LOCALIZATION OF NODES IN WIRELESS SENSOR NETWORKS

Priya Puri	081424
Anshul Dhingra	081434
Anubhuti Verma	081452
Deepali Gupta	081459

Under the Supervision of

Mr. Amol Vasudeva

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

In

Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT,

SOLAN, HIMACHAL PRADESH

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the work titled "**LOCALIZATION OF NODES IN WIRELESS SENSOR NETWORKS**" submitted by "**Priya Puri, Anshul Dhingra, Anubhuti Verma and Deepali Gupta**" in partial fulfillment for the award of degree of **Bachelor of Technology** of Information Technology of Jaypee University of Information Technology, Waknaghat, has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor : [signature]

Name of Supervisor : Mr. Amol Vasudeva

Designation : Senior Lecturer, Dept. of CSE & IT, JUIT

Date : 29-05-2012

ACKNOWLEDGEMENT

This project has been an outcome of meticulous efforts on the part of every member of the project group. We take this opportunity to express our immense gratitude towards our project guide, Mr. Amol Vasudeva for his constant guidance that has led to the successful completion of this project.

Signature : Priya.
Name of Student : Priya Puri
Date : 29/05/2012.

Signature : Anshul Dhingra
Name of Student : Anshul Dhingra
Date : 29-05-2012

Signature : Anubhuti
Name of Student : Anubhuti Verma
Date : 29-05-2012

Signature : Deepali
Name of Student : Deepali Gupta
Date : 29-05-2012

SUMMARY

Our project is titled '**Localization of Nodes in Wireless Sensor Networks**'. The aim of the project is to determine the location of nodes in a wireless sensor network using certain strategies. These ways of locating nodes are Received Signal Strength Indicator (RSSI), Angle of Arrival (AOA), and Time difference of Arrival (TDOA).

In the Received Signal Strength Indicator method, we analyze the location of the node using the strength of the received signal. The strength of the signal received is inversely proportional to the distance it has travelled. This forms the basis of this method. When different sensor nodes receive signal from the same sensor node, we can calculate the location of the sensor node using mathematical analysis.**[12]**

In the Angle of Arrival method, a node called an anchor node has the information about the angle between the different nodes. This information helps in calculating the coordinates of the sensor node with greater accuracy. Combined, with RSSI method, it yields better results.

In the Time Difference of Arrival method, we make use of the principle that time taken by a signal to travel from one point to another is directly proportional to the distance between the two points. There are two ways to implement this. In the first way, the two sensor nodes need to have synchronized clocks and only one signal is sent. This way the time difference is calculated and the distance is estimated. In the other implementation, the clocks of the two sensor nodes need not be synchronized and instead two different types of signals are sent from the sender node at the same time. At the receiving node the time difference of arrival of the two signals helps in the calculation of distance between the two nodes.**[13]**

These methods help us to determine the location of the sensor nodes, which finds applications wherever wireless sensor networks are being used.

Signature of Supervisor : ____________________

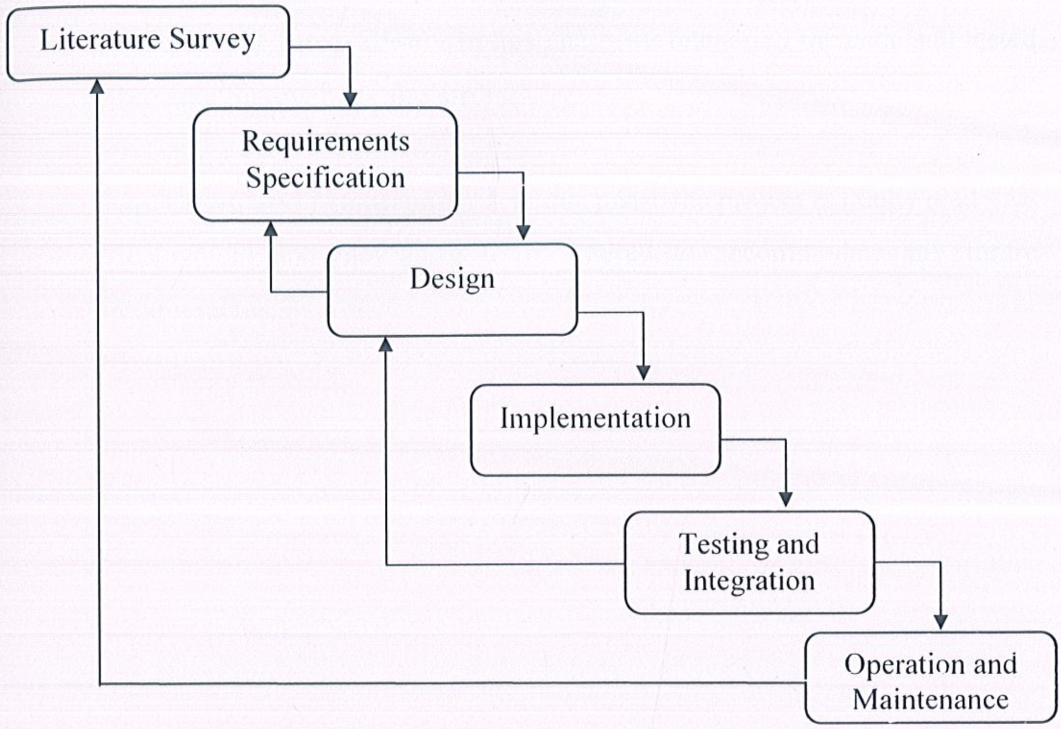
Name : Mr. Amol Vasudeva

Date : 29-05-2012

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
RSSI	Received Signal Strength Indicator
AOA	Angle of Arrival
TDOA	Time Difference Of Arrival
NS-2	Network Simulator 2
TCL	Tool Command Language
OTCL	Object-oriented Tool Command Language
WSN	Wireless Sensor Network

SDLC MODEL USED



The SDLC model that we have used to implement our project is the Waterfall Model. This model has the following phases :-

Literature Survey – In this phase of the SDLC model we did an extensive study of various literature and research papers available on Wireless Sensor Networks and localization on Wireless Sensor Networks.

