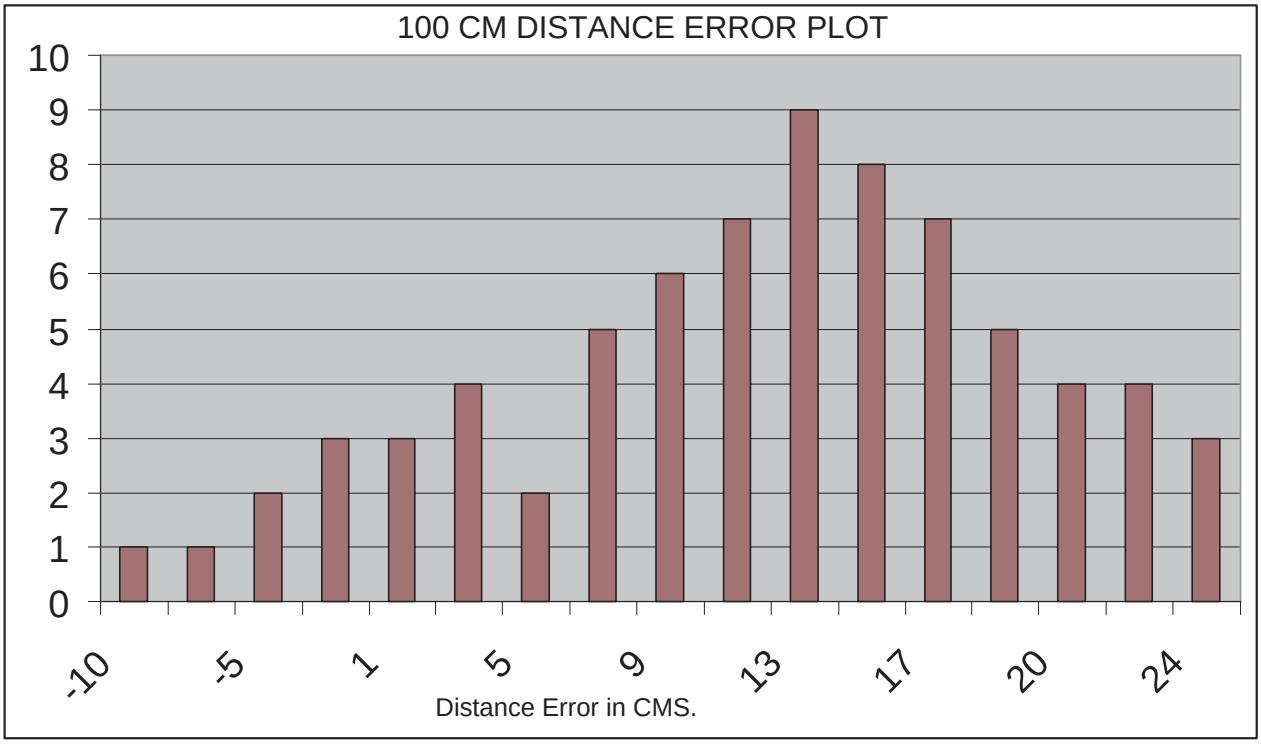

Figure 16. Acoustic ranging Gaussian error plot for 100 cms separation.

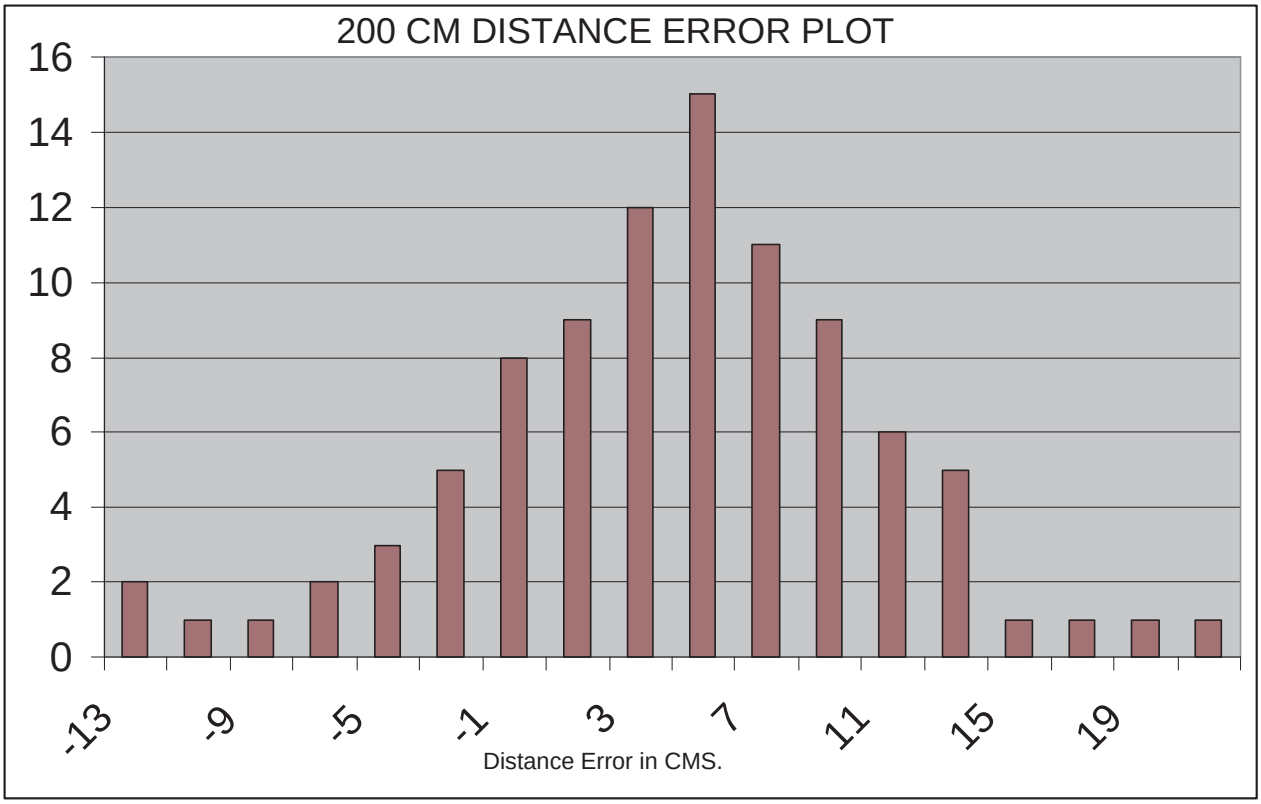

Figure 17. Acoustic ranging Gaussian error plot for 200 cms separation.

