ABSTRACT

Mobile ad hoc networks have found many applications in recent years. Many of these applications such as climate control, habitat monitoring and disaster relief require that the ad hoc nodes know their positional information. However because of the dynamic nature of the mobile networks, it needs to be determined only based on the characteristics of the network. In this thesis, a positioning system is designed and implemented using the signal strength behavior of the radio communications on real systems.

Basically, the position of any object is defined in reference to a fixed environment. The relation a random object bears with its surroundings helps define the positional coordinates of the object. If an accurate physical relation can be obtained from the random object to other fixed objects surrounding it, a series of mathematical steps like triangulation helps to precisely locate a random object. Earlier, researchers have established that strengths of radio signals are inversely related to the distance between the sender and the receiver. Since this relation is subject to changes depending on the medium and other interferences, a relationship is derived in the thesis specific for the environment we experimented. This relation can be used to derive the topology of the ad hoc network.

Experiments are performed with laptops and handheld PCs. A mobile ad hoc network is built and the experimental results of positional coordinates of a random object are obtained and evaluated for their accuracy. A step by step procedural guide is made as an appendix to the thesis which can be useful for further experimentation purposes and make developments on the positioning system.

Chapter 1

INTRODUCTION

Recent advances in networking technologies cannot be discussed without mentioning the prolific growth in mobile networks. With the development of *IEEE 802.11* wireless standards [1], the popularity of mobile networks has increased dramatically over the past few years. The popularity of the wireless standards has motivated many vendors to develop hardware that support wireless communications.

The mobile networks can be broadly classified into [3] *Infrastructure based networks* and *Mobile ad hoc networks* also referred to as *MANETs* [4]. Though the infrastructure based networks have been in the communications market for quite sometime (cell phones, wireless Internet etc.), recently the MANET*s* have found extensive applications in many areas of personal and commercial life. Significant examples of MANET include establishing communication for emergency or rescue operations, disaster relief efforts [9], navigation systems [10], habitat monitoring in

remote places [11]. A MANET is a collection of two or more devices nodes or terminals with wireless communications and networking capability that communicate with each other without the aid of any centralized administrator. For unexpected scenarios where centralized connectivity is not possible the MANETs are an efficient solution. In a typical ad hoc network [5], nodes themselves execute discovering other nodes and delivering messages. In addition the MANET can be connected to other networks such as the Internet or General Packet Radio Service (GPRS) [6] via a gateway with minimal settings.

In infrastructure based networks, cells are defined around the central base stations. But ad hoc networks having no centralized infrastructure, the nodes have the task of both creating links with other nodes as well as forwarding data via routing. Thus routing algorithms have been developed specifically for routing among ad hoc networks [7]. Based on earlier studies on the routing algorithms we choose the Ad hoc on Demand Distance Vector Routing algorithm (AODV) [8]. We use AODV-UU an implementation of AODV routing protocol developed by the University of Uppsala for all routing tasks in the experiments conducted

Main theme of the thesis

For most of the applications mentioned such as climate control, disaster relief, navigation systems and habitat monitoring it is necessary that each node in the ad hoc network be able to determine its position with respect to its neighboring nodes. Smart homes and interactive exhibits with positioning abilities could use the positions of people wearing ad hoc nodes relative to other ad hoc nodes fixed on interactive objects to further

customize an environment. For disaster relief the randomness of the situation demands the positioning of the places free of disaster affects. Finally, habitat monitoring and navigation systems present obvious motivations for positioning.

Organization of the thesis

This thesis discusses an ad hoc positioning system with no help from Global Positioning system or infrastructure. We use an ad hoc network consisting of Toshiba Laptops and handheld devices, iPAQ 3760s and iPAQ 5550s with IEEE 802.11b standard over the 2.4 GHz ISM band. The key idea is to use the signal strength information obtained from inter-nodal communication to determine the inter-nodal distances. This distance information goes through a series of mathematical steps, called Triangulation [12], to find the position of a random node in an ad hoc network. For improving the accuracy of the triangulation additional data has to be obtained from the network.

The thesis work is documented in six chapters, Chapters 2 and 3 give a background of the theoretical concepts that are operative for this research work. Chapter 2 gives a description about the ad hoc networks and the ad hoc routing protocols. Chapter 3 discusses about the positioning system in context of history and present research. Chapters 4 and 5 discuss the experimental work and results obtained. Chapter 4 studies the usage of the AODV routing protocol implementation and the variation of signal strengths with distance. Chapter 5 discusses about position determination using the signal strength information. We finally conclude the thesis with the conclusions and future work in Chapter 6. The Appendix attached at the end of the document describes the installation

procedures and the various steps involved in building the ad hoc network of laptops and handheld PCs using AODV routing protocol.

Chapter 2

BACKGROUND

This chapter gives a brief overview of the positioning system and the ad hoc networks. We start the discussion of the positioning system in context with the widely used GPS (Global Positioning System) in Section 2.1 and proceed to describe the reasons for not using GPS for all positioning applications. The Section 2.2 discusses about the concept of positioning. Then we describe various ad hoc routing protocols in Section 2.3. Section 2.4 gives a detailed explanation of the AODV protocol as well as its real implementations. Note that the AODV routing protocol is used for all routing purposes for building a positioning system in this thesis.

2.1 GLOBAL POSITIONING SYSTEM

The concept of positioning cannot be discussed without mentioning Global Positioning System (GPS) [19].The GPS is a world wide radio-navigation system. It was implemented way back in the history of positioning systems by the Department of Defense (DoD) for military based applications. The DoD installed 24 satellites and several ground stations exclusively as a part of the positioning system. At any given point of time data needs to be collected from at least 4 satellites to know the position of an object. Since with the aid of these satellites an object can be *located anywhere on the planet* (where ground stations are located), the term "Global Positioning System" came into being.

The GPS gadgets help find the location of any object based on the position of some landmarks in the vicinity. A landmark is a physical object that has a known fixed location for the entire duration of the positioning process. The information about the landmarks like the positional co ordinates are known from records and based on the information we procure the position of any object which has a GPS receiver. The GPS has evolved over the years with developments like the Differential GPS and can give pretty accurate results.

Though the GPS was implemented initially for military applications it found its way into automobiles, boats, planes, industries and even cell phones. The mathematical principles involved in the GPS formed the basic principle for many other positioning systems.

The Global Positioning System gives pretty accurate results but it is not suitable for many of the cases involving positioning. The reasons for the above are listed below

1. The heavy cost involved in the receiver equipment

2. The size and weight of the receiver equipment.
3. Lack of control over the system since the DoD maintains the satellites.

Many applications need a GPS free positioning system that can work in absence of satellites and heavy infrastructure.

2.2 POSITIONING CONCEPTS [16]

The position of an object is conceptually defined based on a predefined frame of reference. For instance a particular house address is defined by a specific name in a specific street. The street name in a specific county, a county in a city, a city in a state and so on based on the scale we desire. Thus the position is a relative concept depending on the coordinates we choose. The GPS defines the position of an object in global coordinates. The global coordinates are irrelevant in most of the positioning systems. The problems discussed in most cases are application specific and need to be solved only in relation to the particular application. In the case of ad hoc networks the position of a node needs only to be determined in relation to the network it belongs to. Thus we can formulate a positioning system that can work without depending on the GPS satellites and receivers. In other words a GPS free positioning system can be built.

Triangulation is a technique to locate the position of an object uniquely based on the positional information provided by the neighboring landmarks. This is the technique that forms the basis for most of the positioning systems including the GPS.

Triangulation is a concept that uses the boundaries of the overlapping regions of the landmarks to determine position. An object can be uniquely positioned when at least three reference points are associated with the object in a two-dimensional space, thus

forming a triangle. The object is related to these landmarks in terms of measured distances and angles. The Ad hoc Positioning Systems (APS) algorithm we discuss uses a specific implementation of triangulation that only considers distances between the objects. The triangulation algorithm is further explained clearly in geographic terms. Depending on the context we can employ the results obtained from the geometric approach or proceed to the mathematical approach.

We use the geometric concept [16] assuming the area of deployment is a wide open space without any other obstructions. In the figure 2.2 given we can see that the object "***x***" is surrounded by three other objects ***p, q*** and ***r*** whose positions are already known. Thus ***p, q*** and ***r*** are the landmarks and based on their locations we need to determine the position of ***x***.

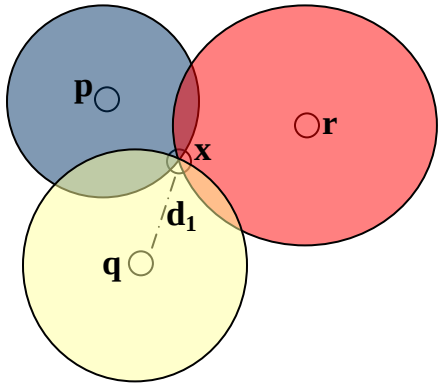


Figure 2.2: Diagrammatic explanation of Positioning System's Geometric concept

If ***x*** is located at a distance $\boldsymbol{d_1}$ from ***q,*** then ***x*** lies somewhere on the perimeter of a circle with center ***q*** and radius $\boldsymbol{d_1}$. Similarly ***x*** can be located on a circle around p. The circles around ***p*** and ***q*** intersect in precisely two points. When a similar circle is constructed around ***r*** we can pinpoint exactly one intersection point of all the three circles

and that will be the precise location. Thus with the help of three landmarks we can precisely know the location an object in a two dimensional space.

The algorithm can be extended to a three-dimensional space and the position of an object can be precisely determined using four landmarks. In the case of three-dimensional space the spheres around the landmarks are considered and the intersection of four such spheres around the landmarks gives the precise location of the unknown object. In the absence of any errors, these reference points would provide a perfect solution. But practically no real environment is error free. Thus the location determined will be erroneous to some extent. Errors can result from situations with poor geometry, or poor constellations. In this context, the constellation refers to the topology of the reference points with respect to the unknown. Singularity failures can also result over time in iterative algorithms if errors compound enough to greatly imbalance the significance of enough reference points. Mathematical models are proposed to minimize errors and give better positioning results.

Given the positional coordinates and the range estimates from the unknown object we can mathematically state the triangulation principle using the following representation.

If the coordinates of the ith reference point are (x_i, y_i, z_i), the positional coordinates of a random object u are (u_x, u_y, u_z); d_i is the distance that is computed from the landmark "***i***" to the unknown object then the coordinates of the unknown object can be calculated using the following equation [37]

$$u = (P'P)^{-1} * P'Q$$

where

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, P = -2 * \begin{bmatrix} (x_1 - x_n)(y_1 - y_n)(z_1 - z_n) \\ (x_2 - x_n)(y_2 - y_n)(z_2 - z_n) \\ . \\ . \\ (x_{n-1} - x_n)(y_{n-1} - y_n)(z_{n-1} - z_n) \end{bmatrix}$$

and

$$Q = \begin{bmatrix} r_1^2 - r_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 - z_1^2 + z_n^2 \\ r_2^2 - r_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 - z_2^2 + z_n^2 \\ . \\ . \\ r_{n-1}^2 - r_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 - z_{n-1}^2 + z_n^2 \end{bmatrix}$$

2.3 AD HOC ROUTING PROTOCOLS [7]

The importance in MANETs is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Each node in a wireless ad hoc network functions as both a host and a router. The network topology is in general dynamic, because the connectivity among the nodes may vary with time due to node mobility, node departures and new node arrivals. Hence, there is a need for efficient routing protocols to allow the nodes to communicate.

Conventional routing protocols are based on either *distance vector* or *link state* algorithms [12]. Distance vector protocol makes shortest path decisions based on a hop count metric, while link state makes decisions based on cost of each link. They are not designed for the type of dynamic topology changes that may be present in MANET. Since topology changes at any time in MANET scenarios, convergence to stable routes may be quite slow, particularly with distance vector algorithms. Link state protocols will

take a path which has more hops, but that uses a faster medium over a path using a slower medium with fewer hops. But distance vector algorithms require less processor overhead, compared to link state. The speed of convergence, in distance vector algorithms, may be improved by sending routing updates more frequently, but such a shift only wastes more bandwidth and battery power when topology does not change much. Thus we see that for a highly dynamic topology like that present in the MANET the tradeoffs are pretty unbalanced for conventional routing protocol. Thus we need to design new routing protocols which give us a proper balance of convergence speed and minimum bandwidth utilization.

A major focus of research on ad hoc networking has been on routing protocols. Over the past few years many routing protocols have been proposed. For the purpose of convenient study they have been broadly classified [14] as

- Proactive routing protocol
- Reactive routing protocol

Proactive or static routing protocols are like the distance vector algorithms and a routing table describing the network topology is maintained with every node. A popular proactive routing protocol is Dynamic Sequencing Distance Vector (DSDV) [12].

As the name suggests Reactive or Dynamic routing protocols are more “on demand” and routes are established as the necessity arises. A popular reactive routing protocol is Dynamic Source Routing (DSR) [15]. The DSDV and DSR routing protocols are explained subsequently in this section.

DSDV is a hop-by-hop distance vector routing protocol which requires each node to periodically broadcast routing updates. All nodes keep a routing table that holds the

routes for all reachable nodes. The advantage of this approach is that a packet can be forwarded immediately if there is an entry for its destination in the routing table. DSDV routing protocol is derived from a classical distance vector algorithm, *Distributed Bellman-Ford* (DBF) algorithm, where in each node maintains the shortest distance to all destinations through all of its neighbors. Periodically each node creates a vector containing shortest distance to each destination, and sends this vector to its neighbors. Upon receiving the vector from a neighbor, a node updates its minimum distances to all the destinations via this neighbor. To avoid the routing loops that can be caused by the dynamic nature of the ad hoc network, the DSDV uses sequence numbers. By using sequence numbers a data packet can be prevented from visiting a single node more than once. Though the DSDV shows an improvement over the Distance vector algorithm by avoiding routing loops the bandwidth requirement is still high.

DSR uses source routing rather than hop-by-hop routing. When using source routing, each packet to be routed carries in its header the complete, ordered list of nodes through which the packet must pass. A key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they receive, since the packets themselves already contain all the necessary routing information. This fact, coupled with the dynamic, on-demand nature of the DSR's route discovery, completely eliminates the need for the periodic route advertisement and neighbor detection packets, presented in proactive protocols.