Requirements Specifications – In this phase we specified the objective of the project and the specifications of the technology or software that would be used to implement the project.

Design – In this phase we devised the design on which the project would be based. This includes the various algorithms that would be used to solve the localization problem.

Implementation – This refers to the coding phase of the project. In this phase we implemented the various algorithms in NS2.

Testing and Integration – In this phase we integrated the code and tested the same for any bugs and errors.

Operation and Maintenance – In this phase the project is finally ready for deployment and may have to be altered to accommodate any future requirements.

CHAPTER – 1

INTRODUCTION AND LITERATURE SURVEY

What is a wireless sensor network?

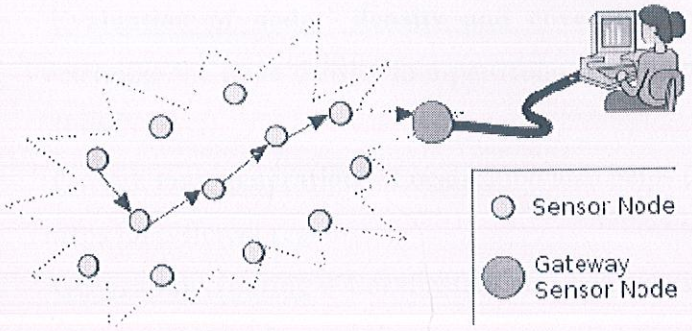
Wireless sensor network is a distributed system, which is made up of a large number of distributed sensor nodes. These nodes are generally inexpensive and have limited resources. A sensor receives and responds to signals. Therefore, in a wireless sensor network there'll be a large number of distributed sensor nodes that will be connected via signals. They communicate with each other by sending signals and they have no physical connection with each other.

The nodes send messages to each other for communication. The network is generally configured such that the sensor node and the base station are many hops away. Each node in the network is identified through a unique ID.

A wireless sensor network is not configured to a definite topology. As a result, there isn't a designated communication protocol and the network becomes ad hoc. Normally, the nodes are scattered throughout the region of interest in a random fashion leading to a random network topology. The nodes in such networks are aware of their environment, can perform some basic computations and communicate with the other nodes in the network including the central unit, if one exists. Such networks are increasingly being implemented in various fields today. They find application in areas ranging from military applications to environment monitoring to medical applications.

The network sensor node consists of the following components which help in the communication with other nodes in the network

- Radio transceiver with the internal antenna.
- Microcontroller, used for interfacing the sensor nodes.
- Energy source, usually a battery.



What is the need for localization of nodes?

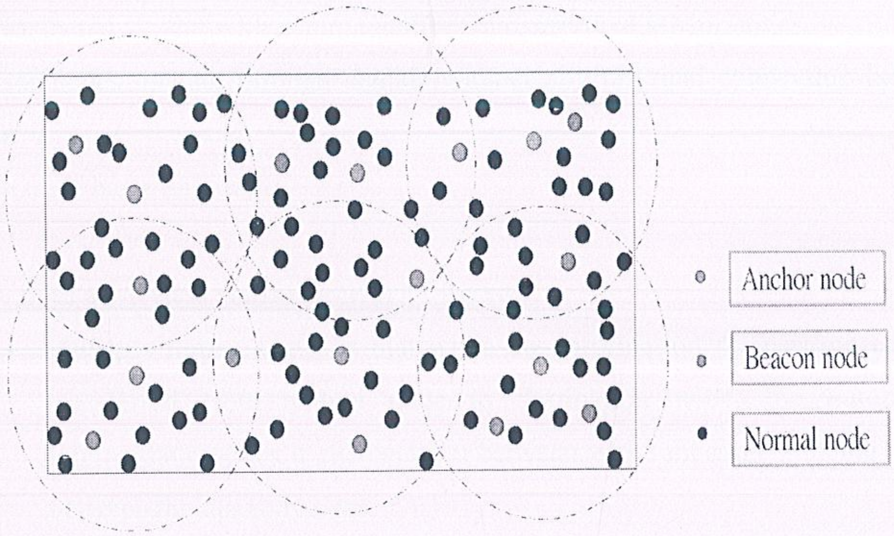
- **To report origin of events** – Localization helps reporting the origin of events. If a sensor node has detected something that it needs to report, then the exact location of the node ought to be known.

- **Group querying of sensors** – Localization also helps in group querying of sensors.
- **Routing** – Localization also helps in routing. Communication takes place by hop to hop forwarding. Localization helps in determining which nodes are in the same vicinity and thereby the nearest node to which a packet has to be forwarded.
- **To assess network coverage** – Localization also helps in assessing the network coverage by giving us the locations at which the nodes are spread.
- **Identification and correlation of gathered data** – Localization helps in identification and correlation of gathered data especially in surveillance applications. The data gathered at different locations may help point out a pattern which could be used further as per the demands of the application.
- **Node addressing** – Localization also helps in assigning addresses to the nodes.
- **Management and query of nodes localized in a determined region** – Localization helps us determine the number of nodes in a particular region. This is further used for management of these nodes and can be used to obtain results for area specific queries.

- **Evaluation of nodes' density and coverage** – Localization helps us determine the node density in a particular region and of the network as a whole.
- **Energy map generation** – Localization also helps us in generating energy maps for different regions.
- **Geographic routing** – Localization also helps us in routing according to geographical location.
- **Object tracking** – Sensor nodes attached to an object would help in object tracking through the localization of that node.
- **Other geographic algorithms** – Localization finds applications in many geographic algorithms like remote sensing, habitat monitoring etc.
- **Security** – Localization plays a vitally important role in security applications where secured communication protocols are established so that secured message passing is ensured in the network and effective communication takes place.
- **Air pollution monitoring** – Wireless sensors being mobile in nature provides the measuring of the level of the air pollution
- **In telecommunications and computer science**- Wireless sensor networks find a pivotal role in the computer science applications and telecommunications fields.