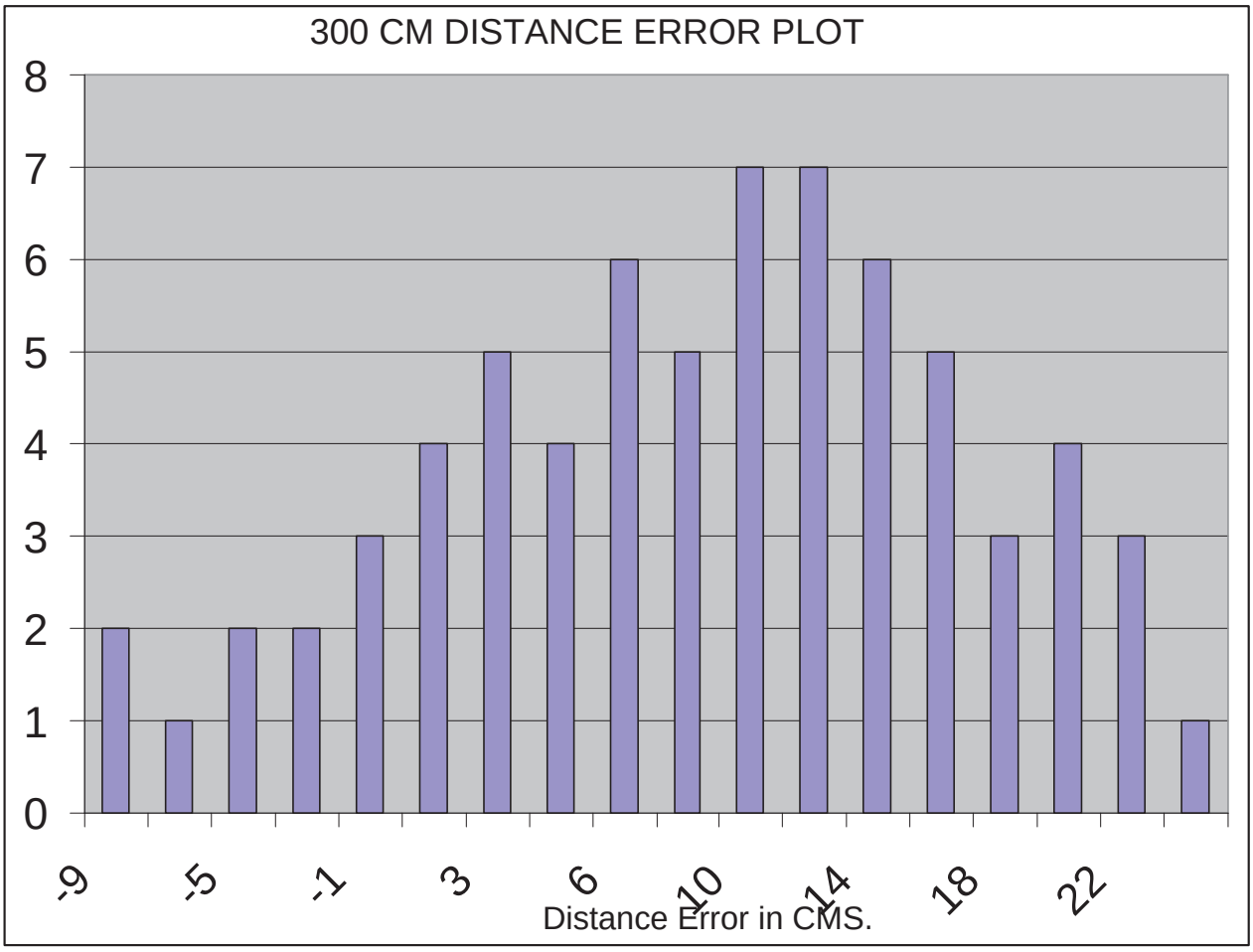

Figure 18. Acoustic ranging Gaussian error plot for 300 cms separation.

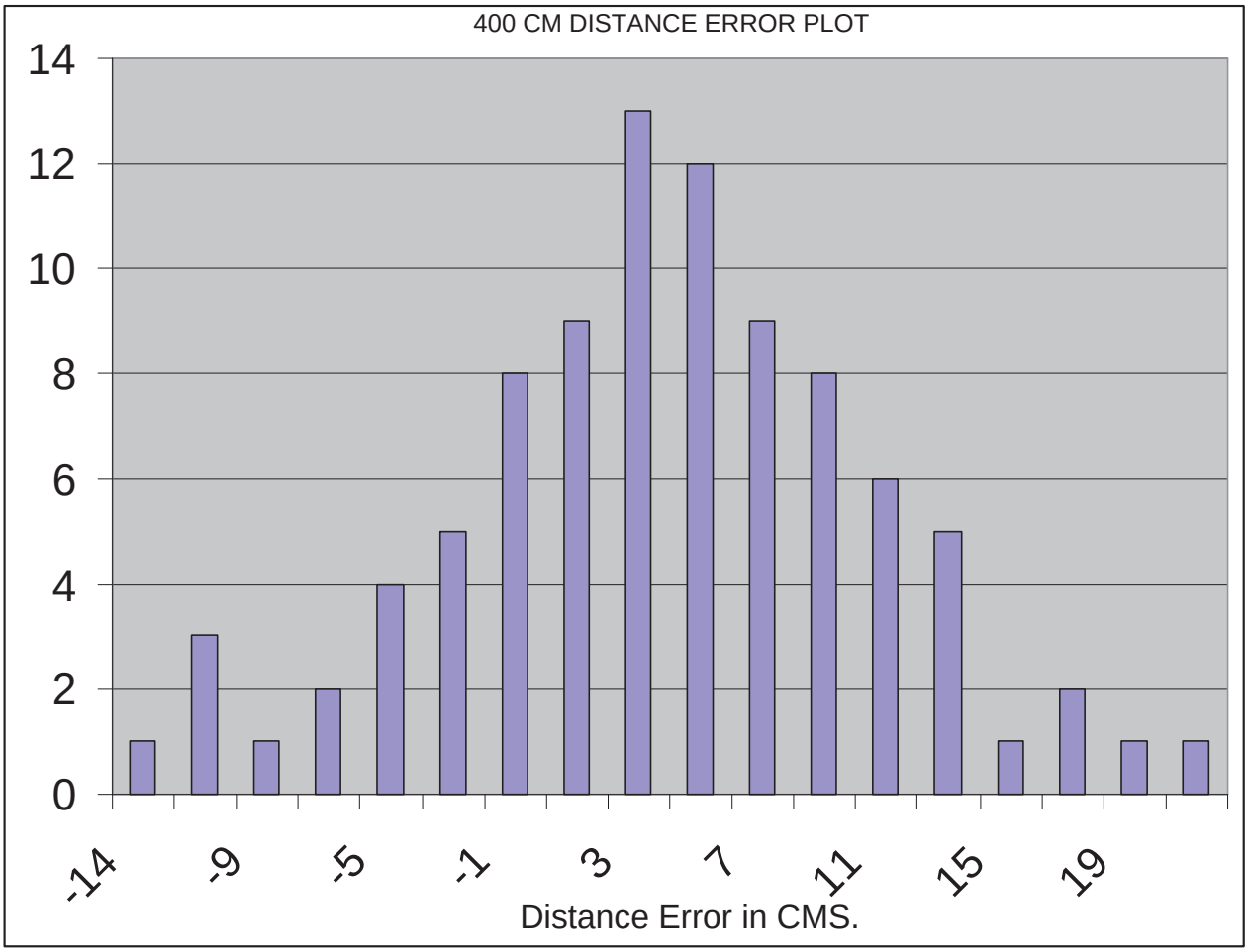

Figure 19. Acoustic ranging Gaussian error plot for 400 cms separation.

CHAPTER 5

SIMULATION

The simulation environment [17] for mobile localization consists of mobile and static nodes placed at random. As the mobile nodes move, their position is calibrated by the mobile localization protocol as proposed in chapter 3. The statistical data from the implementation is incorporated in the simulation to exactly emulate implementation.

The key features of the simulation environment are:

- Mobility of the environment, the radio range, and the time interval for location refreshing can be given as input parameters.
- It has the ability to randomly change direction of motion of the mobile nodes.
- The directional error due to the magnetic compass is incorporated by a random error parameter.
- Actual and calibrated node locations are simultaneously stacked into the data sheet.
- Above all, the environment is reusable.

In order to make the simulation easily understandable and more reliable, it is designed to randomly pick a mobile node as subject.

The simulation environment is built using MFC's in Visual C++ with excel as the backend database. The reason for choosing VC++ to develop the simulation environment is its well defined classes for GUI development, good object structuring, and easy data base connectivity.