In the following section we study another routing protocol which combines the features of both the DSDV and DSR routing protocols

2.4 AODV (AD HOC ON-DEMAND DISTANCE VECTOR) [8]

AODV routing combines the features of both the DSDV and DSR routing protocols to give the advantages of both the routing protocols. It is basically an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes but all the nodes in the route maintain the path information as long as the data transfer takes place. AODV uses sequence numbers like the DSDV to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request/route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number ***greater than or equal to*** that contained in the RREQ. If the sequence number corresponding to the destination is less than the sequence number carried by the RREQ the path containing the intermediate node is eliminated (to avoid loops) and the RREQ is rebroadcast. If the former case is true, the intermediate node unicasts a RREP back to the source. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

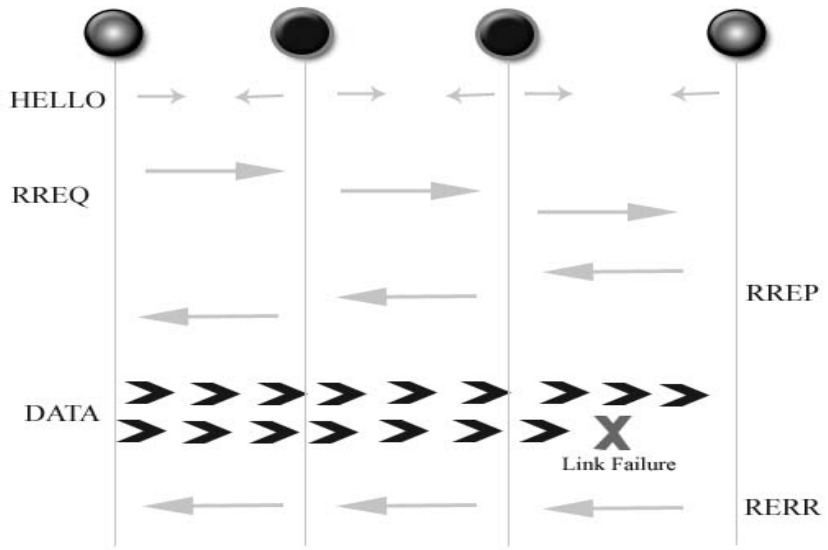


Figure 2.1: Ad hoc On-demand Distance Vector routing

As the RREP propagates back to the source, nodes set up "*forward pointers*" to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically transmitted from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable

destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

EXISTING IMPLEMENTATIONS

Few implementations of AODV that are publicly available are *mad-hoc* [21], *AODV-UIUC* [22], *AODV-UU* [23] and *AODV-UCSB* [24], which are user space implementations and *kernel AODV* [25], which is a kernel space implementation. The most publicly recommended implementations of AODV are kernel AODV, AODV-UU and AODV-UCSB [26]. Kernel AODV, a kernel space implementation, has few advantages; it operates faster than user space implementation and it does not require any mechanism for transferring packets from kernel to user space and vice versa. However, it is less portable, difficult to maintain, reduces the protocol functionality and weakens memory management. Also kernel AODV has some known bugs like memory leaks, assertion failures, routing loops etc [27]. Thus the user space implementations are preferred.

Research has been conducted in the MCRL (Mobile Computing Research laboratory) at Cleveland State University[28], on AODV-UU and AODV-UCSB. The implementations are from Uppsala University (AODV-UU) and University of California at Santa Barbara (AODV-UCSB), and both the implementations are implemented in Linux and written in the C language. Comparison experiments performed on the two implementations show that AODV-UU implementation gives better performance results than the other. Thus we use the AODV-UU for all routing purposes in this thesis work.

CHAPTER 3

NETWORK POSITIONING SYSTEMS

We have discussed about the general positioning systems and the mathematical concepts underlying the functionality of the positioning systems. This chapter discusses about the positioning systems in context of mobile networks. We start the discussion with the ranging techniques that are used to find the positional relation among the network nodes in terms of internodal distances, angles of orientation etc. When this data goes through a series of mathematical steps the positional coordinates can be determined. Section 3.1 covers these ranging techniques. The Section 3.2 gives brief explanations of a few network positioning systems researchers have been working on. We conclude the chapter in Section 3.3 where we discuss the various problems faced in building a positioning system.

3.1 RANGING TECHNIQUES

The position of a random node in a network can be determined in relation to the other nodes in the network using the characteristic properties of the network such as the strength of the signals propagated, orientation of the antennas etc. There are 4 algorithms from which we can choose based on the capabilities of the network system.

Hop based estimation [37]

This scheme relies only on the connectivity of the network. The number of hops between different nodes is counted and based on the hop count the position of the random node is determined.

- Using the AODV routing all the nodes get shortest paths to the landmarks in hops. Each node maintains a table $\{X_i, Y_i, h_i\}$ and exchanges this information with its immediate neighbor. X_i, Y_i are the coordinates of the landmark 'i' and h_i is the distance to the landmark in hops
- Each landmark calculates the average distance of one hop based on its distances to the other nodes and landmarks. This information is propagated as a Correction to all the nodes in the neighborhood.

  Correction $c_i = \sum p_j / \sum h_j$, $i \neq j$, all landmarks j heard by i

  h_j is the shortest distance, in hops from node i to node j and p_j is the straight line distance between i and j
- The nodes calculate the distances to the landmarks finally based on the corrections received.

This technique is easy to use and relies only on connectivity. The major disadvantage of this estimate is that it can be applied only to networks which have uniform physical characteristics in all directions (isotropic).

Distance based estimation (Euclidean propagation) [37]

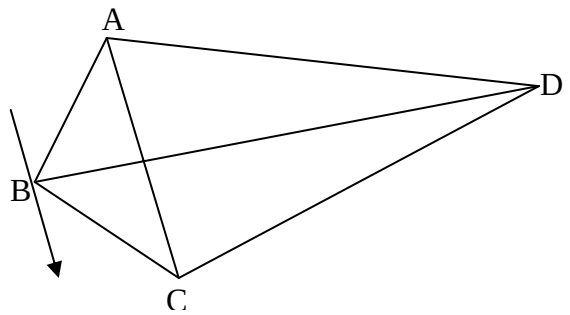


Figure 3.1: Distance based estimation

The position of a node is obtained by measuring the distances between the ad hoc nodes in the network. A table with the mapping between the distance and positional coordinates is built and maintained. At a later point of time when we have distance estimates we refer to the table and find the location of the node by mapping the distance information to the positional coordinate. This technique can be logically explained in two steps *distance estimation* and *Mapping procedure.*

First we discuss about the distance estimation. Distance Estimation in wireless ad hoc networks can be done using certain properties of the network like the strength of the signal received or based on the time taken by the signal to traverse the path from source to the destination.

The strength of RF signals is directly related to the distance from the source. In vacuum space (ideal conditions) the signal strength is inversely proportional to the square of the distance, but in experimental conditions such a consistency is not possible because

of various interferences and path loss effects. However the various ad hoc networking experiments show that strength of the signal received is directly related to the distance i.e. decreases with distance. Based on the above experimental fact the location of a random node is estimated by the mapping procedure discussed later.

It is universally applicable for RF signals that they travel at the speed of light. Thus given a constant velocity of the signals, the distances can be estimated using the time taken for the RF signals to travel from the source to destination i.e. Time of Arrival (ToA). Based on the distance estimates the location of a random node can be determined using the mapping procedure discussed next.

The mapping procedure is the second step in this ranging technique. It is necessary that any node B has 2 neighbors A and C which know the range measurements to landmark D. B knows the range measurements to A and C and also know the distance between A and C. So given the quadrilateral ABCD and known values AB, BC, AC and AD, CD we can calculate the 2nd diagonal BD i.e the range measurements between the node B and D. Once the ranges have been measured the node A can use triangulation technique to find its location with respect to the landmark D.

Angle based estimation [37]

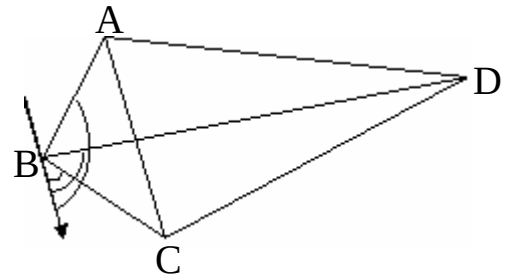


Figure 3.2: Angle based estimation

This technique uses orientation angles of antennas attached to network nodes. Angle measurements for B can be calculated with respect to landmark D if we know all the angles in the triangles BAC and ACD. Thus the knowledge of the radial information of A and C with respect to the landmark D gives the information of the neighbor B. In case of angle based technique similar to the Euclidean method we use GPS but here we use the angular information. Thus the propagation of orientation information coupled with triangulation helps us to find the position of the node B. Nevertheless angle measurements need specific hardware like the antenna arrays that are expensive to implement and maintain. Thus they are not the best choice where cost is a criterian.

Distance and angle based estimation

Once we make the range and angle measurements we can make use of both the measurements to pinpoint the location of the nodes subject to accurate calculations. They can be applied in the case of any kind of network atmosphere isotropic or anisotropic. But the expense we need to bear is a drawback.

Experimental results obtained by positioning experts show that with the use of more sophisticated measurement hardware the positioning accuracy can be greatly improved

3.2 POSITIONING SYSTEMS

The motivations for determining the location of nodes arise from many applications of ad hoc networks. Researchers are working in many theoretical directions

to obtain the position information of nodes in a network. Some of the related work in the field of positioning systems is discussed further in this section.

3.2.1 In-building Infra Red (IR) networks [35]

Infra Red signals exhibit some properties like reflection and refraction that can be utilized for building a positioning system. The *Active Badge* system [29, 30], based on IR technology, was an early and significant contribution to the field of location-aware systems. In this system, a badge worn by a person emits a unique IR signal every 10 seconds. Sensors placed at known positions within a building pick up the unique identifiers and relay these to the location manager software. While this system provides accurate location information, it suffers from several drawbacks. It scales poorly due to the limited range of IR, it incurs significant installation and maintenance costs, and it performs poorly in the presence of direct sunlight, which is likely to be a problem in rooms with windows.

Another system based on IR technology is described in [36]. In this system IR transmitters are attached to the ceiling at known locations in the building. An optical sensor on a head mounted unit senses the IR beacons, which enables the system software to determine the user's location. This system suffers from similar drawbacks as the Active Badge system.

3.2.2 MDS-MAP algorithm [42]

The MDS-MAP method needs only basic information that is likely to be already available, namely, which nodes are within communications range of which others. The

method has three steps. Starting with the given network connectivity information, it uses an all-pairs shortest-paths algorithm for roughly estimating the distance between each possible pair of nodes. Then it uses multidimensional scaling (MDS), a technique from mathematical psychology, to derive node locations that fit those estimated distances. Finally, the resulting coordinates are normalized to take into account any nodes whose positions are known. If in addition to the connectivity information the distances between neighboring nodes can be estimated, that information can be easily incorporated into the pair wise shortest-path computation during the first step of the algorithm. MDS yields coordinates that provide the best fit to the estimated pair wise distances, but which lie at an arbitrary rotation and translation. If the coordinates of any nodes are known, they can be used to derive the transformation of the MDS coordinates that allows the best match to the known positions. Only three such `anchor nodes' are necessary to provide absolute positions for all the nodes in the network.

There are two possible outputs when solving the localization problem depending on the number of landmarks.

- Relative map
- Absolute map

The task of finding a relative map is to find an embedding of the nodes into either two or three-dimensional space that results in the same neighbor relationships as the underlying network. Such a relative map can provide correct and useful information even though it does not necessarily include accurate absolute coordinates for each node.

The task of finding an absolute map is to determine the absolute geographic coordinates of all the nodes. If sufficient number of anchor nodes is available the absolute coordinates of the nodes can be found.

3.2.3 Truncated Singular Value Decomposition (TSVD) based location estimation [34]

Using the TSVD an ad hoc node is localized to a network by mapping the distance information onto proximity information and the location of the node is determined. A coordinate space is defined such that each axis corresponds to a reference set of objects, and the coordinate values of an object are the distances from the object to the reference points. This is built to estimate locations in anisotropic networks. Most of the network conditions where locationing needs to be employed are anisotropic. Anisotropic characteristics result from various factors such as geographic shape of the region, variable node densities, irregular radio patterns, anisotropic terrain etc.

The TSVD based technique infers the network topology based on the geometric structure or other network attributes such as hop count. The technique can explained in 2 major steps

- Information Collection
- Linear transformation calculation

In the information collection stage the geographic distance and the proximity information between nodes is collected. Every node initializes a beacon list which contains the location and proximity information for beacon nodes. The beacon nodes broadcast their information in probing packets. All the nodes constantly update this

information. The proximity information can be hop count or any other characteristic of the network. The beacon list is periodically updated using packet flooding.

In the second step the proximities between beacon nodes with known geographic locations are analyzed and an optimal linear transformation is derived. This linear transformation which describes the relationship between proximities and the geographic distances is called the Proximity-Distance map. Depending on the right choice of landmarks this transformation map retains the components of proximities in all directions, thus it accurately characterizes anisotropic network topologies.

3.3 INACCURACY IN POSITIONING SYSTEMS [31]

There are many positioning systems being proposed in the wireless market but the location results obtained are far from accurate in most of the cases. This is because of many reasons such as physical obstacles, interferences due to common usage of the same transmission band and media etc.

The prominent factors affecting accuracy can be stated as

- Multi path propagation
- Non Line of Sight
- Multiple access interference

3.3.1 Multi path propagation [31]

Multi path Propagation is the one of the primary sources of error in the AoA and the RSS techniques. It also affects the time based positioning systems.

Positioning problems are often encountered in areas where there are a lot of obstructions such as debris in disaster relief; concrete walls, doors etc in indoor locations; trees and rocks in wildlife monitoring etc. RF signals can reflect off of these obstructions. Any given mobile device may receive both the primary signal and one or more reflections of that same signal .This phenomenon is referred to as multipath. Strong reflected signals cause interference by adding or subtracting to primary signal amplitude. Reflections can also leave gaps in RF coverage by canceling out the primary signal. When the reflected rays arrive within a chip period of the first arriving ray the errors are more prominent. If the first ray arrives with less power than later arriving rays, the correlation delay estimators will detect a delay in the vicinity of these later arriving rays.

Antenna diversity can be used to combat multi path propagation. A mobile node with multiple antennas can continuously sample incoming signals, choose the input source (antenna and receiver) with the best signal and then can use the chosen antenna for the next transmission.

3.3.2 Non Line of Sight (NLOS) [32]

NLOS propagation will greatly bias the TOA and TDOA measurements even when high-resolution timing techniques are used and there is no mutipath interference, because signal arriving at the base station for the mobile node is reflected or diffracted and takes a longer path than the direct path. Among the sources of error, NLOS error is a major error factor that affects the accuracy of time-based positioning system. Therefore, it is very important to identify the NLOS propagation and mitigate NLOS error.

The range measurement can be modeled as

$$r_i(t_n) = d_i(t_n) + n_i(t_n) + NLOS_i(t_n)$$

where $d_i(t_n)$ denotes the true distance between the mobile target and the ith base station at time t_n, $n_i(t_n)$ denotes the measurement error which can be modeled as a zero-mean additive Gaussian random variable with variance σ_i^2. $NLOS_i(t_n)$ is the NLOS error which can be treated as a non-negative random variable with unknown probability distribution in prior.