What is the meaning of localization of nodes?

The goal of localization is to determine the physical coordinates of a group of sensor nodes.



Anchor nodes are those nodes whose positions are known and which sends the angle information to another node whose position is to be determined with respect to the Anchor node. Generally the no of anchor nodes are less in number as compared to the beacon nodes. Moreover, Anchor nodes have the higher communication range and are deployed manually.

Beacon nodes are those nodes whose positions are also known and they send the packets to other nodes. Beacon nodes have the higher capacity of computation and more energy resource.

Normal nodes are those nodes which do not have any location information. Their positions are to be determined using the Anchor nodes and Beacon nodes. In other words, normal nodes localize themselves with the help of beacon nodes and anchor nodes.

What is a sensor node?

Sensor node is a device that receives and responds to a signal. They have limited computation capability and limited power supply.

What is wireless sensor network?

A distributed system with a large number of distributed sensor nodes.

Sensor nodes operate through signals without any physical connection between them.

Each node has a unique ID.

How does a wireless sensor network operate?

- **Multiple hops-** it means that while the information or message is to be transferred from source node to destination node, the number of intermediate nodes it encounter in between while message is being passed between the desired nodes.
- **Message Passing-** Communication takes place through the messages being passed between the nodes.
- Nodes can **sense** the environment, perform simple computations and communicate with other nodes or the central unit .
- **Ad hoc network-** No definite location of the nodes in the network .Nodes are distributed randomly.
- **Random topology-** No predefined topology of the nodes is being made. They are being distributed in the wireless network randomly and discretely.

The network is often configured such that the communication between the sensor nodes and the base stations requires multiple hops. These nodes operate through each other by sending messages to each other. Each node is assigned an ID, which is unique identity of the node. Each node is able to sense the environment, perform simple computations and communicate with its other sensors or to the central unit.

One way of deploying the sensor networks is to scatter the nodes throughout some region of interest. This makes the network topology random. Since there is no a priori communication protocol, the network is ad hoc.

The main aim is to enhance the efficiency of computation and reduce the power consumption of nodes. Since they have low computation and power supply algorithms should not be very complex and power saving schemes should be designed

Limitations of wireless sensor networks

- Limited hardware
- Limited transmission range
- Large scale network systems and traditional protocol mechanisms can't be used in WSNs

Normally, sensors are intended to be low-cost disposable devices, and currently GPS are inadequate for the hardware and power-limited sensors. Traditional localization techniques are not well suited for these requirements as they are not being made for the low- cost disposable devices. Including a global positioning system (GPS) receiver on each device is cost and energy prohibitive for many applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications. Local positioning systems (LPS) rely on high-capability base stations being deployed in each coverage area, an expensive burden for most low-configuration wireless sensor networks.

Hence, automatic localization of the sensors in wireless networks is a key enabling technology to be pondered upon. The overpowering reason is that a sensor's location must be known for its data to be meaningful. Sensor location information can be extremely useful for scalable, and geographic routing algorithms.

CHAPTER -2

REQUIREMENTS SPECIFICATION AND DESIGN

2.1 PROBLEM DEFINITION

Consider the case when we have deployed a sensor network consisting of N sensors at locations S = {S_1, S_2,.......,S_N}. Let S_{xi} refer to the x-coordinate of the location of sensor i and let S_{yi} and S_{zi} refer to the y and z coordinates, respectively. Constraining S_{zi} to be 0 makes the 2D version of this problem. Determining these locations constitutes the localization problem. Some sensor nodes are aware of their own positions, these nodes are known as anchors or beacons. All the other nodes i.e. normal nodes localize themselves with the help of location references received from the anchors. So, mathematically the localization problem can be formulated as follows: given a multi-hop network, represented by a graph G = (V, E), and a set of beacon nodes B, their positions {x_b, y_b} for all b ε B, we want to find the position {x_u, y_u} for all unknown nodes u ε U. Since it is a distributed network so all nodes are distributed randomly and hence we have to locate the particular node to be targeted.

2.2 TOOLS AND SOFTWARES USED

Operating System: The operating system used in Linux (Ubuntu 10.04). Linux was chosen as the network simulator that we are using, NS2 is compatible with Linux only. Linux belongs to the family of Unix like computer OS which use the Linux kernel. The benefit of Linux is that it can be installed on a wide-range of computer hardware starting from mobile devices, tablets, to desktop computers, laptops and supercomputers. Also, it is free and open source.

Network Simulator: The network simulator used is NS2. NS2 has been written in C++ and uses OTcl as an interface. NS2 is an Otcl interpreter with network simulation object libraries. The newer version NS2 has various advantages and improvements over the earlier verison NSv1. The more complex objects in *NSv1* have been decomposed into simpler components for greater flexibility. The configuration interface in NS2 is Otcl which an object-oriented version of Tcl. Also, the code to the OTcl interpreter is now separate from the main simulator.

NS2 is the ideal choice for a network simulator as it meets the needs of network simulation by using two languages – C++ and Otcl. C++ is faster to run but takes quite some time when it has to be changed, which makes it desirable for detailed protocol implementation. On the other hand, OTcl runs much slower but can be altered very quickly, making it suitable for simulation configuration. NS2 is the most widely used simulator tool these days for research on networks.

2.3 DIFFERENT SOLUTIONS TO LOCALIZATION PROBLEM IN A WSN

While communication, each node see other nodes by broadcasting messages with multiple node identifiers (ID). For this we require lightweight solutions which will support the transmission range of the sensor nodes of the wireless sensor network and also the resource constraints imposed on the wireless network sensor nodes.