The main functionality of the simulation environment (figure 20) is distributed in two frames: SnapCursorDemoView and SceneView. The SnapCursorDemoView is the form view which takes input parameters and SceneView is the active floor where static and mobile nodes are placed.

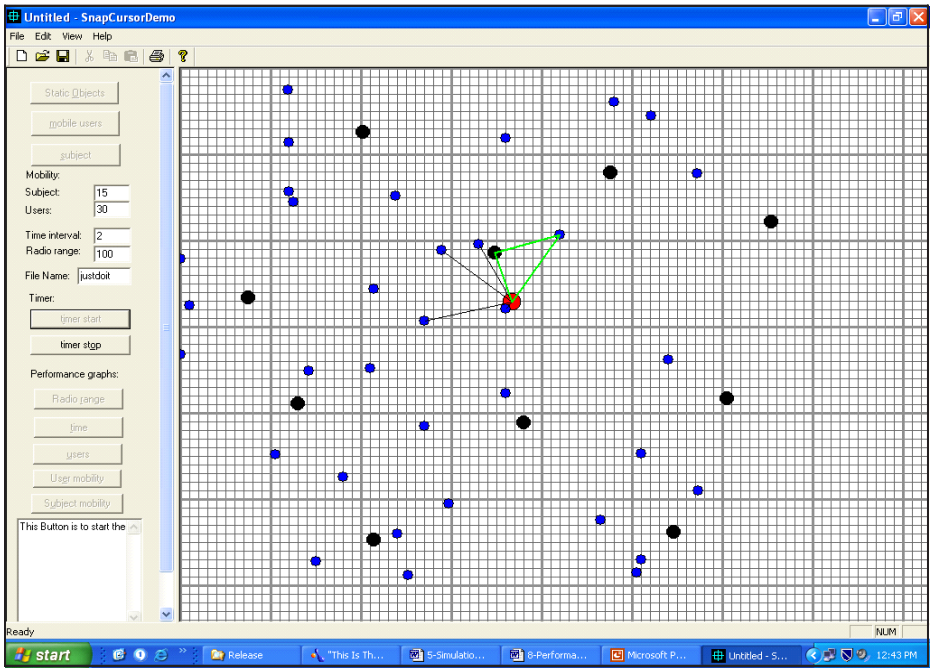

Figure 20. Typical simulation environment.

A typical localization scenario is shown in figure 21. The subject mobility plot obtained from a simulation of a 10 m by 10 m environment is shown in figure 22.

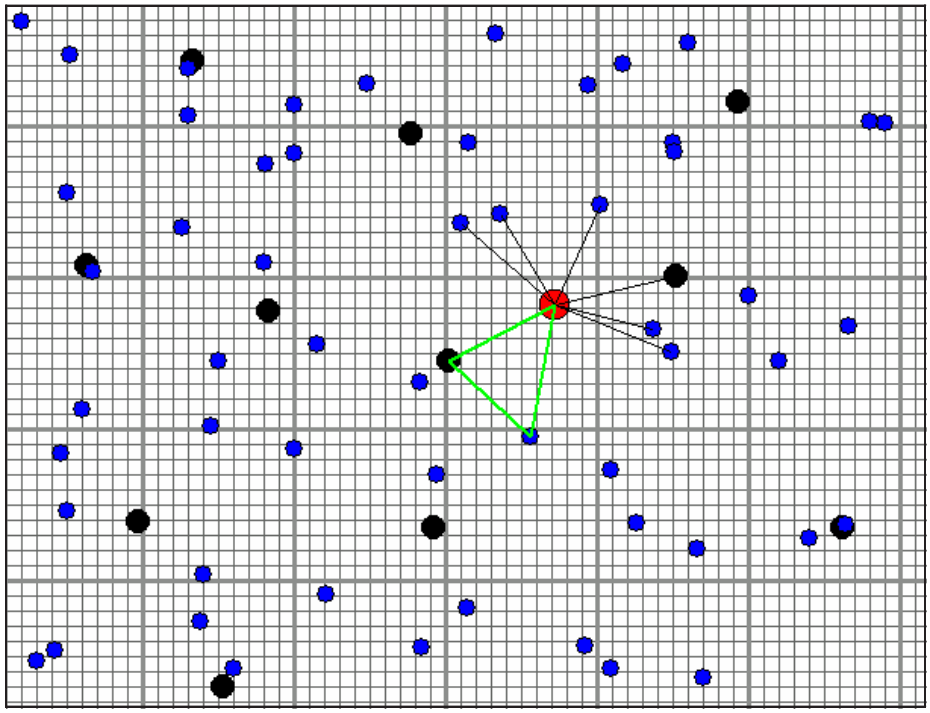

Figure 21. Scene view active floor plot.

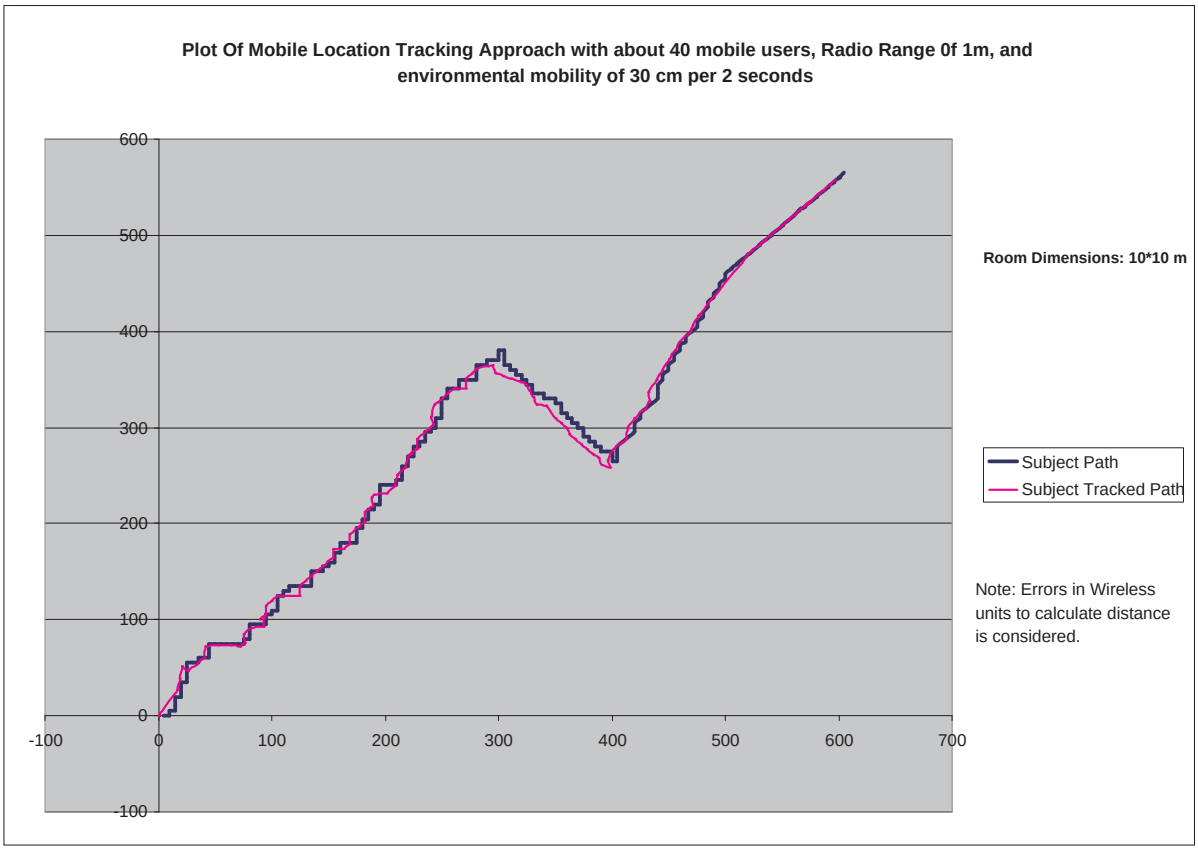

Figure 22. Subject mobility plot.