NLOS error can be identified based on the observation that the measured range will show a significantly large average deviation than that without NLOS error. In the first step, NLOS measurement is identified by using the time history of the range measurement from each mobile node individually and the apriori knowledge of the standard deviation of the standard measurement noise. At the base station, the range measurements are first smoothed. When there is no NLOS error, the measured range deviates by the standard deviation of the measurement noise $n_i(t_n)$ but if the NLOS error is present, the measured range will deviate from the smoothed curves on the average by

$$\sigma_i^{'} >> \sigma_i$$

Therefore, the proposed NLOS identification technique essentially requires a comparison of the standard deviation of a sample statistic with the known standard deviation of that statistic under the hypothesis of LOS measurements. Then we can have

$$\begin{array}{ll} \text{no NLOS error}, & H0: \quad \sigma_i^{'} \approx \sigma_i \\ \text{with NLOS error}, & H1: \quad \sigma_i^{'} >> \sigma_i \end{array}$$

After the NLOS error is identified by rejecting $H0$, and then LOS reconstruction is achieved to correct NLOS error. The LOS is reconstructed in two steps.

- The data are smoothed using an *n*th order polynomial fit. This is done assuming that the major effect of the NLOS error is to bias the data.
- After the data is smoothed, the deviation of each value of the measured range from the polynomial fit can be computed. Assume the $n_i(t_n)$ has a value over $(-\alpha, \alpha)$. Given a sufficient observation interval, the maximum deviation of the measured range below the smoothed curve will be very nearly equal to $d_i(t_n) - \alpha$. So a smoothed curve can be corrected as the estimate of the LOS range.

3.3.3 Multiple access interference [31, 32]

Modern wireless networks function in the 2.4 GHz ISM band. This 2.4 GHz band is divided into 14 channels at different frequencies (2.412GHz to 2.484GHz). The variability of wireless channels presents both challenges and opportunities in designing wireless communication systems. With the tremendous growth of wireless networks channel reuse is inevitable. To maximize throughput for a given channel the link needs to adapt to the actual channel conditions, changing the transmitter power level, antenna beam pattern, equalizer settings, and possibly the symbol rate and constellation size. The attenuation and directionality of signals makes re-use of the time/frequency resources in space possible, permitting a large number of users to access the shared medium.

Techniques such as the Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) are defined for channel reuse. Though they contribute to the growth of wireless networks multiple users cause mutual interference to one another. For an acceptable quality of service, each user will typically need a signal to interference ratio (SIR) above some

target. The techniques which enhance reliability of a link also affect the interference seen by other users. A fundamental limit on the capacity of a multiple access wireless system is imposed by the inability of the various users to perfectly estimate and predict the time-varying channel and interference.

Chapter 4

SIGNAL STRENGTH EXPERIMENTS

Positioning of a random node in a network is a complex problem and researchers are specifically focusing in solving the problem by using only the network parameters. A lot of research is going on in this area and various network characteristics have been experimented upon to build a positioning system. Experiments have been performed to study the variation of various network characteristics such as strength of the signals received (RSSI), time of arrival (ToA), time difference of arrival (TDoA), orientation of the RF antennas i.e. Angle of arrival (AoA) etc.

This thesis work focuses on real time experimental results and focuses on the strength of the signal propagation to derive the position of a random node. Various ad hoc networking scenarios have been created, experimented upon and studied in detail to observe the behavior of signal strength and how it can be utilized to design a positioning

system. This chapter gives a description of the experimental scenarios and the results on signal strengths.

4.1 SIGNAL STRENGTH vs DISTANCE

The inverse square law by Newton states that "some physical quantity or strength is inversely proportional to the square of the distance from the source."

The inverse square law of propagation applies to radio waves and gives a mathematical model for determining the intensity of an emitted radio signal at a certain distance from the source. But we know that the inverse square is defined for vacuum space. In real life scenarios each environment presents a multitude of interference and multi-paths that make it more difficult to reliably predict the signal attenuation parameters and their distance correlation. Research in the related area has led to many radio propagation models. However radio propagation in mobile wireless channel is mainly influenced by three major factors, attenuation due to path loss, shadowing due to obstacles, and fading due to multiple paths. The models explaining these effects are discussed subsequently in this section.

- Path loss model [39]

In this model the path loss $L_p(d) \propto d^{\kappa}$, where d is the distance and κ is path loss exponent. The path loss exponent varies in different environments. The Newton's law is a special case of the path loss model and is defined in free space where $\kappa = 2$. The path loss exponents for various environments are defined in the [39].

- Shadowing model [39]

The shadowing model describes the signal propagation when the mobile node moves behind a building, a hill or other obstacles. In [39] the shadowing model consists of two parts. The first one is known as path loss model, which also predicts the mean received power at distance d, denoted by $\overline{P_r(d)}$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0}\right)^{\kappa}$$

So,

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\kappa \log\left(\frac{d}{d_0}\right)$$

The second part of the shadowing model reflects the variation of the received power at certain distance. It is a log-normal random variable, that is, it is of Gaussian distribution if measured in dB. The overall shadowing model is represented by

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\kappa \log\left(\frac{d}{d_0}\right) + X_{dB}$$

where X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}. σ_{dB} is called the shadowing deviation, and the above equation is also known as a log-normal shadowing model.

- Fading model [38, 40]

A fading model describes how propagation is affected by multiple path signals. Rayleigh fading and Ricean fading are two common model for mobile wireless channel. Rayleigh fading model describes the NLOS propagation. According to this model the amplitude fading, α, follows a Rayleigh distribution with parameter σ_z^2,

$$f_\alpha(x) = \begin{cases} \dfrac{x}{\sigma_z^2} e^{-\frac{x^2}{2\sigma_z^2}} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Ricean fading model describes the LOS propagation. According to this model the amplitude fading, α, follows a Ricean distribution given by

$$f_\alpha(x) = \begin{cases} \dfrac{x}{\sigma_z^2} e^{-\frac{x^2+\alpha_0^2}{2\sigma_z^2}} I_0\left(\dfrac{\alpha_0 x}{\sigma_z^2}\right) & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where α_0^2 is the power of LOS component and is non-centrality parameter, and $I_0(\cdot)$ is the zero-order modified Bessel function of the first kind and is given by

$$I_0(x) = \frac{1}{2\pi}\int_0^{2\pi} e^{x\cos\theta} d\theta$$

4.2 MEASURING UNITS [41]

The signal strength has various measures defined for use depending on the context. There are four units of measurement that are all used to represent RF signal strength. These are

- mW (milliwatts)
- dBm (dB-milliwatts)
- RSSI (Received Signal Strength Indicator)
- Percentage signal strength.

All of these measurements are related to each other, some more closely than others. It is possible to convert from one unit to another, albeit with varying degrees of accuracy, and not always in the extremes of the measurement range.

When signal strength is measured in milliwatts (mW),the mW signal level is, simply, the amount of energy present. A typical wireless access point or quality wireless client NIC has a rated output of 100 mW. Because of the peculiarities of measurement, measuring RF signal strength in mW units is not convenient. This is because the signal strength does not vary in a linear manner but is guided by the inverse square law. Because of the signal strength properties a convenient logarithmic measurement dBm (dB-milliwatt) was developed that can easily converted to and from the older mW. The conversion is done using the equation below.

$$dBm = \log (mW) * 10.$$

The IEEE 802.11 standard defined a mechanism by which energy is to be measured by the circuitry on a wireless NIC. This numeric value is an integer called the Received Signal Strength Indicator (RSSI). The RSSI offers an accuracy of one byte value to measure the signal strength i.e. the signal strength value has a range of (0-255). RSSI is an arbitrary integer value, defined in the 802.11 standard and intended for use, internally, by the microcode on the adapter and by the device driver.

In general no vendors choose all the 256 different values to measure the value of the signal strength. Thus each vendor's 802.11 NIC has a specific maximum RSSI value ("RSSI_Max") which defines the maximum signal strength that the NIC can receive. The percentage represents the RSSI for a particular packet divided by the RSSI_Max value, multiplied by 100 to derive a percentage. The percentage signal strength metric is used to circumvent the complexities and potential inaccuracies of using RSSI as a basis for reporting dBm signal strength.

Experimental results show that there is not much change in the dBm values above roughly 5 mW. The range of energy that is typically measured begins at or below -10 dBm which is relatively a very weak signal and for values above that the dBm varies very less. Thus RSSI is used for most of the signal strength measures in general. We use the RSSI for most of the experimental results in the thesis work.

4.3 EXPERIMENTS

An ad hoc network is constructed and the propagation of the RF signals are studied in reference to the strengths of the signals. The distance between the ad hoc nodes in the network is varied and the variation of signal strength with distance is recorded and analyzed.

The experiments are conducted in minimum human interference scenario. These experiments are conducted in the Stilwell Hall building, Cleveland State University. A node is placed at a fixed position and another node is moved to different locations. The fixed node is referred to as a *landmark* for all experimental purposes and the mobile node is referred to as the *random node*. The experiments are majorly classified as "experiments

on the same floor", conducted on floor 3 of Stillwell Hall building and the "multiple floor experiments" that are conducted on different floors. Considering a 3 dimensional coordinate system for the experimental field of study, the *X and Y* coordinates are defined on a single floor and the floors are defined on the *Z* coordinate axis. We start the experiments with multiple floor experiments and proceed to get a more discrete analysis of the signal strength behavior by concentrating on a single floor.

4.3.1 **Multiple floor experiments**

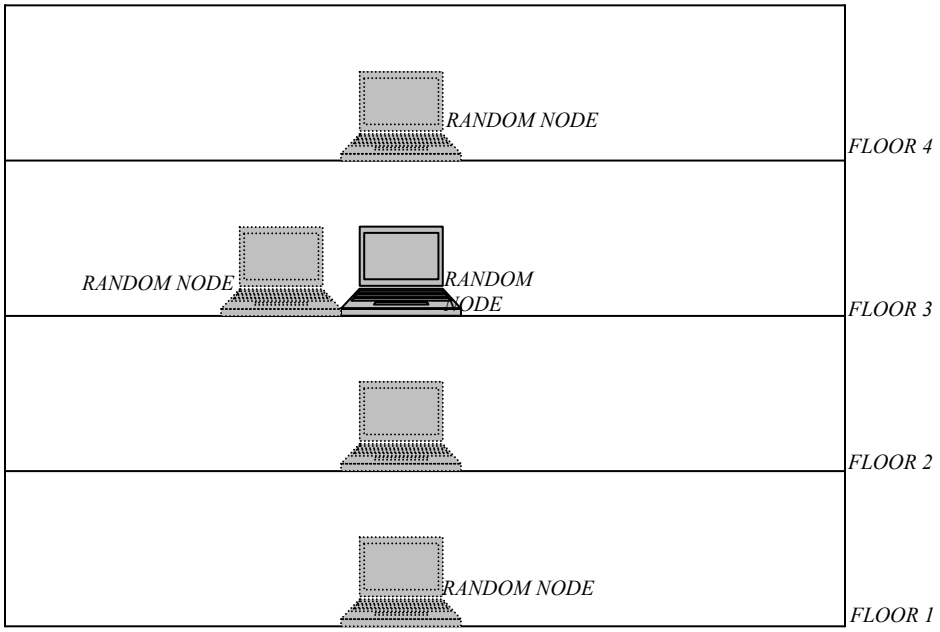


Figure 4.2: Experimental procedure for multiple floor signal strength experiments

The experimental procedure of the multiple floor experiments is shown in the figure 4.2. The landmark is placed on the third floor and the signal strengths are collected by the random node by moving it along all the floors.

Since the signal strengths vary due to many factors such as change in orientation of the antenna; human interferences; radio interferences etc. we collect the signal strengths many times at constant intervals and record the average value. For all the

experimental purposes we record the average value to minimize environmental variations.

The exact location of the landmark is shown in the figure below with the “X” mark in front of the Mobile Computing Research Lab (SH 305). The entire experiment is replicated by transferring the landmark on all the floors

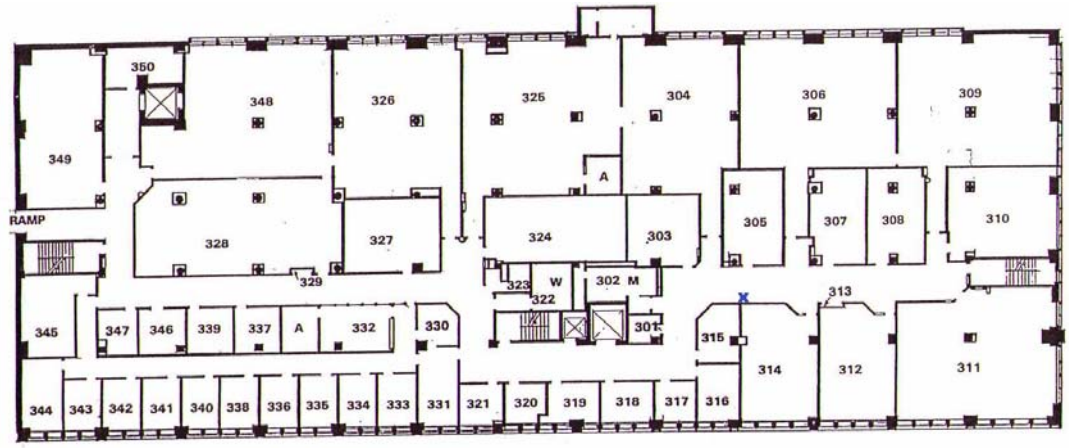


Figure 4.3: Multiple floor experiments- Location of landmark on floor 3

Considering the coordinate system analogy the landmarks and random nodes have the same *(x, y)* coordinates but the signal strengths are recorded at different *z* coordinates.

The experiments are replicated by placing the landmarks on different floors i.e. different *z* but same *(x, y)* coordinates. The resulting values are shown in the table below and the graph shows the pattern of the signal strengths.

Examining the graph in figure 4.4 for landmark 3 we can see that the signal strengths recorded at floor 2 and floor 4 are same. One more important noticeable fact from the graph is that as the random node moves farther from the landmark the signal strength (RSSI value) decreases drastically.