Localization's two distinct components:

• ***Distance/angle estimation*:** It's responsible for determining information about distances and angles between two nodes, which is used by the other components of the localization system.

• ***Position computation***: It uses available information such as distances/angles and position of reference node and computes the position of unknown node.

Distance/angle estimation includes the

- RSSI (Received Signal Strength Indicator).
- AOA (Angle of Arrival).
- TDOA (Time Difference of Arrival).

Position computation includes the

- Trilateration
- Multilateration
- Triangulation

2.3.1 RSSI (Received Signal Strength Indicator)

RSSI uses the information about the strength of the signal received to estimate the distance between two nodes. A sender node sends a signal with a certain strength that eventually fades as the signal propagates. The larger is the distance between sender and receiver, the lesser is the strength of the signal when it arrives at that node. The signal which is sent by the sender is not actually similar to the signal received by the receiver, it actually attenuates and distorts or we can say that power intensity present in the sent signal is actually more as compared to the power intensity present in the received signal**[8]**. Received Signal has less power strength in the signal because attenuation increases with the increase in distance. More is the distance separation, more is the attenuation.

The strength of the signal is inversely proportional to squared distance. Generally a radio propagation model is used to convert the signal strength into distance but in real-world environments, this indicator is accompanied by noises and obstacles.**[6]**

This method has both advantages and disadvantages associated with it. The main advantage is that its cheap as most receivers are capable of estimating the strength of the received signal.**[6]**

The disadvantage is that it is very susceptible to noise and interference, which results in higher rate inaccuracy.

Understanding Localization using power concept in RSSI

If we have sensors monitoring radio signals, then no user can hide its location.

Suppose node **i** receives radio signal from node **0**, then the RSSI is

$$R_i = (P_0 \cdot K)/d_i^{\alpha} \qquad \text{------} \qquad (\textit{equation 1})$$

Where P_0 represents transmitter power,

R_i is RSSI,

K is constant which depends on the medium present in which power signal is being transferred and communication takes place,

d_i is Euclidean distance,

a is distance-power gradient.

Now further, if we assume that node **k** receives radio wave from node **0** at the same time, then the P_k is similar to above equation

$R_k = (P_0 \cdot K)/d_k^a$ --------- *(equation 2)*

Hence further, the RSSI ratio of node i to k is obtained by dividing (1) and (2)

$$R_i/R_k = ((P_0 \cdot K)/d_i^a)/((P_0 \cdot K)/d_k^a)$$

$$= (d_k^a)/(d_i^a)$$

Applying the distance formula concept to find the Euclidean distance then

$$((x - x_i)^2 + (y - y_i)^2) = (R_i/R_k)^{1/a}((x - x_k)^2 + (y - y_k)^2)$$ [6]

To understand the RSSI concept which illustrates the increasing attenuation with increasing distance is lognormal model. Also the mean RSS decays between transmitter and receiver is measured by radio propagation model. The most commonly used propagation model is the Log normal shadowing model that considers the shadowing effect both outdoor or indoor environment, interference effects and noise effects. The model indicates that the average received signal strength decreases logarithmically with distance. In general, the average path loss for an particular T-R separation can be expressed as follows:

$$P_r(d) = P_t(d_0) - 10n\log(d/d_0) + x_q$$

where

n is the path loss exponent, which relays on the specific propagation environment,

P_r **(d)** represents the received signal strength **(RSS).**

$P_t(d_0)$ representing the transmission power at reference distance (d_0)

X_q is a random variable that accounts for the random variation of the path loss, and is supposed to be Gaussian distribution with zero mean random variable (in dB) with standard deviation **q** (also in dB).**[9][10]**

Understanding RSSI using coordinate computation

Let (x, y) be the coordinate of the normal node,

(x_1, y_1) be the location of beacon node B_1, and

$(x_a; y_a)$ be the location of anchor node.

Distance between the beacon and normal node is d_1.

Let, the angle between the anchor node's x-axis and the line joining the normal and anchor node be **θ**

Based on these information, we can obtain two equations.

Then,

When a line makes an angle of θ with x-axis ,then the slope is given by **tanθ** , so the line equation becomes as follows:

$$y_a = (\tan\theta) * x_a + k$$

Rearranging the equation, it becomes:

$$k = y_a - x_a * \tan\theta$$

Considering the boundary of the sensing range of a beacon node as equation of a circle with radius d_1, The locus of normal node whose position coordinates are to be determined is circle, we get equation as follows:

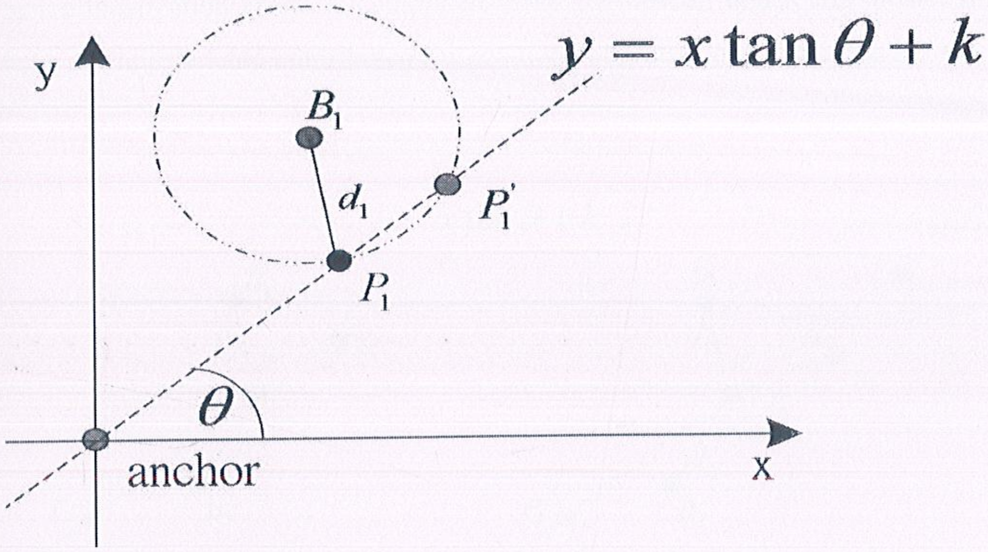
$$(x - x_1)^2 + (y - y_1)^2 = d_1^2$$

Substituting equation into above and upon simplification we obtain equation

$$(1 + \tan^2\theta)x^2 - (2x_1 + 2y_1\tan\theta - 2k\tan\theta)x + R = 0 \qquad (1)$$