CHAPTER 6

RESULTS

This chapter describes different error evaluation methods to determine the reliability of the mobile localization approach. Below are the different kinds of tests performed, their plots, and comments on their behavior.

6.1. Euclidian Distance Error Calculation

This is calculated by a two-point distance error between the original and the tracked location. It is a technique adapted by many researchers [5, 11] to verify their localization approaches. Figure 23 shows the Euclidean distance error plot of the subject with respect to its mobility.

The maximum error in localization occurred when the subject was not able to find mobile nodes in range for a long time. The error is limited by approximate localization. The error is not a constant value because it depends on the number of mobile users in range.

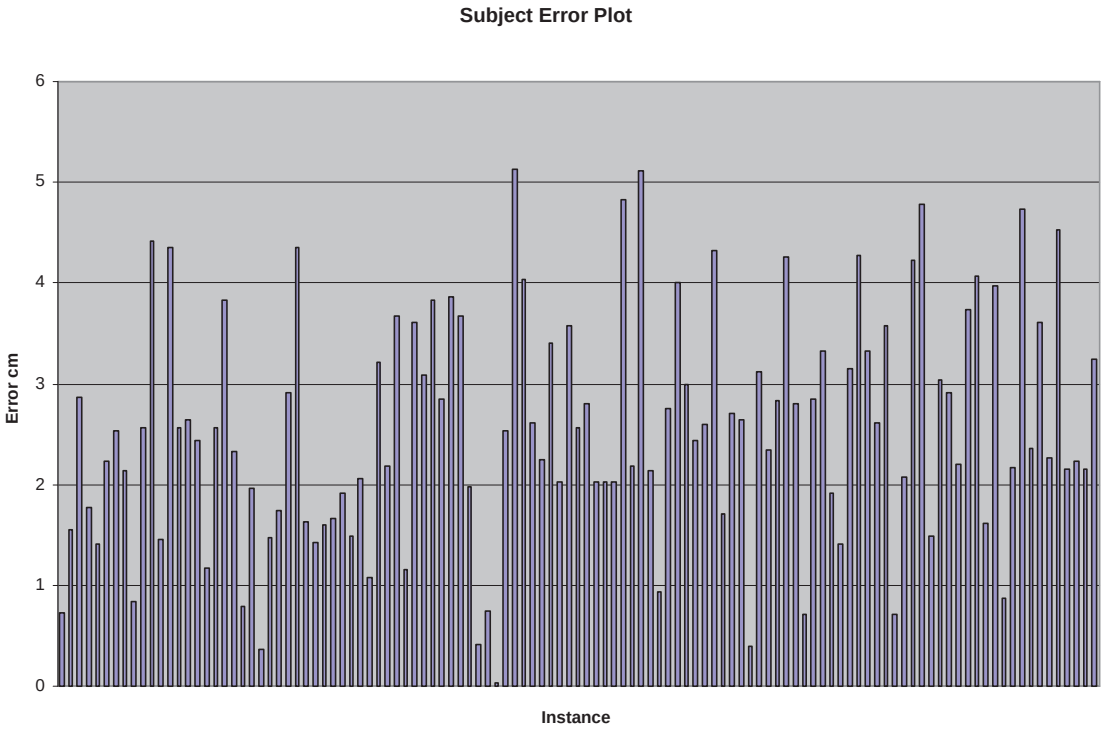

Figure 23. Euclidian distance error plot for mobile localization.

6.2. Mean, Median, Mode of Error

The mean of the ideal mobile localization approach is 3.924 cm, the median is 3.5681 cm and mode is 2.9755 cm.

6.3. Standard Deviation

This is the root mean square of the deviation. It gives the measure of how variables move over time away from the mean value.

Standard deviation

$$\sqrt{(1/(n-1)) * \Sigma(X_{di} - \overline{X_d})^2}; \tag{6.1}$$

The standard deviation for mobile localization is 2.459 cm.

6.4. Average Deviation

This is the mean of all deviations of distance measurements. Theoretically, the ratio of standard deviation to average deviation (s/a) is 1.25 when N tends to infinity.

Average deviation:

$$\sum(X_d)/n; \tag{6.2}$$

The average deviation for mobile localization is 1.888686 cm.

The ratio of standard deviation to average deviation is 1.280043.

6.5. Gaussian Error

This is represented by a symmetrical curve of normal distributions with distance error as the X-axis, and the number of times the error occurs as the Y-axis. An ideal Gaussian curve looks as shown in figure 24.

Most of the points lie within the bell-shaped curve. It is equally likely that points may lie on either side of the perpendicular within the curve.

The plot of distance error vs. times the error occurred as shown in figure 25 is in the form of a Gaussian distribution. The error is gaussian for the subject mobility from one end to the opposite end of a 10m by 10m area.

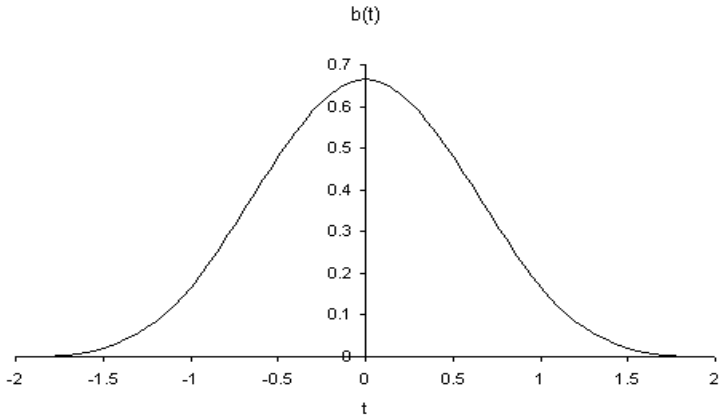

Figure 24. Ideal Gaussian curve

6.6. Parameter affecting Protocols Performance

Since mobile localization depends on node mobilty, the time interval for location refresh, the radio range and the number of mobile nodes, these determine the performance of the protocol.

6.6.1. Number of Mobile Nodes

Even though an increase in the number of nodes in the environment increases the complexity in terms of time multiplexing and beacons spread in space; it also aids in improving the performance of the location tracking approach. As the number of mobile nodes increases, the probability of the Subject finding reliable dependants increases. The performance of the protocol increases with the increase in the number of mobile nodes. The plot of performance with respect to the mobile node count is shown in figure 26.

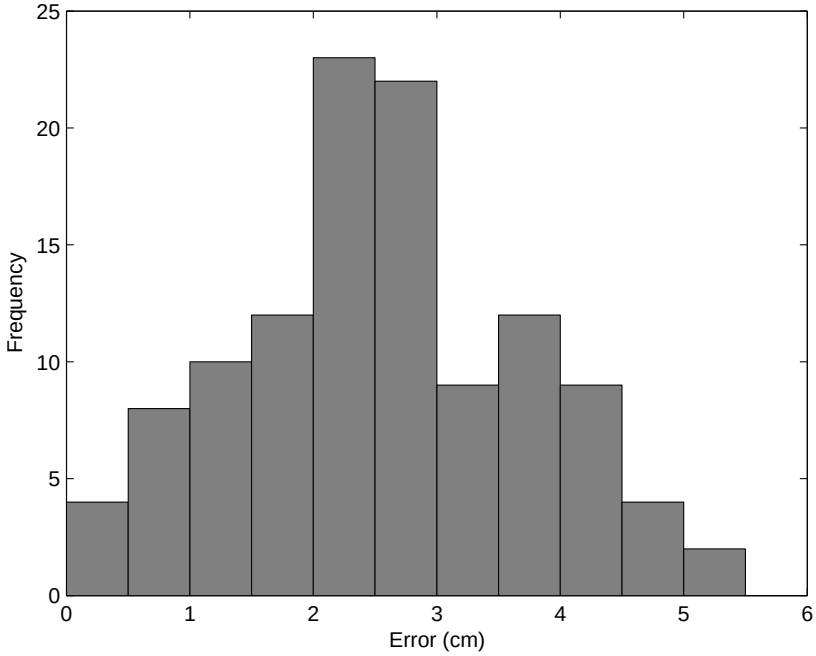

Figure 25. Error recurrence histogram

Observe that as the number of nodes increases, the error value decreases drastically and eventually comes to a constant value. The drastic decrease in error is because mobile localization need not depend on approximate localization with an increase in the number of mobile nodes. The reason for error saturation is because of a constant hardware calibration error.