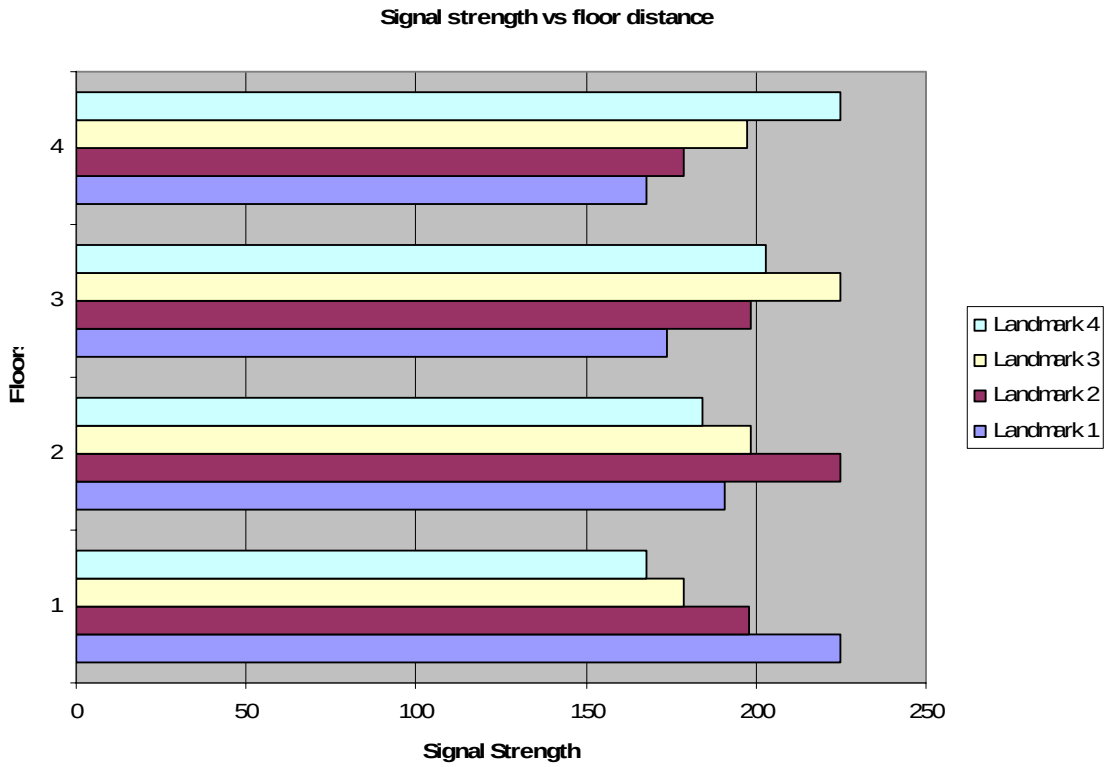


Figure 4.4: Signal strength vs Floor Distance

4.3.2 Single Floor experiments

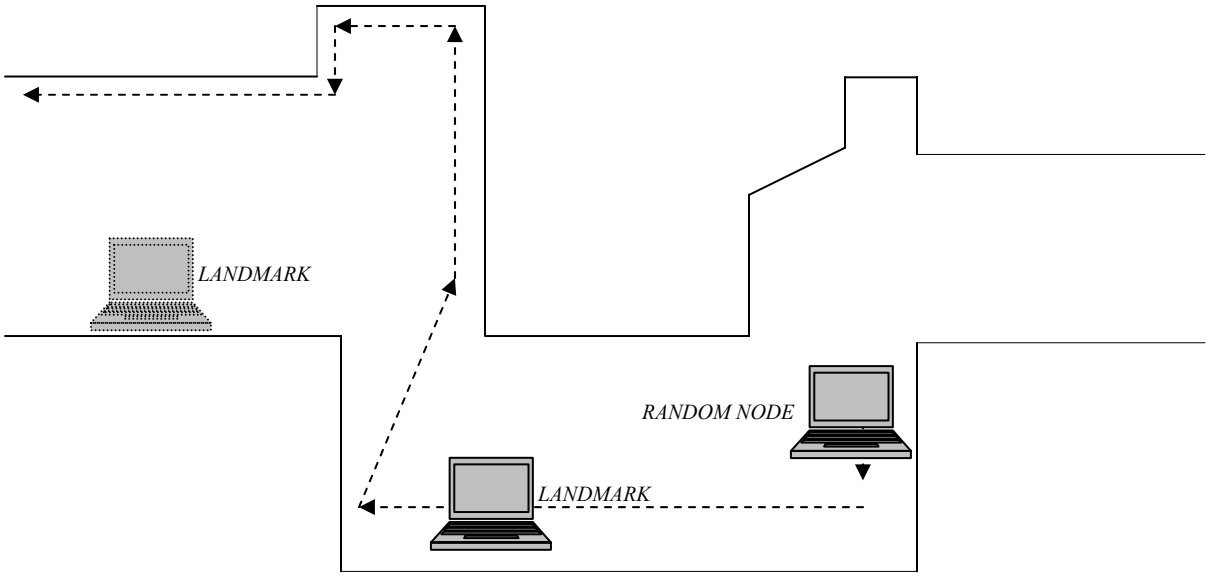


Figure 4.5: Experimental procedure for Single Floor signal strength experiments

The single floor experiments are conducted by choosing two different locations for the landmarks and the corresponding signal strengths are recorded at different locations on floor 3. The experimental procedure is shown in the figure 4.5. These experiments are conducted by placing one ad hoc node (Laptop 1) as the landmark in a single location. The other ad hoc node (Laptop 2) is taken to various locations on the same 3rd floor

- Landmark at room SH320

The first set of experiments are conducted by placing one ad hoc node at room SH320 and moving the other ad hoc node to other locations on floor three of the Stilwell Hall . The exact locations are shown in Figure 4.5. Signal strengths are collected at all the locations and recorded. Different signal strength ranges are shaded in different ways in the figure 4.5. We can see that signal strengths are maximum around the landmark and as the random node moves away from the landmark the signal strength decreases.

Figure 4.6: Single floor experiments with Landmark at SH320

- Landmark at room SH337

The second set of experiments is conducted similar to the above experiments but the location of landmark in these experiments is at SH 337. The exact positional coordinates of the other locations are shown in Figure 4.6. Signal strengths are collected at all the locations and recorded. Different signal strength ranges are shaded in different ways in the figure 4.5. We can see that signal strengths are maximum around the landmark and as the random node moves away from the landmark the signal strength decreases.

Figure 4.7: Single floor experiments with Landmark at SH320

4.4 CONCLUSION OF SIGNAL STRENGTH EXPERIMENTS

We observe that though the Newton's Third law which describes the relationship between signal strength and distance is defined for vacuum space we observe a definite

trend in the signal strength with distance, with an additional factor of the building architecture. In the multiple floor experiments the signal strength shows a very consistent behavior in signal strength variation. Similar behavior i.e. decrease of signal strength with variation of distance are shown even in the single floor experiments Thus a relation between signal strength and distance can be built locally for the experimental environment. This relation is used in the next chapter to build a positioning system.

Chapter 5

AD HOC POSITIONING SYSTEM

The Chapter 4 discusses the signal strength experiments and the variation of signal strength with distance. We use the experimental results obtained from study of signal strength behavior to build a positioning system. This chapter starts with a brief revisit of positioning methodologies. In section 5.2 we describe the algorithm we employ to locate a random node. The section 5.3 describes the experimental scenario and positioning methodology. We conclude the chapter with the error analysis and experimental conclusions in section 5.4.

5.1 INTRODUCTION

We have seen in the earlier chapters that a lot of researchers are experimenting on the positioning problem. Experiments have been performed on the IR based positioning systems, Global positioning system and RF based positioning systems. The thesis work focuses on RF based positioning system. We use the trends of the RF signals with varying positional coordinates to build the positioning system. We try to build a map between the positional coordinate and the signal strength variation in the context of the experimental environment. This map can be used for all purposes and helps in deriving the positional coordinates from the signal strength information.

5.2 AD HOC POSITIONING ALGORITHM

The Ad Hoc Positioning algorithm is built to determine the positional coordinates of a random point based on signal strength information collected from landmarks. The algorithm is based on the theoretical background and experimental work discussed in the earlier chapters. This Ad hoc Positioning system is primarily derived from the *distance based ranging technique* discussed in Section 3.1 and the *Truncated value singular value decomposition technique* discussed in Section 3.2.3

Step 1: *Read the positional coordinates,* (x_i, y_i) *where* $i = 1, 2, ...n$ *of the n physical landmarks from the table to create the Coordinate matrix - C*

Step 2: *Read the signal strengths* s_{ij} *where* $i = 1, 2, ...n;\ j = 1, 2, ...m$ *from m mobile landmarks at n physical landmarks at the corresponding locations from the table to generate the Signal Strength matrix: S*

Step 3: *Generate the Distance matrix D from the positional coordinates (Step 1) using the formula* $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ *where* d_{ij} *is the geographic distance measured from the ith node to jth node and* $d_{ii} = 0$.

Step 4: *Generate the proximity matrix P from the signal strengths (Step 2) using the formula* $p_{ij} = \sqrt{\sum_{k=1}^{m} (S_{ik} - S_{jk})^2}$ *where* p_{ij} *is the signal strength difference measured from the ith node to jth node and* $p_{ii} = 0$

Step 5: *Calculate the Proximity Distance Map using the formula* $T = DP^T (PP^T)^{-1}$

Step 6: *Read the signal strengths at the random point* $s_R = [s_{R1}, s_{R2}, s_{R3}, \cdots, s_{Rm}]$, *from the m mobile landmarks.*

Step 7: *Generate the proximity vector* $p_R = \left[p_{r1}, ..., p_{rN}\right]$ *from the random location to all the n physical landmarks using the equation* $p_{rj} = \sqrt{\sum_{k=1}^{m} (s_{rk} - s_{jk})^2}$ *where* $j = 1$ to n

Step 8: *Generate the distance vector* $d_R = \left[d_{r1}, ..., d_{rN}\right]$ *from the random location to all the physical landmarks using the Proximity Distance map i.e.* $d_R = Tp_R$

***Step 9**: Generate the Coordinate Distance Map (CDM) L using the formula*

$$L = 2\left[\begin{bmatrix} x_1 & y_1 \\ x_1 & y_1 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_1 & y_1 \end{bmatrix} - \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}\right]$$

***Step10**: Generate the constant matrix K using the formula*

$$K = \begin{bmatrix} k_2^2 \\ k_3^2 \\ k_4^2 \\ \vdots \\ k_n^2 \end{bmatrix} where\ k_i^2 = d_{ri}^2 - d_{r1}^2 - x_i^2 - y_i^2 + x_1^2 + y_1^2$$

***Step 11**: Generate the positional coordinates of the random point Xr using the equation*

$$X_R = (L^T L)^{-1} L^T K$$

Table 5.1: Ad hoc positioning algorithm

The detailed explanation of the algorithm is given in this section in three major steps Data Tabulation, Data Processing and finally Location Determination

- Data Tabulation

A coordinate system is constructed and mapped into the experimental area. The positional coordinates of all the physical landmarks are recorded in the coordinate matrix C.

$$C = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}_{nX2}$$

Signal strength values are measured from mobile nodes whose positional coordinates are known and these mobile nodes are called *mobile landmarks*. The signal strength values are measured from the mobile landmarks at various locations called *physical landmarks*. For *m* mobile landmarks, the signal strength values at each physical landmark are represented by the signal strength vector s_i.

$$s_i = [s_{i1}, s_{i2}, s_{i3}, \cdots, s_{im}]$$

where s_{ij} is the signal strength measured at the ith physical landmark from the jth mobile landmark. For *n* physical landmarks the overall signal strength matrix can be represented by the *n X m* matrix *S.*

$$S = \begin{pmatrix} S_{11} & S_{12} & S_{13} & \cdots & S_{1m} \\ S_{21} & S_{22} & S_{23} & \cdots & S_{2m} \\ S_{31} & S_{32} & S_{33} & \cdots & S_{3m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{n1} & S_{n2} & S_{n3} & \cdots & S_{nm} \end{pmatrix}_{nXm}$$

The signal strengths measured are directly related to the positional coordinates of the physical landmarks.

- Data Processing

The signal strengths are related to the positional coordinates. The relation can be mapped using the distance and proximity matrix. The above data is processed to obtain distance and proximity matrices. The distance of any physical landmark from any other physical landmark is

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

where d_{ij} is the geographic distance measured from the ith node to jth node and $d_{ii} = 0$. Given N physical landmarks the distance vector at any of the physical landmark can be represented by the vector

$$d_i = [d_{i1}, d_{i2}, d_{i3}, \cdots, d_{in}]$$

The distance matrix D is represented by

$$D = \begin{pmatrix} d_{11} = 0 & d_{21} & d_{31} & \cdots & d_{n1} \\ d_{12} & d_{22} = 0 & d_{32} & \cdots & d_{n2} \\ d_{13} & d_{23} & d_{33} = 0 & \cdots & d_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{1n} & d_{2n} & d_{3n} & \cdots & d_{nn} = 0 \end{pmatrix}_{nXn} \quad where\ d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The entries in the S matrix are used to determine a signal strength difference matrix also called the proximity matrix P. Each entry in the P matrix can be determined by using the equation given below.

$$p_{ij} = \sqrt{\sum_{k=1}^{m} (S_{ik} - S_{jk})^2}$$

where p_{ij} is the signal strength difference measured from the ith node to jth node and $p_{ii} = 0$. Given N physical landmarks the proximity vector at any of the physical landmark can be represented by the vector

$$p_i = \left[p_{i1}, \ldots\ldots, p_{iN} \right]^T$$

The proximity matrix P can be represented as

$$P = \begin{pmatrix} p_{11}=0 & p_{21} & p_{31} & \cdots & p_{n1} \\ p_{12} & p_{22}=0 & p_{32} & \cdots & p_{n2} \\ p_{13} & p_{23} & p_{33}=0 & \cdots & p_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1n} & p_{2n} & p_{3n} & \cdots & p_{nn}=0 \end{pmatrix}_{nXn} \quad where\ p_{ij} = \sqrt{\sum_{k=1}^{m}(s_{ik}-s_{jk})^2}$$

The internodal distances are a functional difference of positional coordinates between the positions of various physical landmarks. Similarly the proximity matrix P is a functional difference between the signal strengths values recorded at various physical landmarks. Since the signal strengths are a function of the positional coordinates we try to build an optimal linear transformation T, called the *proximity distance* map (PDM) [34] that gives a mapping from the proximity matrix *P* to the distance matrix *D*. This in turn helps to obtain the relation between the signal strengths and positional coordinates. Note that T is an *N* –by- *N* square matrix. Each row vector t_i of T can be obtained by minimizing the following square error:

$$e_i = \sum_{k=1}^{M}(d_{ik} - t_i p_k)^2$$

$$= \left\| d_i^T - t_i P \right\|^2$$

The least-square solution for the row vector t_i is

$$t_i = d_i^T P^T (PP^T)^{-1}$$

As a result, the PDM is defined as

$$T = DP^T (PP^T)^{-1}$$

- Location Determination

Assuming the coordinates of the random point R whose location needs to be determined has the positional coordinates (x_R, y_R) and the signal strength values recorded at that point are represented by the vector

$$s_R = [s_{R1}, s_{R2}, s_{R3}, \cdots, s_{Rm}]$$

Using the s_R vector a proximity vector which gives the signal strength difference from all the s_i is determined. The proximities of the random point from all the physical landmarks are determined and recorded as the vector

$$p_R = \left[p_{r1}, \ldots\ldots, p_{rN}\right]$$

We determine the distances of the random point from all the physical landmarks using the Proximity distance map (PDM), T. The distance vector of the the random point R can be determined by substituting p_R in the PDM equation.