where R is

$$R = x^2_1 + k^2 - 2ky_1 + y^2_1 - d^2_1 \qquad (2)$$



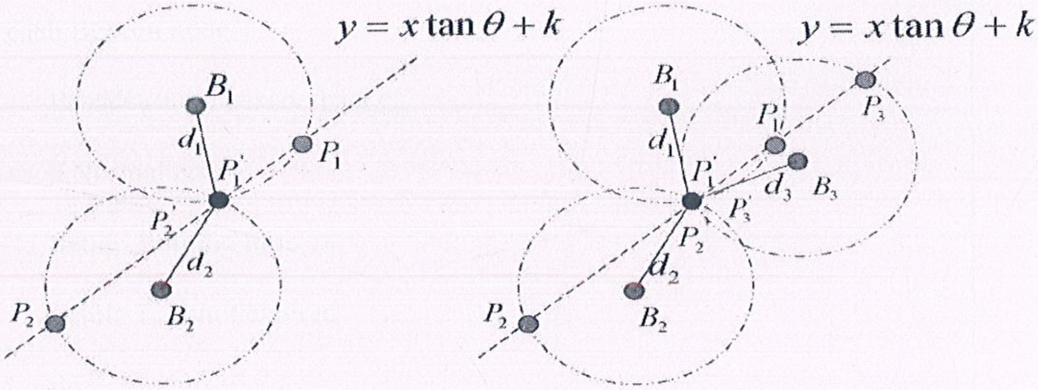
Hence, location of the normal node (x; y) can be estimated from equation (1) as given in equation (2), which is obvious.

$$x = (-b + (b^2 - 4aR)^{1/2})/2a$$

where Discriminant (D) is given by $(b^2 - 4aR)^{1/2}$, a and b are coefficient of x^2 and x, respectively and R represents a constant term. Then, we substitute equation in above equation to get the y coordinate.

What actually happens in finding the coordinates of the normal node?

Suppose that in wireless scenario we have **N** beacon nodes and **N** anchor nodes. We require at least one beacon node which sends information to the normal node for localization and an anchor node that sends angle information to the normal node. When normal node receives a beacon packet from one beacon node and angle information from the anchor node, it waits for a predefined timeout T_n to receive RSSI value from other beacon nodes of its cluster.**[9][10]** When normal node receives the beacon packets from all the beacon nodes and angle information from all the other anchor nodes, then it need not to wait for the predefined time and hence localization effect is completed as all the beacon nodes and anchor nodes have participated and contributed for position computation of the normal node.



RSSI Algorithm

```
Initial;
        Initialize: Waiting time Tn for each normal node;
        Initialize: All fields of coordinate table ={ Φ };
Do
        Start: Node deployment strategy;
For each Anchor nodes:
        Check: Neighbours of normal nodes;
        Measure: Angle information for each neighbour of normal nodes;
        Transmit: Angle information to each normal nodes;
For each Beacon nodes:
        Broadcast the beacon packet;
For each Normal nodes:
        Setup: Waiting time Tn;
        While Tn is not expired
        do
                Listen the network;
                If   Any beacon packet is arriving
                        Translate: RSSI into Distance;
                        Computation:
                        Update the coordinate table;
                End If
        Calculate: Final result from all entries of the table;
        Output: Normal node's location;
End
```

Distributed algorithm maintains the Coordinate table which will have the effect of all the beacon and anchor nodes in vicinity and when all the entries are filled which are shown below, then only we can say that the wireless sensor network is a localized one. **[9]**

When all the beacon and anchor nodes have contributed to the position computation of the normal node , then we can make the coordinate table which will illustrate which beacon nodes and which anchor nodes have made its contribution effect and hence RSSI values pertaining to each of the normal nodes.

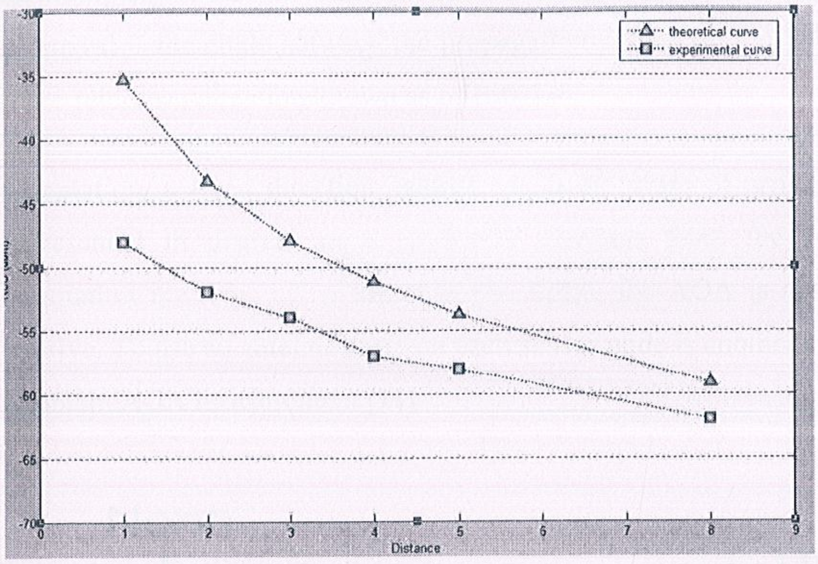
The updated coordinate table looks like:

Table 1: COORDINATE TABLE

BN-ID (Beacon node ID)	BN-loc. (Beacon node location)	RSSI	P-loc (Point Location)
B_1	(X_{B1},Y_{B1})	$RSSI_{B1}$	P_1
B_2	(X_{B2},Y_{B2})	$RSSI_{B2}$	P_2
B_3	(X_{B3},Y_{B3})	$RSSI_{B3}$	P_3
B_4	(X_{B4},Y_{B4})	$RSSI_{B4}$	P_4
…….	…….	…….	…….