6.6.2. Time Interval for Location Refresh

The Location Refresh Interval is the time interval after which the subjects gets a chance to triangulate its position from its nearest neighbors. This is an important factor which depends on the number of mobile nodes present in environment. An

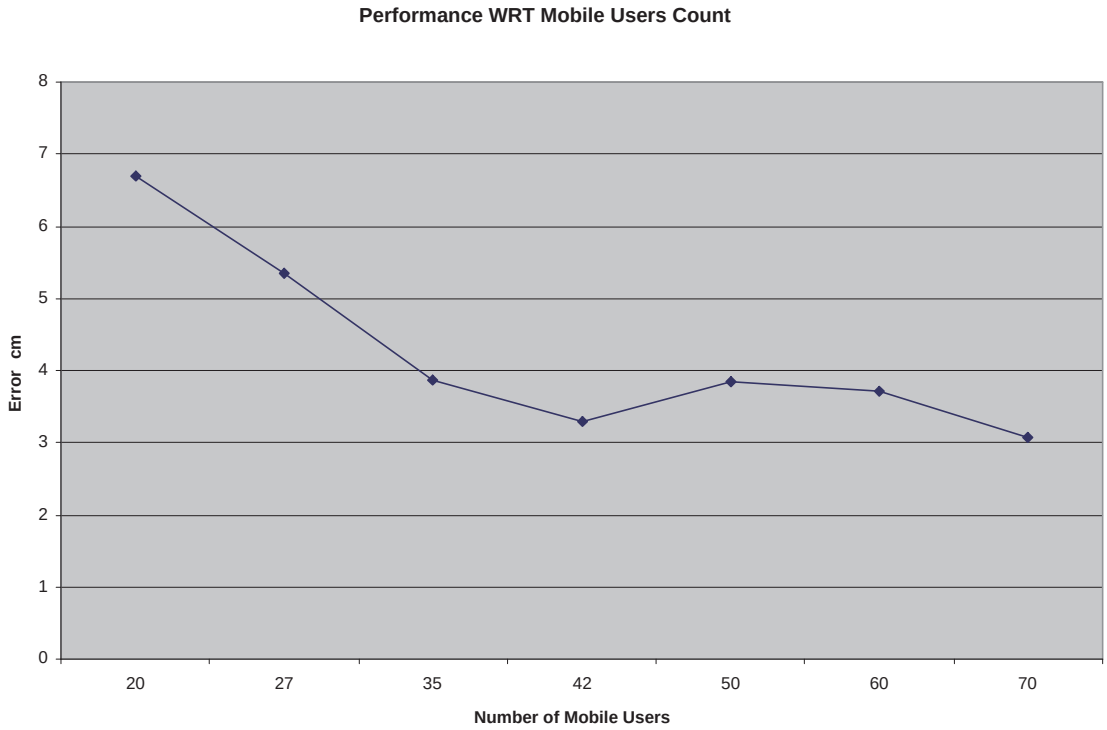

Figure 26. Performance plot wrt mobile node count

increase in the number of nodes increases the time gap for location refreshing. This leads to the mobile nodes depending on an approximate location estimate for most of the time. The plot of time refresh against the error it causes is shown in figure 27.

Observe the increase in error with the increase in refresh time as explained in section 6.7.2. The saturation of the graph after 4 sec is because the subject depends on approximate location most of the time, and the error averages to a constant value over the long subject track.

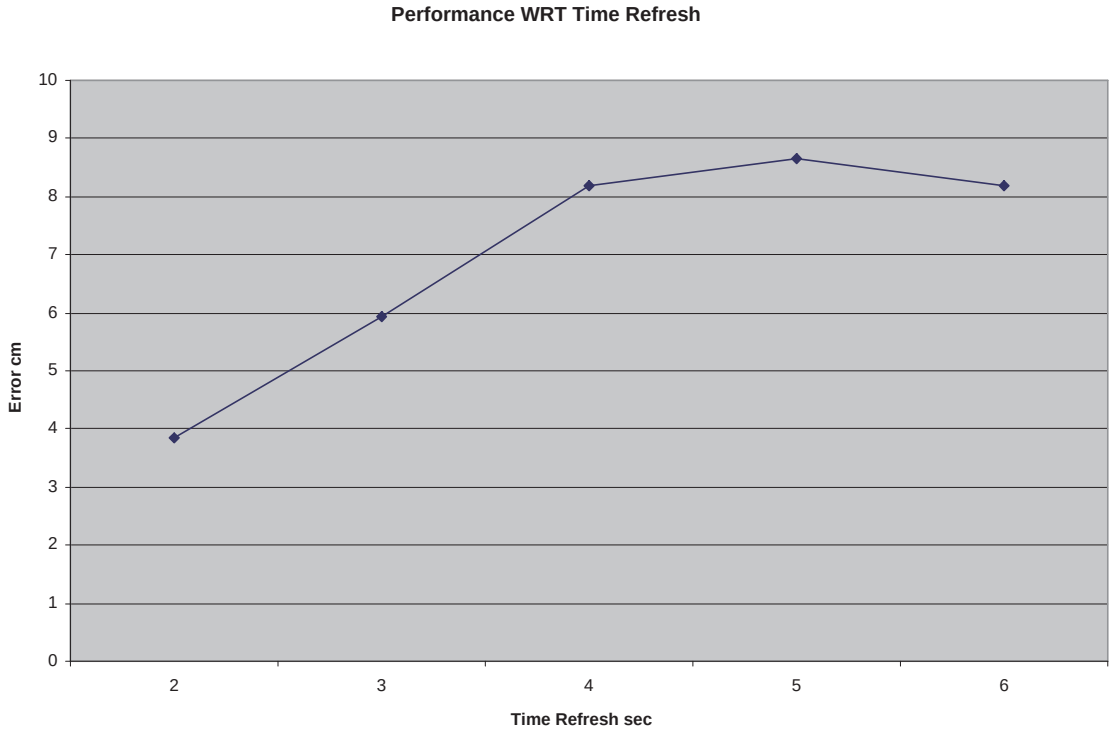

Figure 27. Performance plot wrt Time refresh

6.6.3. Mobility of the Environment

The mobility of a normal person when walking varies from 15 cm to 50 cm per sec. Mobility is an important parameter to be considered for mobile localization. As the mobility increases, the subjects probability of finding reliable dependents decreases thereby increasing the localization error. The plot of mobility against error is shown in figure 28.