$$d_R = Tp_R$$

then we obtain the vector

$$d_R = \left[d_{r1}, \ldots\ldots, d_{rN}\right]$$

Given coordinates of all the physical landmarks in the matrix C and the respective distances to them in the vector d_R, we build the nonlinear system

$$(x_R - x_i)^2 + (y_R - y_i)^2 = d_{ri}{}^2 \text{ where } i=1,2,\ldots\ldots n \qquad \text{---------- } (i)$$

In Global Positioning system the system is solved using nonlinear methods based on successive approximations, but it also can be solved by reduction to a linear system by subtracting one equation from the rest [37]

Expanding equation (*i*)

$$
\begin{aligned}
&(x_R - x_1)^2 + (y_R - y_1)^2 = {d_{r1}}^2 ------(1)\\
&(x_R - x_1)^2 + (y_R - y_1)^2 = {d_{r1}}^2 ------(2)\\
&(x_R - x_1)^2 + (y_R - y_1)^2 = {d_{r1}}^2 ------(3)\\
&\vdots\\
&(x_R - x_n)^2 + (y_R - y_n)^2 = {d_{rn}}^2 ------(n)
\end{aligned}
$$

Subtracting (1) from all equations (2) to (n) we get

$$
\begin{aligned}
&(2x_1x_R - 2x_2x_R) + (2y_1y_R - 2y_2y_R) = d_{r2}^2 - d_{r1}^2 - x_2^2 - y_2^2 + x_1^2 + y_1^2\\
&(2x_1x_R - 2x_3x_R) + (2y_1y_R - 2y_3y_R) = d_{r3}^2 - d_{r1}^2 - x_3^2 - y_3^2 + x_1^2 + y_1^2\\
&\vdots\\
&(2x_1x_R - 2x_nx_R) + (2y_1y_R - 2y_ny_R) = d_{rn}^2 - d_{r1}^2 - x_n^2 - y_n^2 + x_1^2 + y_1^2
\end{aligned}
$$

Taking a general equation and solving

$$
\begin{aligned}
&2x_R(x_1 - x_i) + 2y_R(y_1 - y_i) = d_{ri}^2 - d_{r1}^2 - x_i^2 - y_i^2 + x_1^2 + y_1^2 \text{ where } i = 2,3,....n\\
&2\begin{bmatrix}(x_1 - x_i) & (y_1 - y_i)\end{bmatrix}\begin{bmatrix}x_R\\ y_R\end{bmatrix} = k_i^2 \text{ where } k_i^2 = d_{ri}^2 - d_{r1}^2 - x_i^2 - y_i^2 + x_1^2 + y_1^2\\
&2\left[\begin{bmatrix}x_1 & y_1\end{bmatrix} - \begin{bmatrix}x_i & y_i\end{bmatrix}\right]\begin{bmatrix}x_R\\ y_R\end{bmatrix} = k_i^2
\end{aligned}
$$

Expanding the above equation to substitute all values of i

$$
\left[\begin{bmatrix}x_1 & y_1\\ x_1 & y_1\\ x_1 & y_1\\ \vdots & \vdots\\ x_1 & y_1\end{bmatrix} - \begin{bmatrix}x_2 & y_2\\ x_3 & y_3\\ x_4 & y_4\\ \vdots & \vdots\\ x_n & y_n\end{bmatrix}\right]\begin{bmatrix}x_R\\ y_R\end{bmatrix} = \begin{bmatrix}k_2^2\\ k_3^2\\ k_4^2\\ \vdots\\ k_n^2\end{bmatrix}
$$

where

$$
L = 2\left[\begin{bmatrix}x_1 & y_1\\ x_1 & y_1\\ x_1 & y_1\\ \vdots & \vdots\\ x_1 & y_1\end{bmatrix} - \begin{bmatrix}x_2 & y_2\\ x_3 & y_3\\ x_4 & y_4\\ \vdots & \vdots\\ x_n & y_n\end{bmatrix}\right], X_R = \begin{bmatrix}x_R\\ y_R\end{bmatrix} \text{ and } K = \begin{bmatrix}k_2^2\\ k_3^2\\ k_4^2\\ \vdots\\ k_n^2\end{bmatrix}
$$

Let us term the L matrix as the Coordinate Distance map (CDM). The inverse of L cannot be calculated so we multiply both the sides with L^T to get a square matrix

$$L^T L X_R = L^T K$$

We multiply both the sides of the above equation with the inverse of $L^T L$

$$(L^T L)^{-1} L^T L X_R = (L^T L)^{-1} L^T K$$
$$I X_R = (L^T L)^{-1} L^T K$$
$$X_R = (L^T L)^{-1} L^T K$$

5.3 POSITIONING EXPERIMENTS

The Ad hoc positioning system in implemented in a real environment and the effectiveness of the algorithm is determined by analysis of the experimental results The following experiments are performed to determine the positional coordinates of a random point R i.e (X_R,Y_R) by studying the behavior of signal strength experiments in the experimental environment.

The experimental test bed is located on the third floor of the 4-storeyed Stilwell Hall building at Cleveland State University. The layout of the floor is shown in Fig. 5.1.

In the present experiment we have $m = 3$, mobile landmarks A, B and C. The mobile landmarks used for the experimental purposes are two 3670 model iPAQs from Compaq running on Familiar Linux kernel (A and C) and a Toshiba Satellite Laptop (with Intel Pentium III Processor) (B) running on Red Hat Linux operating systems version 9 , as landmarks. The location of the landmarks is shown in the figure 5.1. One more Toshiba Satellite Laptop running on Red Hat Linux 9 kernel is used as the random node (R) to collect signal strength stamps at fourteen physical landmarks in the experimental area as mentioned in the signal strength experiments. All the nodes i.e. the 3

Landmarks and the random node are equipped with a Lucent Orinoco Wireless Card. The experiments are conducted in ad hoc mode using AODV – UU routing protocol. The 2.4 GHz license free ISM band is used as the RF communication medium and the data transmissions are carried out at a fixed data rate of 2Mbps.

A coordinate system is built in the experimental area and the positional coordinates of the locations of the landmarks and table entry points. These positional coordinates are entered in the data table corresponding to each location. The positional coordinates are stored in a Coordinate matrix *C*.

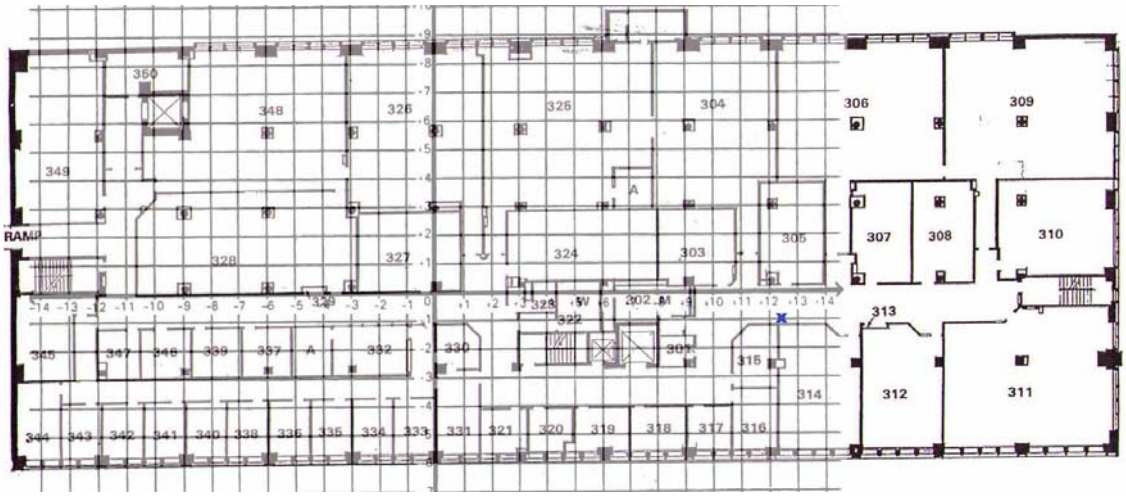


Figure 5.1: Construction of the coordinate system

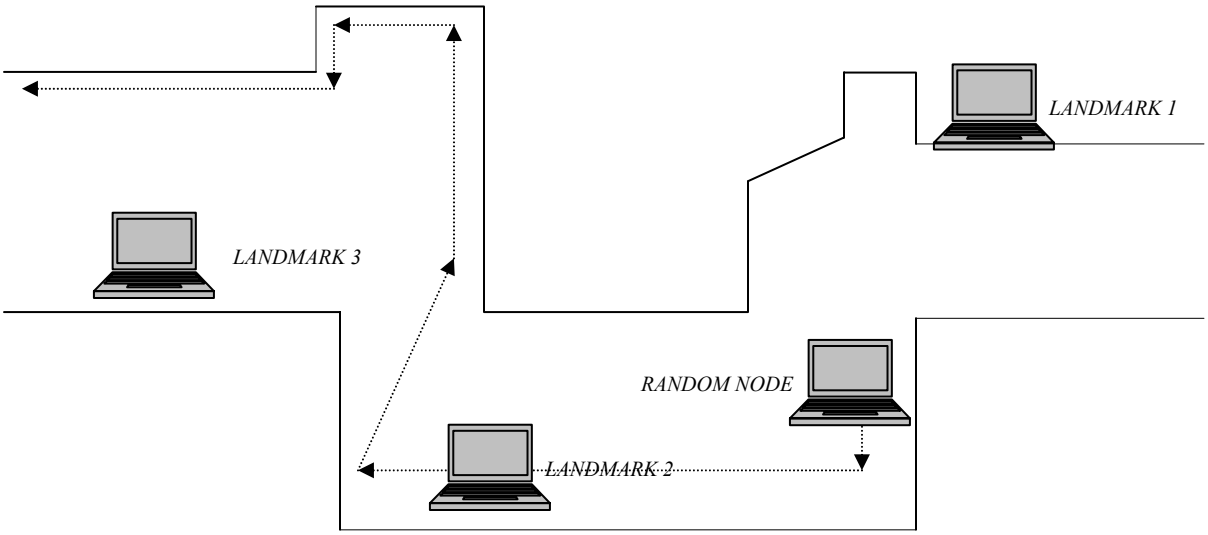


Figure 5.2: Experimental Procedure for the positioning experiments

The experimental procedure is shown in Figure 5.2. As shown in the figure the mobile landmarks are placed at fixed locations and the Laptop 2 is moved along the dotted line to fourteen different physical landmarks. We record the signal strengths from the mobile landmarks at all the fourteen locations as we do in the experiments in Chapter 4. The positional coordinates of the mobile landmarks and the physical landmarks are shown in Figure 5.3 and Figure 5.4 respectively. The experimental data is tabulated and processed for location determination purposes

- Data Tabulation

Collection of signal strengths at 14 points from the three landmarks (S_{iA}, S_{iB}, S_{iC}) where

$i = 1, 2, 3 \ldots 14$. in a signal strength matrix **S.**

$$S = \begin{pmatrix} S_{1A} & S_{1B} & S_{1C} \\ S_{2A} & S_{2B} & S_{2C} \\ S_{3A} & S_{3B} & S_{3C} \\ \vdots & \vdots & \vdots \\ S_{14A} & S_{14B} & S_{14C} \end{pmatrix}_{14X14}$$

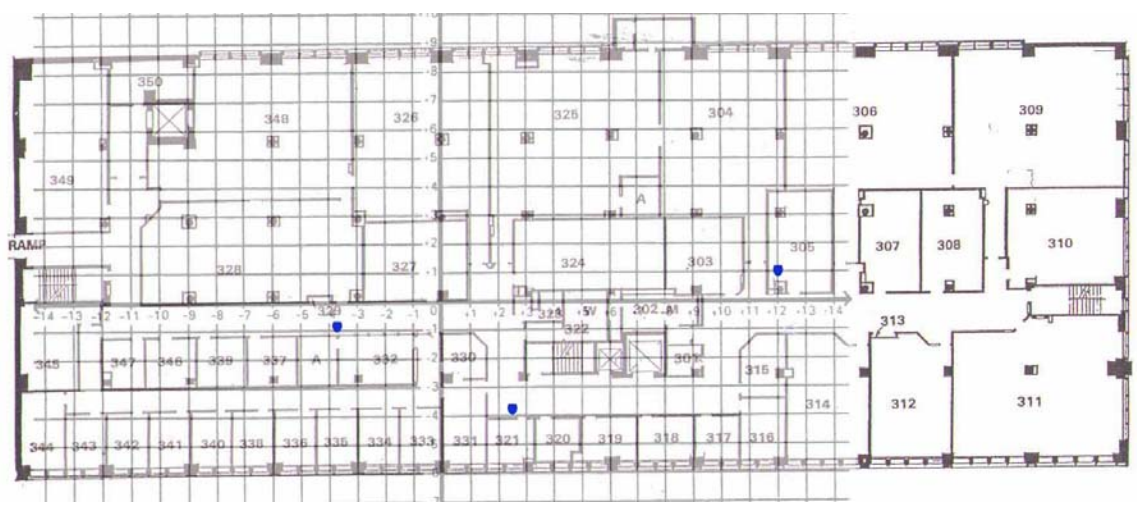


FIGURE 5.3: Positional coordinates of the mobile landmarks

The coordinates $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_{14}, y_{14})\}$ of all the fourteen landmarks are collected and recorded in a Coordinate matrix C

$$C = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}_{nX2}$$

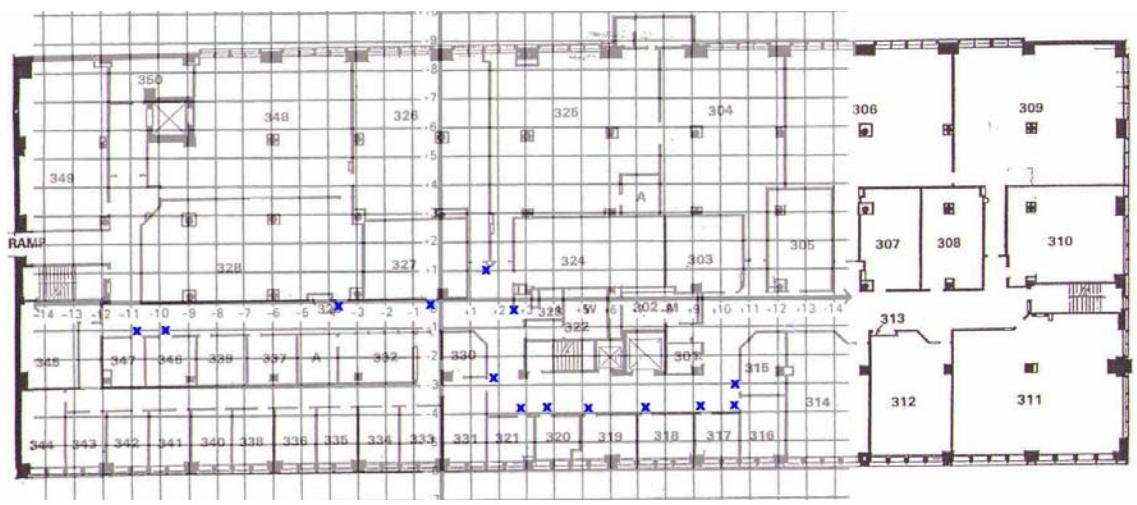


FIGURE 5.4: Positional Coordinates of the Physical Landmarks

Signal strengths from the three landmarks collected by the random node at a location *R* whose position is not known are (S_{RA}, S_{RB}, S_{RC}).