Error estimations in RSSI

The error estimates in RSSI scheme is measured by the comparison of measured and known values of RSSI.**[9]**



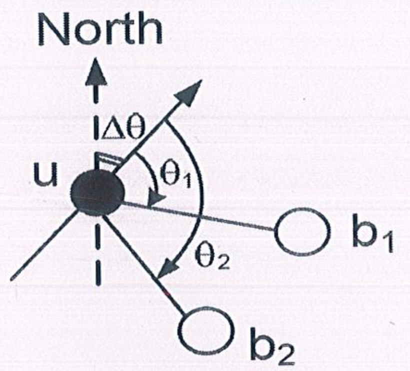
The error in measuring the distance of deployed sensor nodes from reference nodes is estimated by comparing the theoretical values and known results. The table below shows the distance measurement (Dm) corresponding to each measured RSSI values(dBm) and known distance (D). The difference of both the distances will give the error estimate "e".

Distance of Deployed Node(D)(Meter)	Measured RSS Value (dBm)	Measured Distance (Dm)(meter)	Error(e)=Dm-D
1	-48	3.0252	2.0252
2	-52	4.2858	2.2858
3	-54	5.1013	2.1013
4	-57	6.6243	2.6243
5	-58	7.2271	2.2271
8	-62	10.2359	2.2359

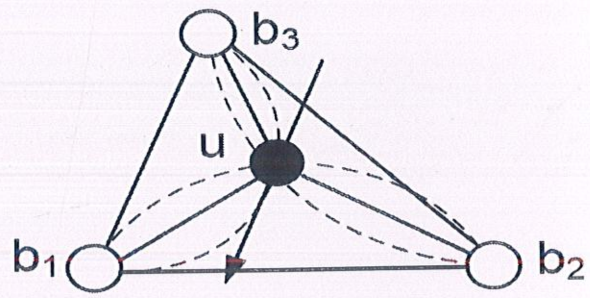
2.3.2 AOA (Angle of Arrival)

RSSI takes into account the power strength of the signal which decreases as the distance increases but in AOA technique we take takes into angle at which incoming signal arrives to the receiver which is called angle of arrival..**[11]**Angle of arrival is the angle between the propagation direction of an incident wave with some reference direction known as orientation.

Orientation, a fixed direction against which the angle of arrival is measured, is represented in degrees in a clockwise direction from the North. When the orientation is 0° or is pointing to the North, the AOA is absolute, otherwise, relative. To use an antenna array on each sensor node is another common approach to obtain AOA measurement.**[11]**



Absolute Orientation



Relative Orientation

At the time of deployment the orientations of the unknowns may or may not be known. Localizations under both of these scenarios can be solved using triangulation. First we consider the case when the orientations of the unknowns are known.

In Fig above, angles θ_1 and θ_2, which are measured at unknown u, are the relative AOAs of the signals sent from beacons $b_1$and b_2, respectively. Assuming the orientation of the unknown is $\Delta\theta$, the absolute AOAs from b_1 and b_2 can be calculated as $(\theta_i + \Delta\theta)(\text{mod } 2\pi)$, $i = \{1,2\}$. Each absolute AOA measurement corresponding to a beacon restricts the location of the unknown along a ray starting at the beacon.**[11]**

In the figure adjacent to the previous figure above, angles b_1ub_2 , b_1ub_3 **and** b_2ub_3 can be computed

using the knowledge of the relative AOAs.

The location of the unknown **u** is located at the intersection of all rays when two or more non-collinear beacons are available.

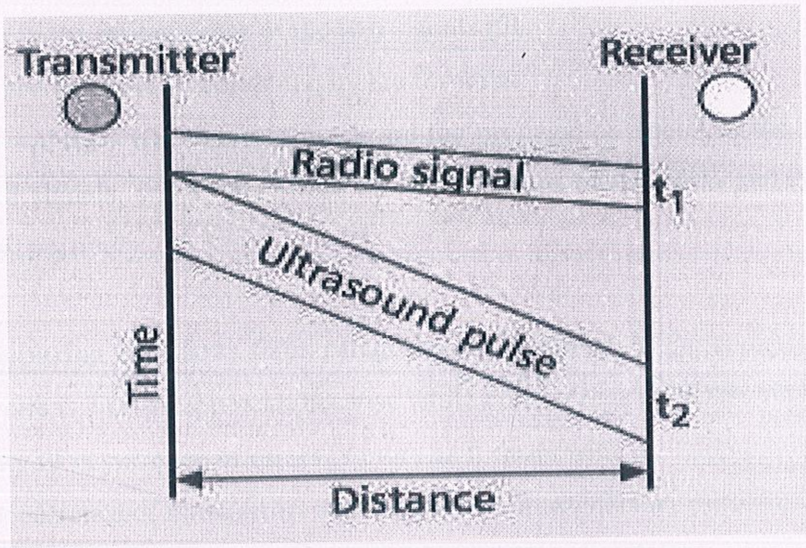
Although the angle of arrival concept gives the accurate results to a certain extent , but it also suffers from interference and noise effects when the unknown nodes are not able to communicate properly

with the beacon nodes which means that the angle measurement is not being done properly.