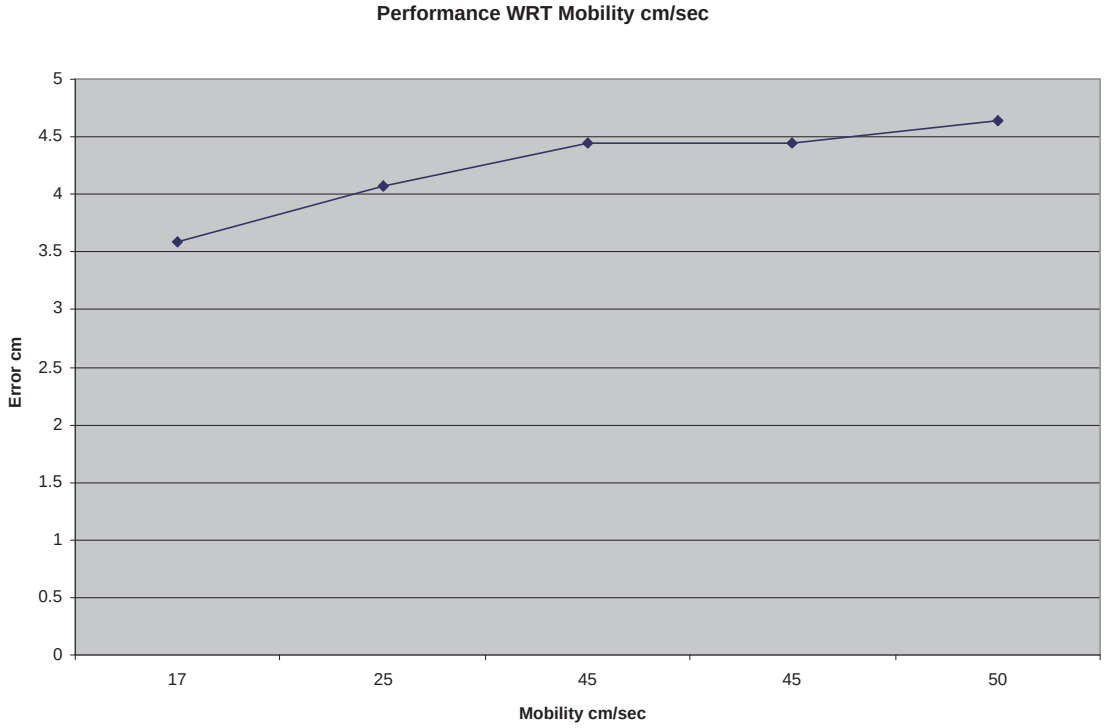

Figure 28. Performance plot wrt environment mobility

6.6.4. Nodes Radio Range

Radio range is also an important metric which affects the performance of the protocol. As the radio range increases, the scope of the nodes decreases, which in turn increases the localization error. The performance plot of error with respect to radio range is shown in figure 29.

An increase in radio range also increases the energy efficiency of the protocol as the subject can rely on the same dependents for multiple localizations. The error saturates after 100 cm.

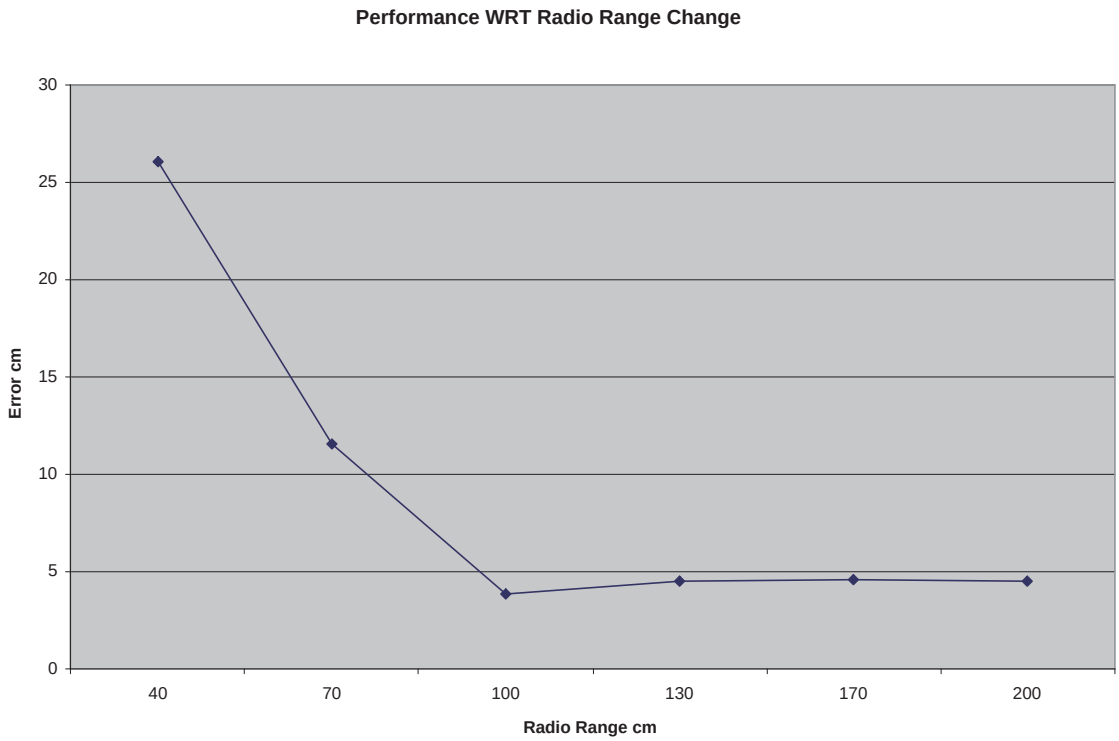

Figure 29. Performance plot wrt radio range

CHAPTER 7

COMPARISON

Though there are many locations tracking approaches available [11, 12, 13, 14], most of them rely on static sources to calibrate location. None of them explored the idea of calibrating location information from neighbors which may not be static.

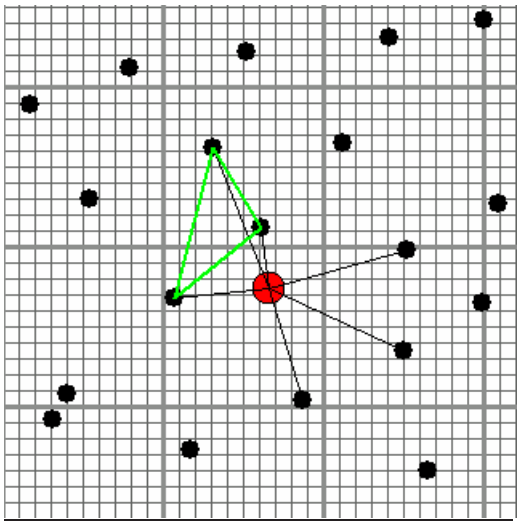

Figure 30. Mobile agent approach plot

The closest location approach similar to the mobile localization approach is "A Mobile-Agent Approach for Location Tracking in a wireless sensor network" [14]. In

mobile agent paradigm, once a new object is sensed, a mobile agent will be initiated to track the roaming path of the object. The agent is mobile since it will choose the sensor closest to the object to stay. That is, the agent actually follows the object by hopping from sensor to sensor. The agent may invite some nearby slave sensors to cooperatively locate the object and inhibit other irrelevant (i.e., farther) sensors from tracking the object. The simulation environment is shown in figure 30.