The collected data is processed into a form on which mathematical calculations can be performed.

- Data Processing

We process the collected data i.e. the collected signal strength information to obtain a proximity matrix.

We use the table entries in the *S* matrix to determine a signal strength difference matrix *P*. Given N experimental points the coordinate of any point p_i in an *nXn* matrix *P* is

$$p_i = \left[p_{i1}, \ldots\ldots, p_{iN} \right]^T$$

where p_{ij} is the Euclidean distance between signal strengths measured from the *i*th node to *j*th node i.e.

$$p_{ij} = \sqrt{(S_{iA} - S_{jA})^2 + (S_{iB} - S_{jB})^2 + (S_{iC} - S_{jC})^2} \text{ and } p_{ii}=0.$$

Finally the proximity matrix is represented as

$$P = \begin{pmatrix} p_{11} = 0 & p_{21} & p_{31} & \cdots & p_{n1} \\ p_{12} & p_{22} = 0 & p_{32} & \cdots & p_{n2} \\ p_{13} & p_{23} & p_{33} = 0 & \cdots & p_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1n} & p_{2n} & p_{3n} & \cdots & p_{nn} = 0 \end{pmatrix}_{nXn} \quad where\ p_{ij} = \sqrt{\sum_{k=1}^{m} (s_{ik} - s_{jk})^2}$$

Geographic distances from one virtual landmark to another are calculated and tabulated in an *(n*n)* matrix *D,* using the positional coordinates.

The coordinate of any point s_i in the *D* matrix is

$$d_i = \left[d_{i1}, \ldots\ldots, d_{iN} \right]^T$$

where d_{ij} is the geographic distance measured from the *i*th node to *j*th node i.e. $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and d_{ii}=0.

Finally the distance matrix is represented as

$$D = \begin{pmatrix} d_{11} = 0 & d_{21} & d_{31} & \cdots & d_{n1} \\ d_{12} & d_{22} = 0 & d_{32} & \cdots & d_{n2} \\ d_{13} & d_{23} & d_{33} = 0 & \cdots & d_{n3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{1n} & d_{2n} & d_{3n} & \cdots & d_{nn} = 0 \end{pmatrix}_{nXn} \quad where\ d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Proximity distance map and the position coordinate determination: Using the equation for the PDM described in the APS algorithm we calculate the PDM for the present experiment

$$T = DP^T (PP^T)^{-1}$$

- Location Determination

The random node i.e. second Toshiba Satellite Laptop, is placed at a location whose positional coordinates (x_R, y_R) are not known. We use the values obtained from the experiment and the Ad hoc positioning algorithm to determine these positional coordinates (x_R, y_R).

Using the information collected in the s_R vector we obtain the proximity vector p_R, i.e. the proximity of the random point from all the virtual landmarks.

$$p_R = [p_{r1}, ..., p_{r14}]$$

The distance vector d_R, i.e. the distance of the random point from all the virtual landmarks is determined using the Proximity Distance Map (PDM), T

$$d_R = Tp_R$$

This gives the distances

$$d_R = [d_{r1}, ..., d_{r14}]$$

Using the equation from the algorithm and the coordinates from matrix C, and the values from d_R we calculate the matrices L, K and finally using the equation

$$X_R = (L^T L)^{-1} L^T K$$

Thus we determine the coordinates (x_R, y_R) i.e. the positional coordinates of the random point.

5.4 ERROR ANALYSIS AND CONCLUSION

Using the above Ad Hoc Positioning system we determine the positional coordinates i.e. (x_R, y_R) of many locations on the third floor of the Stillwell Hall building at Cleveland State University. However the values obtained from the system are subject to some errors. The performance of the system is affected by environmental disturbances and mathematical inaccuracies. The error in the location determined is calculated by measuring the Euclidean distance of the determined value from the original value. The analysis of these errors in different scenarios and locations helps us arrive at some conclusions.

The experiments are conducted at fourteen locations and the errors are analyzed at all the locations. The experimental conditions are varied by changing the number of landmarks. When the number of landmarks is at five we can see that the error mounts as close as to 25cms. As the number of landmarks increases the error value drops drastically first and gradually later. We can see that the error performance is best when the number of landmarks is the maximum i.e. fourteen. Thus we can conclude that the performance of the positioning system can be improved by increasing the number of physical landmarks.

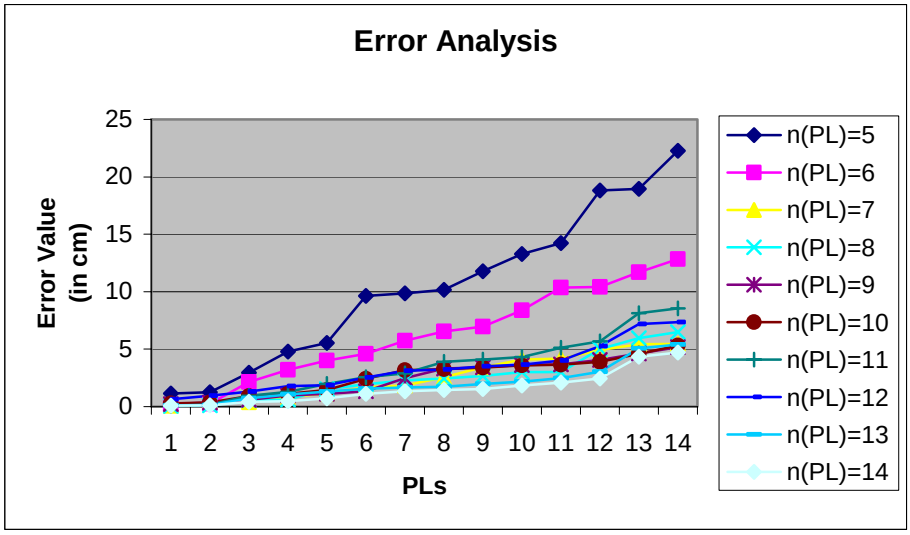
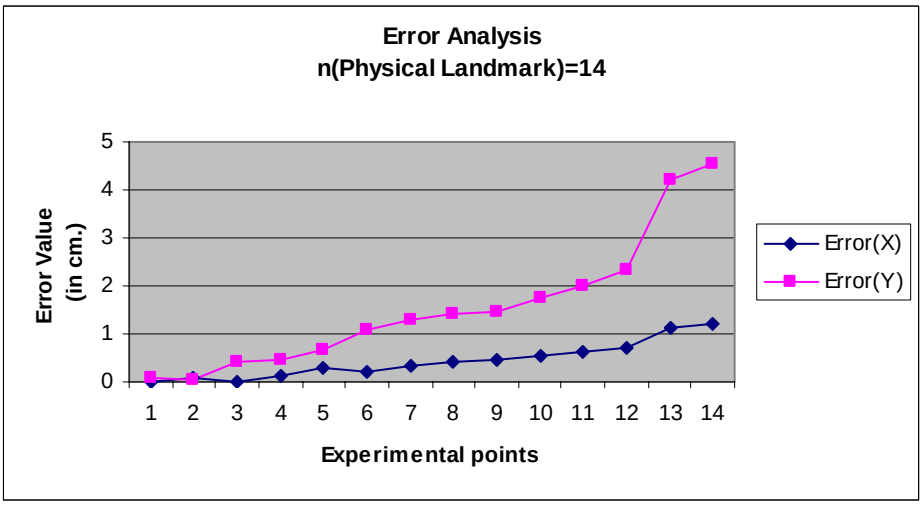


Figure 5.4: Error analysis for varying number of landmarks

The error analysis of the X coordinate and the Y coordinate has given some interesting results. We can see that the error performance along the X axis is much better than that along the Y axis. By observing the locations of the physical landmarks we can see that most of the physical landmarks are parallel to the axis i.e. the most of the physical landmarks have the same Y coordinate. Thus the landmarks give better results when they are not in a straight line.



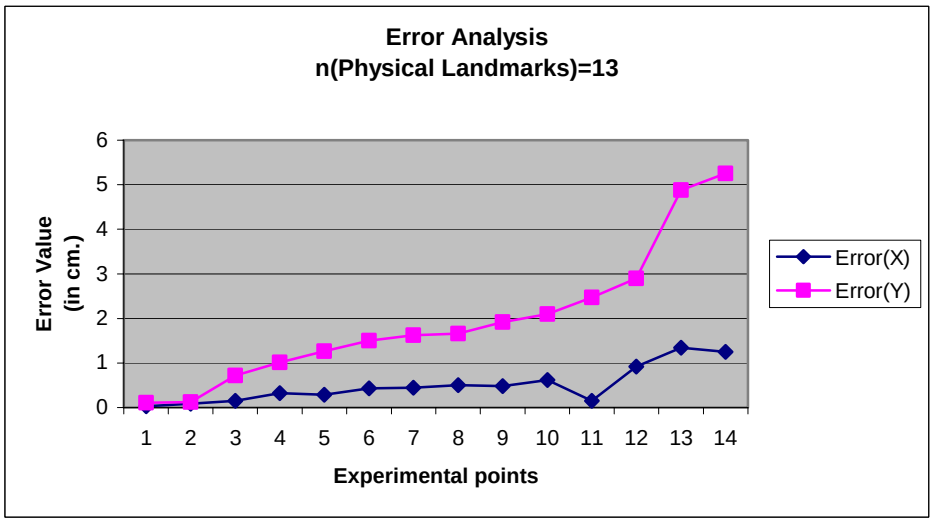
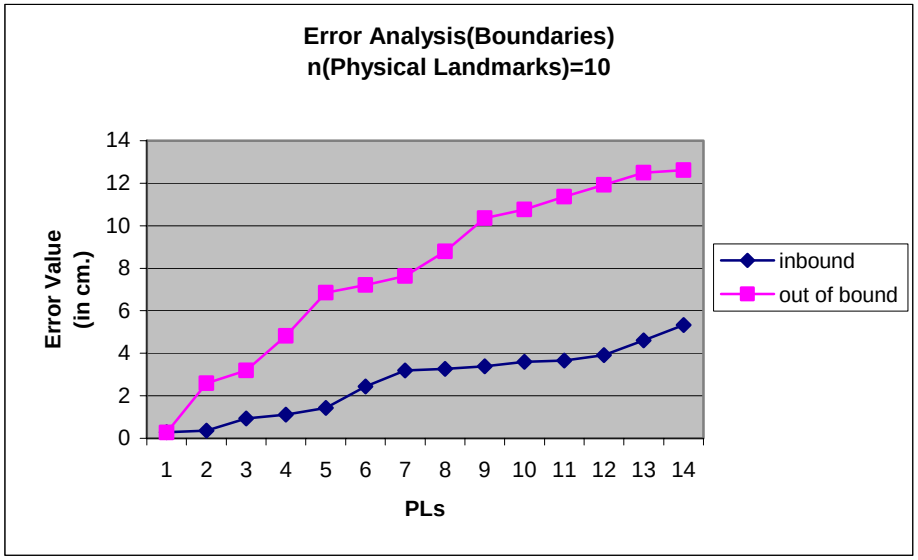


Figure 5.5: Error Analysis for errors on X and Y axis

We vary the experimental conditions again by choosing the area covered by all the landmarks. We evaluate the cases when all the landmarks are very close to each other and cover a smaller area in comparison to landmarks that are far from each other and cover a wider area. We can see that when the landmarks cover a wider area the error performance of the Positioning system in much better in both the cases.



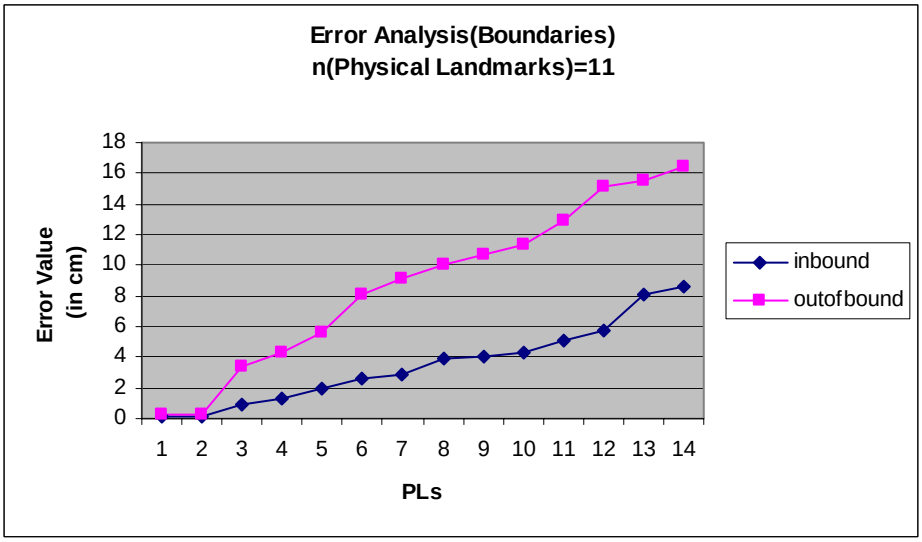


Figure 5.6: Error Analysis with different areas covered by the landmarks

From the experimental results we can conclude that to have better accuracy of positioning

- We need to have more number of physical landmarks
- We need to avoid placing landmarks in a straight line
- The landmarks should cover maximum area until the network has stable internodal communication.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The ability to derive accurate and reliable position estimates for nodes within an ad hoc wireless network is motivated by the intrinsic need for location information to accompany network data. Several methods and techniques for achieving these estimates have been described in detail.

The solution to the positioning problem was divided into two experimental phases, signal strength experimentation and location determination. In the first phase we choose an indoor environment and two mobile devices to create a wireless ad hoc link. The distance between the mobile devices is varied for different values and the signal strengths corresponding to different distances are recorded. These experiments are repeated for different scenarios and signal strength behavioral pattern is observed. Based

on this behavior of the signal strength we conduct the phase two i.e. location determination phase. Some mobile landmarks at fixed locations are used to establish physical landmarks in the experimental area in terms of signal strengths. To determine the positional coordinates the signal strengths are recorded at the random point, the signal strengths of the node at that point are recorded from all the mobile landmarks. A series of mathematical calculations are done to derive the positional coordinates. The impact of physical landmarks on the positioning accuracy has been discussed which helps in optimally choosing the physical landmarks for optimizing system performance.