2.3.3 TDOA (Time Difference of Arrival)

The distance between two nodes is directly proportional to the time the signal takes to propagate from one point to another.**[2][3]**

If a signal was sent at time t_1 and reached the receiver node at time t_2, the distance between sender and receiver is $d = s_r(t_2 - t_1)$, where s_r is the propagation speed of the radio signal (speed of light), and t_1 and t_2 are the times when the signal was sent and received.**[2]**



This type of estimation requires precisely synchronized nodes and the time at which the signal leaves the node must be in the packet that is sent.

TDoA is based on:

- The difference in the times at which a single signal from a single node arrives at three or more nodes.
- The difference in the times at which multiple signals from a single node arrive at another node.**[2]**

The **first** case, which is more common in cellular networks, requires precisely synchronized receiver nodes (in this case base stations). In the **second** case, more common and suitable for WSNs, the nodes must be equipped with extra hardware capable of sending two types of signals simultaneously.

These signals must have different propagation speeds, like for example a radio/ultrasound or

Radio / acoustic. Usually, the first signal is the packet itself, which propagates at the speed of light (» 300.000 km/s), while the second signal is some kind of sound, because of its slower propagation (» 340 m/s), which is six orders of magnitude slower then the first signal.**[3]**

An example of TDOA suitable for WSNs is used, where an ultrasound pulse is sent simultaneously with a radio signal. In this case nodes compute the difference in the arrival times of the two signals.**[5]**

The distance can now be computed by the formula:

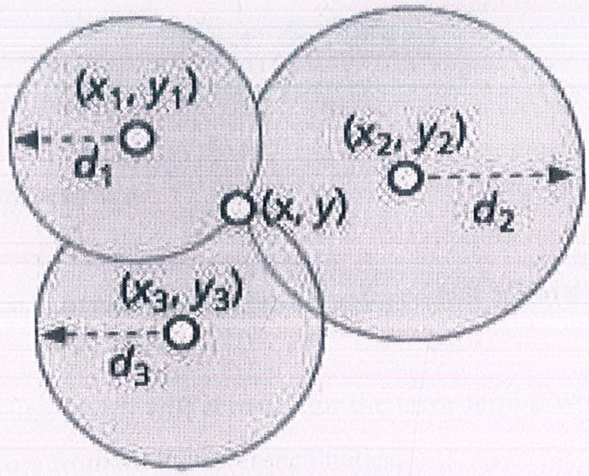
$\mathbf{d = (s_r - s_s)*(t_2 - t_1)}$, where $\mathbf{s_r}$ and $\mathbf{s_s}$ are the propagation speed of the radio and ultrasound signals, and $\mathbf{t_1}$ and $\mathbf{t_2}$ are the arrival times of the radio and ultrasound signals, respectively.

Now **comparing** the above three categories of Distance/ Angle computation, the accurate one is TDOA approach because while considering the RSSI approach, the probability of occurrence of errors is more as it deals with the attenuation concept. Hence, occurrence of more errors will lead to the unsatisfactory performance of the localization system. On the other hand, if we consider the TDOA approach, then as compared to the previous methods, more synchronisation is present in TDOA and it delivers better performance with occurrence of less errors.

Position Computation

2.3.4 Trilateration and Multilateration

This method computes a node's position via the intersection of three circles.**[2]**



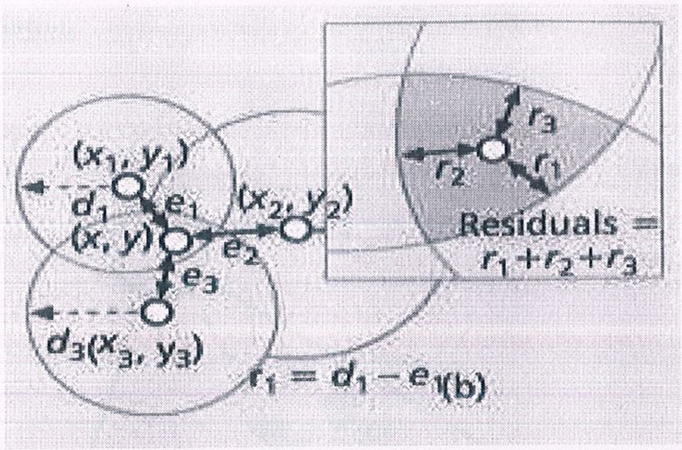
Taking the unknown node in between and the no. of nodes whose positions are known at the centres such that the locus of the unknown node is a circle with a centre of known nodes and the radius of d_i from the unknown node.**[2]**

Mathematically,

$$(x^{\wedge} - x_i)^2 + (y^{\wedge} - y_i)^2 = d_i^2$$

Where $(x^{\wedge}, y^{\wedge})$ is the position we want to compute, (x_i, y_i) is the position of the i^{th} reference node, and d_i is the distance of the i^{th} reference node to the unknown node.

In **Multilateration** , we have larger no. of reference points and the locus of unknown node is a circle with reference node acting as a centre with distance between them as a radius but with an error term between them.