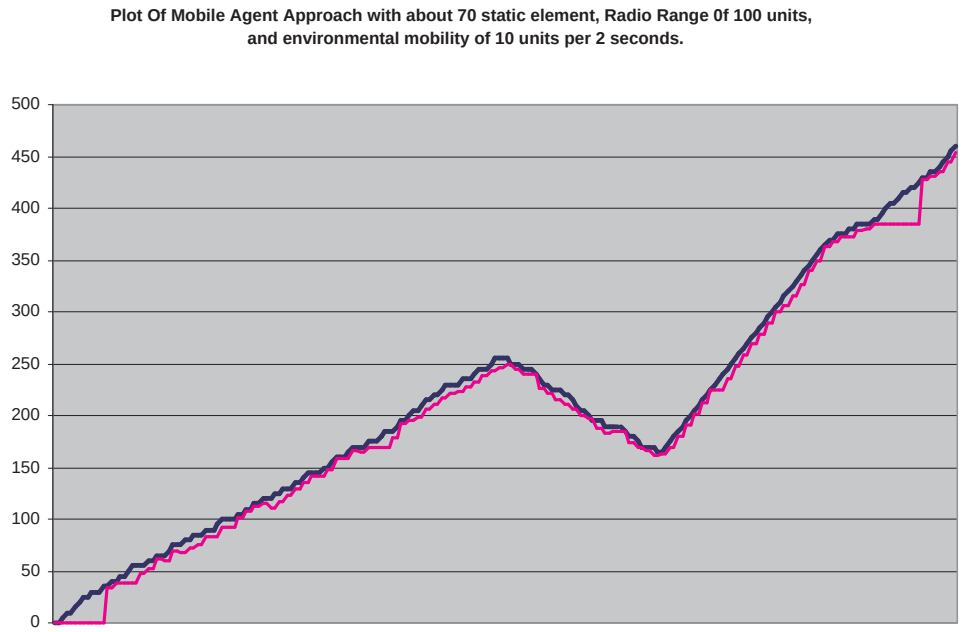

Figure 31. Mobile agent plot

As you observe, the mobile agent looks for three dependants to track location, which decreases its chance compared to a mobile localization approach which requires only two dependents. The mobile agent approach is not able to track multiple mobile agents. The agents motion and its corresponding tracked locations are shown in figure

31.

In the Mobile agent approach, the node often runs into blind spots where it cannot find reliable dependents to triangulate its position. This leads to a high error value, as you would observe from the graph in figure 32.

Also, the mobile agent approach demands a good infrastructure placed at accessible locations for the nodes to triangulate. It is more costly to implement compared to the Mobile localization algorithm.

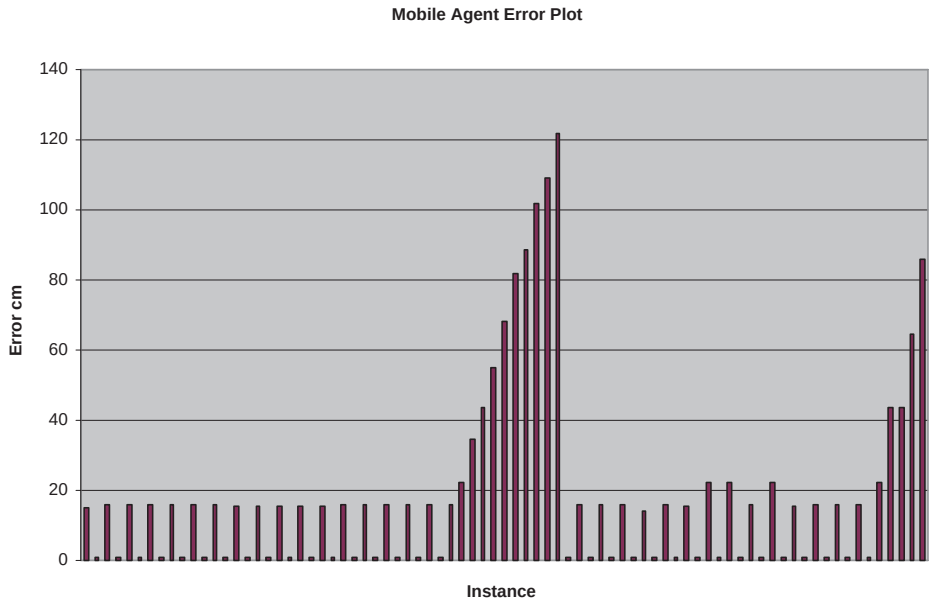

Figure 32. Mobile agent error plot

The average error value of the mobile agent approach over the entire path is 19.48775 cm, average deviation is 17.78172 cm, and the standard deviation is 27.12515 cm, almost four to five time the error of the mobile localization approach.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Every application has different goals, and every problem domain imposes new constraints on system design. Localization is unfortunately a tough problem to address, as different scenarios demand different localization systems. The Mobile localization algorithm is specifically designed for localization in an environment without a static infrastructure. It has constraints as other localization approaches, apart from which, it promises reliability, scalability and energy efficiency.

Mobile localization is a simple system. It is targeted for both indoor and outdoor applications. It is implemented with off-the-shelf hardware. All nodes in the environment run the identical algorithm and does not require costly infrastructure set up.

The algorithm is dynamic in the sense that it localizes from neighboring nodes. Because the mobile localization algorithm requires only 2 dependents, it is energy efficient compared to other systems. It is reliable in situations where a node is deprived of reliable dependents. It is robust against signal collision, corruption, and multipath effects. The proposed protocol is successfully verified by both implementation

and simulation. The performance of the protocol can be improved by incorporating doppler effects in calculating the distance between mobile nodes. The approach can be extended to any number of nodes with frequency hopping along with time multiplexing for node localization. But a trade off has to be made with respect to design complexity and cost. The Mobile localization approach can be extended for scenarios like navigating blind, assisting customers in shopping malls etc.

APPENDIX A

APPENDIX

Table 1. Notation for approximate and precise localization proof

Notation	Definition
X	Correct X coordinate
$\hat{X}$	Calibrated X coordinate
μ	Node speed
ϵ	Error in speed
t	Time
ΔT	Time interval
d	distance
$f[\epsilon]$	Approximate localization error
$g[\epsilon]$	Triangulation (precise localization) error
m	mean
σ^2	Variance

Variance due to Approximate localization method:

From the assumptions made in mobile localization approach, the initial location of each node is known. The proof makes use of notations shown in table 1.

i.e at $t = 0$,

$$X_0 = \hat{X}_0; \tag{A.1}$$

For simplicity of the derivation assume direction of motion is constant and error occurs only because of nodes speed, whose m_ϵ is 0;

at $t = \Delta T$,

$$X = X_0 + \mu * \Delta T; \tag{A.2}$$

$$\hat{X} = \hat{X}_0 + (\mu + \epsilon) * \Delta T; \tag{A.3}$$

$$f[\epsilon] = X - \hat{X} = -\epsilon * \Delta T; \tag{A.4}$$

$$\sigma^2_{f[\epsilon]} = \Delta T^2 * \sigma^2_\epsilon; \tag{A.5}$$

Variance because of precise localization:

From fig. 33, The distance between two edges of the triangle is:

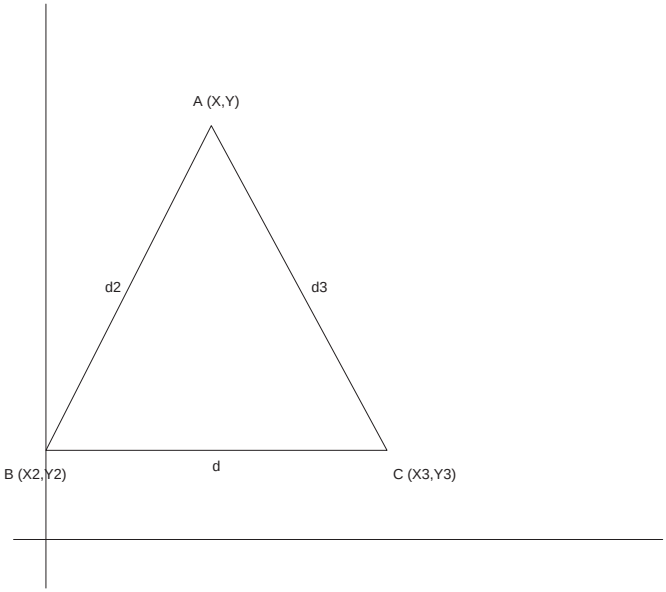

Figure 33. Node localization by Triangulation

$$(X - X_2)^2 + (Y - Y_2)^2 = d_2^2; \tag{A.6}$$

$$(X - X_3)^2 + (Y - Y_3)^2 = d_3^2; \tag{A.7}$$

$$(X_3 - X_2)^2 + (Y_3 - Y_2)^2 = d^2; \tag{A.8}$$

Let $Y_2 = Y_3$, $d_2 = d_3$, and error in distance measurement is neglected. Also X_2 be a static node.