Future developments can be made on the ad hoc positioning system

1. **Positioning system integration into the random node**

In the present implementation of the positioning system the random node only collects the signal strength information. This signal strength information is tabulated in another system and the positioning algorithm is implemented on a Windows Platform using Matlab. However the extra processing time of recording the information from the random node can be reduced if the positioning algorithm can be implemented in the random node itself. This integration makes it possible to directly return the positional coordinates. This integration makes it possible to use any node as a mobile landmark and accommodates better accuracy of the positioning system

2. **Positioning system updates**

We assume that the mobile nodes are fixed in the present positioning system. But when the mobile nodes move to newer locations the positioning system built at an earlier point of time becomes unsuitable for location determination purposes. The AODV-UU routing protocol can be used to solve this problem. A periodic beacon containing the

positional information of the landmarks can be transmitted at regular intervals using AODV routing protocol. The AODV routing protocol can be also used in a reactive manner i.e. an position update beacon can be routed to all the ad hoc nodes in the network whenever a mobile landmarks moves to a new location. The routing experiments discussed in Chapter 4 show that AODV-UU can be used for multi path routing. Thus AODV-UU in unison with Ad hoc positioning system can be used to build a time sensitive positioning system.

APPENDIX

The ad hoc networks for the experimental purposes are built using Toshiba Satellite Laptops and handheld devices, iPAQs from Compaq. All the mobile devices run on Linux Operating system and use AODV routing protocol. AODV is an on demand routing protocol used by MANETS. A version of AODV, “AODV-UU” that was implemented at the Uppsala University is used in the thesis work.

Chapter A1

AODV-UU INSTALLATION AND CONFIGURATION ON LAPTOP

Toshiba satellite laptops along with handheld devices are used as mobile nodes in all the experiments. The Toshiba laptops run on Red hat 9 Linux kernel and AODV-UU is installed on all of them. The installation procedure is discussed next in the chapter.

A1.1 AODV INSTALLATION

The open source code for AODV-UU is available at following URL for download

1) Download the aodv-uu.0.8.1.tar.gz file.
2) Untar the file using the command.

   tar zxvf aodv-uu.0.8.1.tar.gz.
3) The contents will be extracted into the folder aodv-uu0.8.1.
4) Enter the directory and type the following command *make install.*
5) Verify that the location of the header files is matched with the MAKEFILE.
6) Once the compilation is complete, kaodv.o and aodvd files are generated.

A1.2 WIFI CONFIGURATION

1) Open Network configuration from the startup menu > system settings > network or use the command “neat” in a new terminal.

   # neat

   This takes us to the “Network Configuration” dialog box.

2) In the network configuration dialog box choose the sub dialog box for “Devices” and from the menu bar choose “New”. This takes us to an “Add new Device Type” dialog box.

3) Choose the device type as “Wireless Connection” and click *“forward”*. This takes us to a “Select Wireless Device” dialog box.

4) We get a dialog box for selection of a wireless device if the WiFi card is already present in the dialog box chooses the device else choose “Other wireless card” and perform the following steps.

   - You get a “Select Ethernet Adapter” dialog box
   - Select the *Adapter* (Wifi card) from a drop down menu bar.
   - Choose the *Device* (Ethernet interface) i.e. eth1 or eth0
   - Click forward

This takes us to the Configure Wireless Connection

5) Choose the

   - *Mode* as “Ad-Hoc” Mode.

- Choose a common *Network Name (SSID)* for all ad hoc devices you choose to network by choosing Specified and type in a common network name (we have chosen the name as “aodvd”).
- Choose a common “Channel” between the numbers 1 to 14
- Choose a transmission rate from the options (11 Mbps)
- Click forward

This takes us to a “Configure Network Settings “

6) Choose the dhcp for Internet configuration. Our experimental values are (192.168.10.10, 192.168.10.20, 192.168.10.30, 192.168.10.40, 192.168.10.50)

   This takes us to a Create Wireless Device dialog box which is nothing but a confirmation dialog box.

7) We need to flush out the firewalls on all the nodes for the communication to take place. This is done using the command *“# iptables –F ”*.

Chapter A2

INSTALLATION OF FAMILIAR LINUX 0.7.2 ON IPAQ 3670

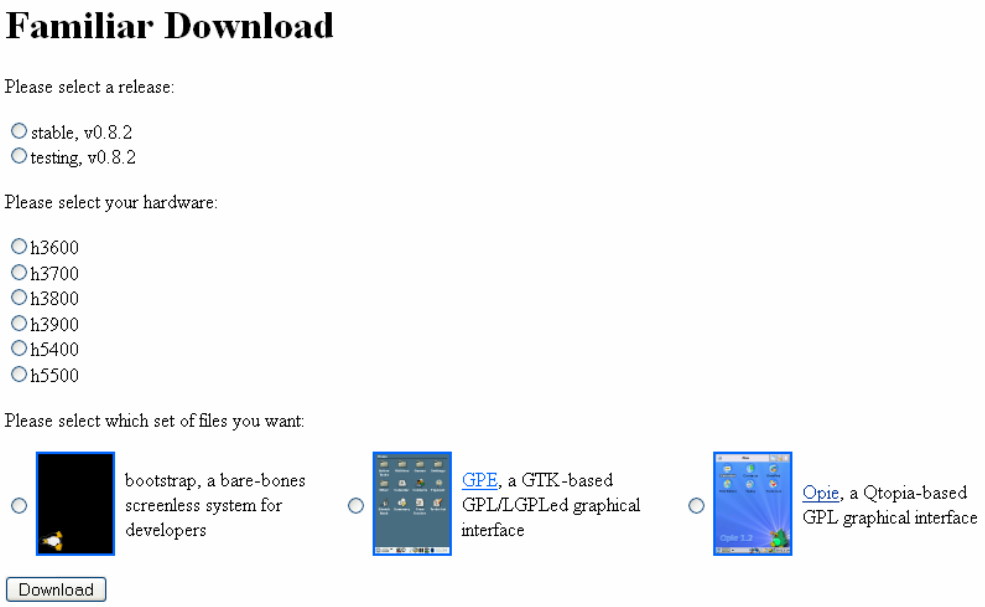
The iPAQ 3670 is an old version of the iPAQ and was manufactured by Compaq Systems. The default operating system installed in iPAQ 3670 is Microsoft Pocket PC 2003 premium. The iPAQ comes with a dual USB/Serial cable which is used for the setting communication from the iPAQ to a Host PC

The Installation process can be described in 3 major steps

1. Downloading the bootblaster and Linux bootloader
2. Install bootloader on the iPAQ
3. Load the Linux image from the bootblaster downloaded on the host PC using serial communication

A2.1 DOWNLOAD FAMILIAR SOURCE.

1) The navigation directions for downloading the Familiar distribution are www.handhelds.org > *Under Quick Links* Familiar distributions > *Under contents* Download > *Under Version description* choose version > *Under installation* choose Download and the following page appears



We choose the stable version for h3600 series and the Opie graphical interface

Due to the complexity of the website, the navigation directions must be useful, also because of constant updating and shuffling of the links the navigation map may not be useful for a long time.

2) Transfer BootBlaster and bootldr to your iPAQ using ActiveSync.

3) If ActiveSync isn't already installed on your Windows PC, install it by inserting the iPAQ Pocket PC Companion CD-ROM into the PC's drive and following the ensuing instructions.
4) Copy the BootBlaster and bootldr files that came with your Familiar Linux distribution to your Windows PC if they're not already there.
   - If you're installing on an H3900 iPAQ, the files are named BootBlaster3900-2.6.exe and bootldr-pxa-2.21.12.bin.
   - Otherwise, they're named BootBlaster_1.19.exe and bootldr-sa-2.21.12.bin.
5) Plug the iPAQ cradle into an AC power outlet.
6) Connect the USB connector from the cradle to the Windows PC.
7) Slide the iPAQ into its cradle. If a "Set Up a Partnership" screen appears on the PC, choose "No" and then click Next.
8) Copy BootBlaster_1.19.exe or BootBlaster3900-2.6.exe to the default folder on the iPAQ by clicking Explore in ActiveSync and dragging their icons there. Ignore any "may need to convert" messages.
9) Do the same thing for bootldr-sa-2.21.12.bin or bootldr-pxa-2.21.12.bin.
10) Save your PocketPC image for later restoration, if desired.
11) Execute "Flash -> Save Bootldr .gz Format" in BootBlaster to save the bootloader in file "\My Documents\saved_bootldr.gz" on the iPAQ.

Note that the Linux Bootloader will also boot PocketPC, so restoration of this file is not generally required. Right at the moment, there is a bug in the Linux Bootloader which causes PocketPC to reinitialize itself every few boots. You may indeed wish to keep and restore this bootloader if you restore PocketPC.

12) Execute "Flash -> Save Wince .gz Format" in BootBlaster to save the PocketPC image in file "\My Documents\wince_image.gz" on the iPAQ. This takes two to three minutes.

13) If no backup of Pocket PC is desired, you can skip this step entirely.

14) Note that this procedure saves your bootloader and Pocket PC executable image: it does not preserve any data you may have entered in your iPAQ under Pocket PC. So also synchronize your iPAQ to your host to preserve this data. Note that Familiar does not *yet* have any way to resynchronize this data to Linux (we hope/expect to have Linux<->Host synchronization in a near future release).

15) Copy saved_bootldr.gz and wince_image.gz to your Windows PC.

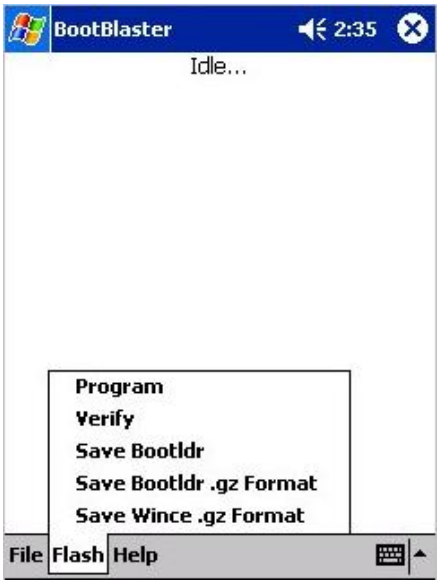
16) Select "View -> Refresh" in the ActiveSync Explore window on the PC. Icons for the saved_bootldr.gz and wince_image.gz files should appear.

17) Drag the saved_bootldr.gz and wince_image.gz icons from the ActiveSync Explore window to a local folder on your PC.

18) As with any backup files, please store saved_bootldr.gz and wince_image.gz in a safe place. We highly recommend verifying the built-in gzip checksum in both files before presuming your backup is safe (see the gzip man page for details).

A2.2 INSTALL THE BOOTLOADER.

1) Before continuing, be sure that the iPAQ is plugged into external power, and that the battery is charged, to protect against the small chance of power failure during the very limited period the iPAQ is reprogramming the bootloader flash. Do NOT touch the power button or reset button on your iPAQ until you have performed the "Verify" step below. From the "Flash" menu on BootBlaster, select "Program".



2) A file dialog will open allowing you to select the bootloader to use. Select bootldr.bin.gz, which may have a version number embedded in it. We use a gzip file because it has an internal checksum. *Wait patiently*. It takes about 15 seconds to program the bootloader. Do not interrupt this process, or the iPAQ may be left in an unusable state.

3) From the "Flash" menu on BootBlaster, select "Verify".

- If it does not say that you have a valid bootloader, do *NOT* reset your iPAQ, do *NOT* turn off your iPAQ.
- Instead, try programming the flash again.
- If that doesn't work, program your flash with your saved bootloader.
- If that doesn't work, send e-mail to bootldr@handhelds.org and/or get on the IRC and ask for help. Leave the iPAQ plugged in and do *NOT* reset it or turn it off.
- If everything has gone well, you have successfully installed the CRL bootldr program, which can run either Linux or PocketPC. As yet, your PocketPC image is intact and should restart normally; the next step actually installs Linux (overwriting Pocket PC).

A2.3 INSTALL FAMILIAR V0.7.2 WITH A SERIAL LINE

1) You will need a serial sync cable or serial sync cradle. The dual USB/Serial cradle that comes with the H3800 and H3900 will also work. You will need to use a terminal program such as minicom, kermit, or Hyperterminal.
2) Hold down the joypad and push the reset button on the iPAQ. You will need to remove it from the cradle to access the reset button.

- For non-H5xxx: When the bootloader splash screen appears, release the joypad.

- For H5xxx: When the iPAQ buzzes, release the joypad. The screen will not change from whatever was previously displayed (blank, PocketPC, etc). If the iPAQ does not stop vibrating, remove the AC adapter and the battery, then reinsert the battery and the AC adapter and perform try this step again.

3) Press the calendar button on the iPAQ. This is the leftmost action button, labelled "Serial Bootldr Console" on the screen.

4) Make sure the terminal emulator is up and running, and is properly interacting with the bootloader. Proper interaction consists of being able to issue commands, and get responses (e.g. the help command should return the bootloader usage). Your terminal emulator must be set to **115200 8N1 serial configuration, no flow control, no hardware handshaking**. Failing to use these settings will lead to trouble, so double and triple check all settings.

5) If you cannot interact with the bootloader, make sure your terminal settings are correct, the iPAQ is properly connected to the host computer, and the iPAQ is actually on. If everything seems fine, try restarting the host terminal emulator and resetting the iPAQ again.
6) Hyperterminal is particularly ill-behaved. Sometimes it uses 100% of the CPU without allowing any interaction with the iPAQ. In that case, you will need to use the task manager to terminate Hyperterminal before you can restart it.
7) At the "boot>" prompt, issue the following command: load root
8) Proceed to send or "upload" the jffs2 file (from the tarball that you downloaded earlier) with ymodem, using the terminal emulator. If you have not used ymodem before or you have any trouble with this command, please see handhelds-faq/getting-started.html#USING-XYZMODEM. Note that the bootldr now expects ymodem by default, not xmodem as in earlier versions. If you are unable to use ymodem for some reason, you can revert to xmodem operation with the command set ymodem 0
9) You should see something like:

```
boot> load root
loading flash region root
ready for YMODEM download..
Erasing sector 00140000
Erasing sector 00180000
Erasing sector 001C0000
```

```
Erasing sector 00200000

.

.

.

addr: 00360000 data: 781590DB

addr: 00370000 data: 642637AE

addr: 00380000 data: E0021985

addr: 00390000 data: 15DA97EC

Erasing sector 00FC0000

writing flash..

addr: 00100000 data: E0021985

addr: 00110000 data: E3BAD617

addr: 00120000 data: 0FA1F57B

addr: 00130000 data: 9343AEEB

.

.

.

addr: 00600000 data: E0021985

addr: 00610000 data: FFFFFFFF

addr: 00620000 data: FFFFFFFF

addr: 00630000 data: FFFFFFFF

verifying ... formatting ... done.

boot>
```

[redacted]

10) At the "boot>" prompt, issue the following command: boot

Linux should now start booting.