On solving equations A.6, A.7, A.8.

at $t = \Delta T$,

$$X = [(d_3^2 - d_2^2) - (X_3^2 - X_2^2)]/2 * (X_2 - X_3); \tag{A.9}$$

$$\hat{X} = [(d_3^2 - d_2^2) - (\hat{X_3^2} - \hat{X_2^2})]/2 * (\hat{X_2} - \hat{X_3}); \tag{A.10}$$

$$g[\epsilon] = X - \hat{X}; \tag{A.11}$$

on substituting $X, \hat{X}$ and $d_2 = d_3$:

$$g[\epsilon] = -(X_3^2 - X_2^2)/2 * (X_2 - X_3) + (\hat{X_3^2} - \hat{X_2^2})/2 * (\hat{X_2} - \hat{X_3}); \tag{A.12}$$

Since X_2 is a static node, $X_2 - \hat{X_2} = 0$; Therefore

$$g[\epsilon] = (X_3 - \hat{X_3})/2 = -\epsilon * \Delta T/2; \tag{A.13}$$

$$\sigma^2_{g[\epsilon]} = (\Delta T)^2/4 * \sigma^2_\epsilon; \tag{A.14}$$

Thus proving that variance due to precise localization is less compared to variance due to approximate localization method.

REFERENCES

[1] Weiser M., "The computer for the 21st century", *Scientific American*, 265(30):94–104, 1991.

[2] Weiser M., "Hot topics – ubiquitous computing", *IEEE Computer*, 26(10):71 – 72, October 1993.

[3] Jiang C., Steenkiste P., "A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing",*The Fourth International Conference on Ubiquitous Computing"*, 246-263, September 2002.

[4] Ward A., Jones A., Hopper A., "A new location technique for the active office", *IEEE Personal Communications Magazine*, 4(5): 42–47, October 1997.

[5] Want R., Hopper A., Falco V., Gibbons J., "The Active Badge Location System", *ACM Transactions on Information Systems*, January 1992.

[6] Werb J., Lanzl C., "Designing a positioning system for finding things and people indoors", *IEEE Spectrum*, 35(9):71-78, September 1998.

[7] hopper A., "Sensor Driven communication and computation", *IEE, savoy place, London WC2R OBL, UK*, 1998.

[8] Hightower J., Borriello G., "A Survey and Taxonomy of Location Systems for Ubiquitous Computing", *Extended paper from Computer*, 34(8) p57-66, August 2001.

[9] "Indoor Radio WLAN Performance Part II: Range Performance in a Dense Office Environment", Young Design, *IEEE 802.11 Tutorials*, November 2003.

[10] Hazas M., Ward A., "A Novel Broadband Ultrasonic Location System", 264-280.

[11] Nissanka B.P., Chakraborty A.,, Balakrishnan H., "The Cricket location support system". *In Proceedings of the Sixth International Conference on Mobile Computing and Networking*, August 2000.

[12] Bahl P., Padmanabhan V. N., "RADAR: An in-building RF-based user location and tracking system", *In Proceedings of IEEE Conference on Computer Communications*, 2 775 – 784, March 2000.

[13] Nissanka B. P., Miu A. K. L., Balakrishnan H., Teller s., "The Cricket Compass for context-aware mobile applications", *In Proceedings of the Seventh International Conference on Mobile Computing and Networking*, July 2001.

[14] Tseng Y. C., Kuo S. P., Lee H. W., Huang C. F., "A mobile-agent approach for location tracking in a wireless sensor network", *Int'l Computer Symp*, 2002.

[15] Randell c., Muller H., "Low cost indoor positioning system", *In Proceedings of Ubicomp 2001*, 42 – 48, September 2001.

[16] Elson J., Estrin D., "Time Synchronization for Wireless Sensor Networks", *Proceedings of the Twenty First International Conference on Distributed Computing Systems*, April 2001.

[17] Ramadurai V., Sichitiu M. L., "Simulation-based Analysis of a Localization Algorithm for wireless adhoc sensor networks"

[18] Bulusu N., Estrin D., Girod L., Heidemann J., "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems", *In Proceedings of the 6th IEEE International Symposium on Communication Theory and Application*, July, 2000.

[19] Bulusu N., Heidemann J., Estrin D., "GPS-less Low Cost Outdoor Localization For Very Small Devices" *IEEE Personal Communications Magazine*, 7 (5): 28 – 34, October, 2000.

[20] Sichitiu M. L., Ramadurai V., Peddabachagari P., "Simple Algorithm for Outdoor Localization of Wireless Sensor Networks with Inaccurate Range Measurements" *International Conference on Wireless Networks*, 300-305, 2003.

[21] Estrin D., Govindan R., Heidemann J., Kumar S., "Next century challenges: Scalable coordination in sensor networks", *In Proc. of ACM MOBICOM '99*, 263-270, 1999.

[22] Ramadurai V., Sichitiu M. L., "Localization in Wireless Sensor Networks, A Probabilistic Approach", *International Conference on Wireless Networks* 275-281, 2003.

[23] Savvides A., Han C., Srivastava M. B., "Dynamic fine-grained localization in ad-hoc networks of sensors" *In Proc. of ACM MOBICOM '01*, 2001.

[24] Savarese C., Rabaey J. M., Beutel J., "Locationing in distributed ad-hoc wireless sensor networks", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4: 2037-40, May 2001.

[25] Beers Y., "Introduction to the Theory of Error", *Addison-Wesley*, 1957.

[26] Gay D., Levis P., Behren R.V.,, Welsh M., Brewer E., Culler D., "The nesc language: A holistic approach to networked embedded systems", *In Proceedings of Programming Language Design and Implementation*, 2003.

[27] Sallai J., Balogh J., Maroti M., Ledeczi A., "Acoustic Ranging in Resource Constrained Sensor Networks", *Technical Report*, 2004.

[28] Annamalai V., Gupta S.K.S., Schwiebert L., "On Tree-Based Convergecasting in Wireless Sensor Networks", *IEEE Wireless Communications and Networking Conference* 3: 16-20, March 2003.

[29] http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc..

[30] Addlesee M., Curwen R., Hodges S., Newman J., Steggles P., Ward A., Hopper A., "Implementing a Sentient Computing System", *IEEE Computer Magazine*, 34 (8): 50-56, August 2001.

[31] Paradiso J., Hu E., "Expressive Footwear for Computer-Augmented Dance Per-

formance", *Proc. of the First International Symposium on Wearable Computers, Cambridge, MA*, 165-166, October 1997.

[32] Wren, C.R., Azarbayejani A., Darrell T., Pentland A.P., "Pfinder: real-time tracking of the human body", *Pattern Analysis and Machine Intelligence* 19 (7): 780-785, Jul 1997.

[33] Orr R., Abowd G., The smart floor: A mechanism for natural user identification and tracking, *ACM CHI 2000 Conference proceedings* 1-6, 2000.