Chapter A3

INSTALLATION OF FAMILIAR LINUX 0.8.1 ON IPAQ 5550

The iPAQ 5550 is an upgraded version of the iPAQs used earlier (i.e. iPAQ 3670 from Compaq) and these are manufactured by Hewlett Packard (HP).The default operating system installed in iPAQ 5550 is Microsoft Pocket PC 2003 premium. The iPAQ comes with a dual USB/Serial cable which is used for the setting communication from the iPAQ to a Host PC

The Installation process can be described in 3 major steps

1. Downloading the bootblaster and Linux bootloader
2. Install bootloader on the iPAQ

3. Load the Linux image from the bootblaster downloaded on the host PC using serial communication

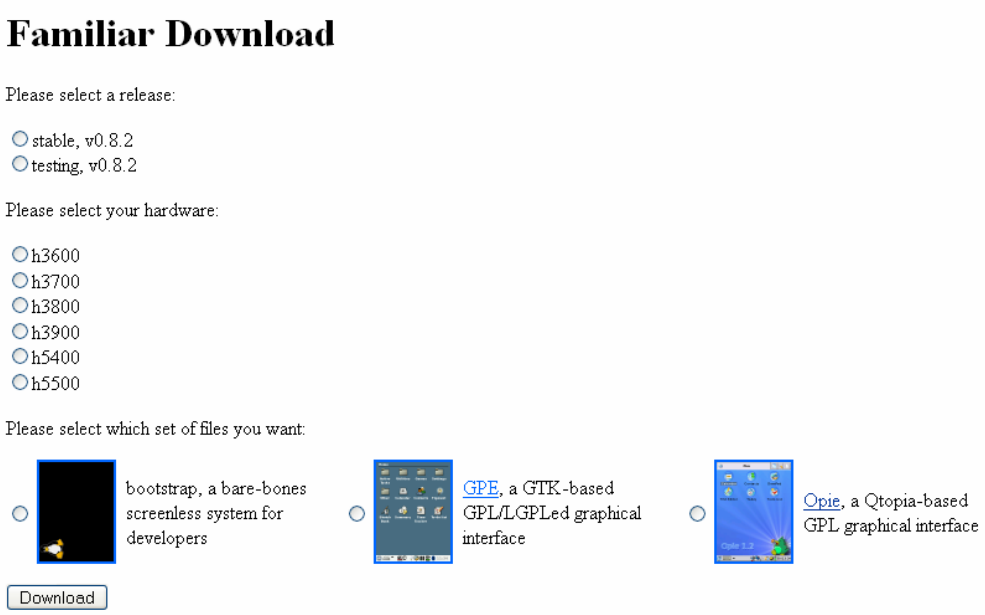
A3.1 DOWNLOADING THE FAMILIAR SOURCE

First we synchronize the Host PC with the PDA. For synchronizing iPAQ 5550 with the Host PC we need to install Active Sync on the Host PC. The iPAQ comes with Active Sync already installed on it and a companion CD. We can install Active Sync on the Host PC using this companion CD.

Notes

- Do **not** place the iPAQ in the cradle and connect to the Host PC until the Active Sync has been installed
- Remember to disable the serial communication during setting up the partnership else the serial port will be blocked for hyperterminal communication

1) The navigation directions for downloading the Familiar distribution are www.handhelds.org > *Under Quick Links* Familiar distributions > *Under contents* Download > *Under Version description* choose version > *Under installation* choose Download and the following page appears



We choose the stable version for h5500 series and the Opie graphical interface

Due to the complexity of the website, the navigation directions must be useful, also because of constant updating and shuffling of the links the navigation map may not be useful for a long time.

2) Copy the downloaded files to the iPAQ

- Plug the iPAQ cradle into and AC power outlet
- Connect the USB connector from the cradle to the Windows PC
- Slide the iPAQ into its cradle. If a "Set Up a Partnership" screen appears on the PC, choose "No" and then click Next.
- Copy **"BootBlaster3900-2.6.exe"** and ***"bootldr-pxa-2.21.12.bin"*** to the default folder on the iPAQ by clicking *Explore* in the Active Sync window and dragging the files there. Ignore any may need to convert messages

3) Start Bootloader

- Select “Start > Programs ” on the iPAQ touchscreen
- Tap on File Explorer
- Tap on BootBlaster

The first 2 steps describe the backing up of the Windows CE and the last 3 steps describe the installation procedure for Linux bootloader installation.

4) Save Pocket PC

- Execute “Flash > Save Bootldr.gz” in BootBlaster to save the bootloader in file
- “\My Documents\saved_bootldr.gz” on the iPAQ
- Execute "Flash > Save Wince .gz" in BootBlaster to save the PocketPC image in file "\My Documents\wince_image.gz" on the iPAQ. This takes two to three minutes

- Copy saved_bootldr.gz and wince_image.gz to your Windows PC.

    a. Select "View -> Refresh" in the ActiveSync Explore window on the PC. Icons for the saved_bootldr.gz and wince_image.gz files should appear.

    b. Drag the saved_bootldr.gz and wince_image.gz icons from the ActiveSync Explore window to a local folder on your PC.

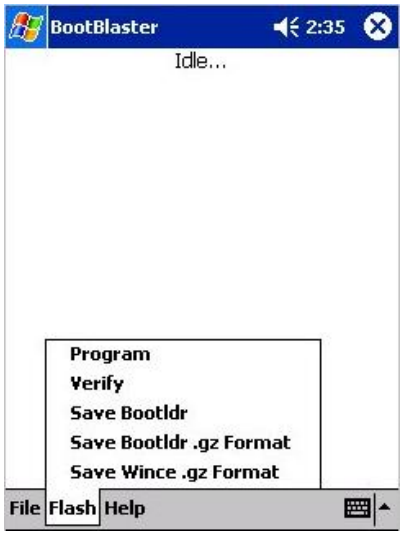
As with any backup files, please store saved_bootldr.gz and wince_image.gz in a safe place.

A3.2 INSTALL BOOTLOADER

The iPAQ needs to plugged into the external power and the battery should be fully charged to prevent power failure when the iPAQ is reprogramming the bootloader flash. Do **NOT** touch the power button or reset button until the “Verify” step is followed.

1) From the Flash menu in BootBlaster, select “Program”.

2) A file dialog will open allowing to select the bootloader to use. Select bootldr.bin.gz, which may have a version number embedded in it.*Wait patiently*. It takes about 15 seconds to program the bootloader. Do not interrupt this process, or the iPAQ may be left in an unusable state. From the "Flash" menu on BootBlaster, select "Verify".



3) From the "Flash" menu on BootBlaster, select "Verify".

4) If it does <u>not</u> say that you have a valid bootloader, do *NOT* reset the iPAQ, do *NOT* turn off the iPAQ.

5) Instead, try programming the flash again.

6) If that doesn't work, program the flash with the saved bootloader.

7) If everything has gone well, CRL bootldr program has been successfully installed which can run either Linux or PocketPC. As yet, the PocketPC image is intact and should restart normally; the next step actually installs Linux (overwriting Pocket PC).

A3.3 INSTALL FAMILIAR V0.8.1 WITH A SERIAL LINE

The Linux image needs to be loaded into the iPAQ from the Host PC through a serial line. We use the dual USB/Serial cable for this purpose.

1) The serial communication is done using a terminal emulator program. A Host PC running on Linux uses minicom or Kermit is the terminal emulator, and Windows PC uses hyperterminal.The terminal emulator must be set to **115200 8N1 serial configuration, no flow control, no hardware handshaking**. Failing to use these settings will lead to trouble, so double and triple check all settings.
2) Hold down the joypad and push the reset button on the iPAQ. You will need to remove it from the cradle to access the reset button. When the iPAQ buzzes, release the joypad. The screen will not change from whatever was previously displayed (blank, PocketPC, etc.).

   Note: If the iPAQ does not stop vibrating, remove the AC adapter and the battery, then reinsert the battery and the AC adapter and perform try this step again.
3) Press enter in the terminal emulator and you should see a boot prompt “boot>”
4) At the boot prompt issue the following command: load root; upload the “*.jffs2” file (from the tarball downloaded earlier) with ymodem, using the terminal emulator.
5) The screen appears as below

```
boot> load root
loading flash region root
ready for YMODEM download..
```

```
Erasing sector 00140000

Erasing sector 00180000

Erasing sector 001C0000

Erasing sector 00200000

.

.

.

addr: 00360000 data: 781590DB

addr: 00370000 data: 642637AE

addr: 00380000 data: E0021985

addr: 00390000 data: 15DA97EC

Erasing sector 00FC0000

writing flash..

addr: 00100000 data: E0021985

addr: 00110000 data: E3BAD617

addr: 00120000 data: 0FA1F57B

addr: 00130000 data: 9343AEEB

.

.

.

addr: 00600000 data: E0021985

addr: 00610000 data: FFFFFFFF

addr: 00620000 data: FFFFFFFF
```

```
addr: 00630000 data: FFFFFFFF

verifying ... formatting ... done.

boot>
```

6) At the “boot>” prompt issue the following command: boot; iPAQ starts booting

Note:The system **may** freeze while booting for the first time, don’t panic, now that the linux bootloader is already on the iPAQ. Just soft reset the iPAQ and the Linux should boot normally.

Chapter A4

INSTALLATION OF AODV-UU ON THE IPAQ 3670 AND IPAQ 5550

The Linux (Familiar kernel) installed on the iPAQ does not have Netfilter support. Thus it is necessary to recompile the kernel with the Netfilter support, which is required for application layer software (AODV-UU) to access the "raw" packets

The resources on the iPAQ are very limited thus we need to compile the AODV-UU on a Host PC and then transfer the executable file onto the iPAQ i.e. we need to cross compile the AODV-UU and transfer the ".exe" file to the iPAQ. The installation steps to be followed are listed below.

1. Download the cross compiler

   # wget ftp://ftp.handhelds.org/projects/toolchain/arm-linux-gcc-current.tar.gz

2. Unpack the cross compiler

cd/;tar zxvf /path/to/arm-linux-gcc-current.tar.gz (current implies the latest version; the zip file maybe in tar.gz or tar.bz2 format)

3. Retrieve the kernel source code matching the kernel used on the ARM Device.

   For the Familiar distribution, the kernel source code can be retrieved via anonymous CVS

   # export CVSROOT=:pserver:anoncvs@cvs.handhelds.org:/cvs

   # cvs login

   Password=anoncvs

   Get the matching version with "-r"

   # cvs export -r K2-4-19-rmk6-pxa1-hh37 linux/kernel

   Note: the kernel for familiar 0.7.2 is 2.4.19-rmk6-pxa1-hh37

4. Re-link the "asm" and "linux" include directories in arm cross-compiler tree to point to those in the ARM kernel source tree:

   # ln -s /path/to/arm-kernel-source/include/linux /path/to/cross-compilier/arm-linux/include/linux

   # ln -s /path/to/arm-kernel-source/include/asm /path/to/cross-compilier/arm-linux/include/asm

   No directory called include is present but creating the include directory in the arm-linux directory makes the relinking task work.

5. Make sure the ARM compiler is in the PATH and that /usr/src/linux points to the ARM kernel source

   # export PATH=$PATH :/path/to/cross-compiler/3.3.2(arm-linux version)/bin

Another way is to modify the Makefile in aodv-uu to explicitly specify the path of cross-compilier.

```
# ln -s /path/to/arm-kernel-source /usr/src/linux
```

The symbolic link may not be necessary.

6. Compile the new kernel to provide for netfilter support. Using the downloaded kernel from step 3, generate ip_queue.o module. The steps to carry on this process are listed.

```
# make ipaqsa_config
# make oldconfig
```

modify .config to change the setting for CONFIG_IP_NF_QUEUE as follows.

CONFIG_IP_NF_QUEUE=m

```
# make dep
# make zImage
# make modules
```

After that the ip_queue.o module will be generated. Copy the generated module to the iPAQ.

Note: Some errors are generated. When errors are generated some lines in the .config file, corresponding to the errors, have to be commented out. Then the ip_queue.o is generated

7. Compile AODV-UU for ARM

```
# make arm
```

Note: Modify the makefile to have the Kernel directory directed to arm kernel. To install, copy the aodvd and kaodv.o files to the iPAQ. The modules sometimes do not

install automatically, in that case we need to install the modules manually using the "insmod" command

```
# insmod kaodv.o
# insmod ip_queue.o
```

After these steps are performed the iPAQ device can be accessed as an ad hoc node.

Chapter A5

NETWORK TOOLS

The wireless networking tools used for performing the experiments on the mobile devices in a Linux environment are discussed in detail in this section.

1) IWCONFIG

The wireless configuration settings are returned when this command is entered in the terminal. The same command, with some additional attributes, is used to configure a mobile node for various settings.The attributes which are particularly used for the positioning experiments are listed below

- *Mode* is the mode of operation of the mobile node. The nodes are configured for Ad hoc mode of operation
- *Essid(Extended service set identifier)* is the identifier used to connect the nodes in a single network.

The syntax for *iwconfig* is

iwconfig *interface* **[essid** *X***] [nwid** *N***] [freq** *F***] [channel** *C***]**

[sens *S***] [mode** *M***] [ap** *A***] [nick** *NN***]**

[rate *R***] [rts** *RT***] [frag** *FT***] [txpower** *T***]**

[enc *E***] [key** *K***] [power** *P***] [retry** *R***]**

[commit]

2) /proc/net/wireless

The wireless statistics collected by the mobile node are saved in the /proc/net/wireless file. The field in the file are discussed below

- *Status* is the status reported by the modem.
- *Link q*uality reports the quality if the modulation on the air (direct sequence spread spectrum) [max=16]
- *Level and noise* refer to the signal level and noise level [max=64].

The *crypt discarded packet* and *misc discarded packet* counters are not implemented.

3) IWSPY

The values from the /proc/net/wireless file can be returned by using the *iwspy* command.The list of nodes, from which we collect the wireless statistics are listed initially before collecting the statistics. The syntax for listing the nodes is give below

iwspy *interface* **[+]** *DNSNAME* **|** *IPADDR* **|** *HWADDR* **[...]**

The statistics are collected by simply using the command

iwspy *interface*

4) NET FILTER AND IPTABLES

Netfilter and iptables are building blocks of a framework inside the Luinux 2.4.x and 2.6.x kernel. This framework enables packet filtering, network address and port translation (NA[P]T) and other packet mangling.

Netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack. The netfilter is the tool that provides for an application layer software (AODV-UU) to access the layer 3 of the network (internetworking layer).

Iptables is a generic table structure for the definition if rulesets. Each rule withina n IP table consists out of a a number of classifiers and one connected action All the nodes that we need in the experiment should be netfilter enabled and we need to flush out the iptables before performing the networkin experiments.