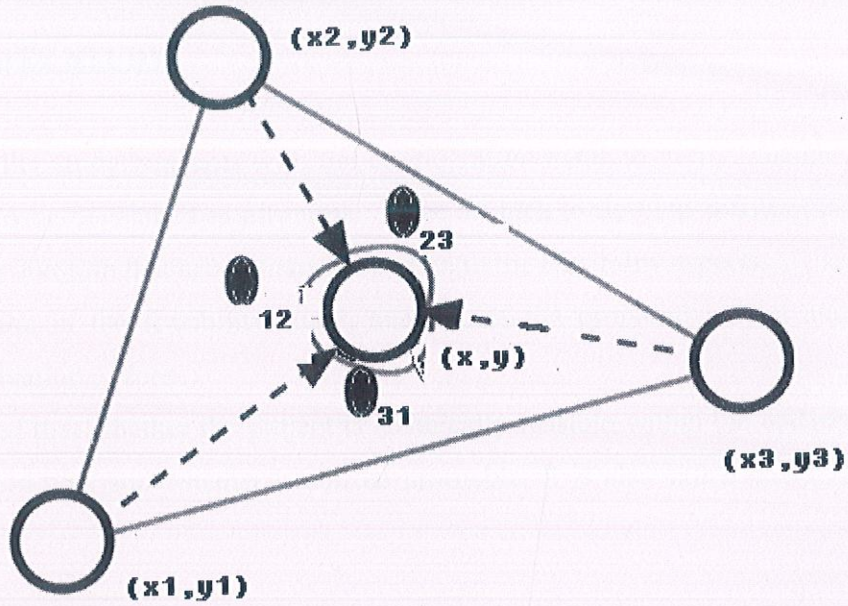


$$(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2 = d_i^2 - e$$

where **e** is normally considered to be an independent normal random variable with zero mean.

The no. of increasing nodes will account for the error term **e** which is cumulative effect of all the errors from all the reference nodes.

2.3.5 Triangulation



In triangulation information about angles is used instead of distances.

Position computation can be done remotely or by the node itself, the latter is more common in WSNs.

In triangulation, at least three reference nodes are required. The unknown node estimates its angle to each of the three reference nodes and, based on these angles and the positions of the reference nodes (which form a triangle), computes its own position using simple trigonometrical relationships.**[2][3]**

Here the locus of the unknown node is the triangle with the reference nodes acting as the three vertices of the triangle.

CHAPTER – 3

FEASIBILITY STUDY

FEASIBILITY STUDY

The feasibility study concerns with the consideration made to verify whether the system fit to be developed in all terms. Once an idea to develop software is put forward the question that arises first will pertain to the feasibility aspects.

Moreover the in the feasibility study, analysis of the proposed project and its further evaluation is done.

It is checked that whether the project is technically feasible within the underlying constraints of cost and whether it will be profitable. It is also called cost benefit analysis.

There are different aspects in the feasibility study:

- Operational Feasibility.
- Technical Feasibility.
- Economical Feasibility.
- Time Feasibility.

OPERATIONAL FEASIBILITY:

There in no difficulty in implementing the system, if the user has the knowledge in internal working of the system. Therefore, it is assumed that he will not face any problem in running the system. The main problem faced during development of a new system is getting acceptance from the users. As users are responsible for initiating the development of a new system this is rooted out.

TECHNICAL FEASIBILITY:

Technical feasibility deals with the study of function, performance, and constraints like resources availability, technology, development risk that may affect the ability to achieve an acceptable system.

ECONOMICAL FEASIBILITY:

One of the factors, which affect the development of a new system, is the cost it would incur. The existing resources available in the company are sufficient for implementing the proposed and hence no extra cost has to be incurred to run the system developed. Thus, the system is financially feasible.

TIME FEASIBILITY:

Time feasibility tells about the period of time in which the software/ hardware should produce an accurate result. The time given(1 year) is sufficient to carry out the project work. Thus it is time feasible as well.

CHAPTER – 4

IMPLEMENTATION

SYSTEM SPECIFICATIONS

HARDWARE REQUIREMENTS

CPU : 1.6 GHz

RAM : 2 GB

SOFTWARE REQUIREMENTS

OS : Linux (Ubuntu 10.04)

Software : Network Simulator(NS 2)

PROGRAMMING LANGUAGE

Tool Command Language

C++

LINUX OPERATING SYSTEM

It is basically a Unix-like computer operating systems using the Linux kernel .Linux can be installed on a wide variety of computer hardware, ranging from mobile phones to desktop computers, mainframes and supercomputers.

The **main reason** of using the Linux while installing the network simulator NS2 is that there are some components of NS2 i.e. **Xgraph and NAM window** which do not works on the windows operating system. Alternatively, if we want to work on windows then, we can use the **cygwin software** which is similar to that of the NS2 but the main advantage of it is that it can be installed on windows unlike NS2.

NETWORK SIMULATOR

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors.

Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to substantial contributions from the players in the field.

NS2 provides users with an executable command "ns" which takes on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command **ns**. Then a simulation trace file is created which is used to plot graph and/or to create animation.

Simulation provides us with following **advantages:**

– sometimes cheaper

– find bugs (in design) in advance

– generality: over analytic/numerical techniques

– detail: can simulate system details at arbitrary level

And it has following **drawbacks**:

– caution: does model reflect reality

– large scale systems: lots of resources to simulate (especially accurately simulate)

– may be slow (computationally expensive – 1 min real time could be hours of simulated time)

–art: determining right level of model complexity

–statistical uncertainty in results

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as *handles*. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g., _o10) in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It may defines its own procedures and variables to facilitate the interaction. Note that the member procedures and variables in the OTcl domain are called instance procedures (instprocs) and instance variables (instvars), respectively

Tcl is a general purpose scripting language. While it can do anything other languages could possibly do, its integration with other languages has proven even more powerful. Tcl runs on most of the platforms such as Unix, Windows, and Mac. The main strength of Tcl is its simplicity. It is not necessary to declare a data type for variable prior to the usage. At runtime, Tcl interprets the codes line by line and converts the string into appropriate data type (e.g., integer) .

Tcl's features include

- All operations are commands, including language structures. They are written in prefix notation.
- Everything can be dynamically redefined and overridden.
- All data types can be manipulated as strings, including source code.
- Event-driven interface to sockets and files. Time-based and user-defined events are also possible.
- Variable visibility restricted to lexical (static) scope by default, but uplevel and upvar allowing procs to interact with the enclosing functions' scopes.

4.1 BASIC ARCHITECTURE

NS2 provides users with an executable command ns which takes on input an argument, the name of a Tcl simulation scripting file.

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL.

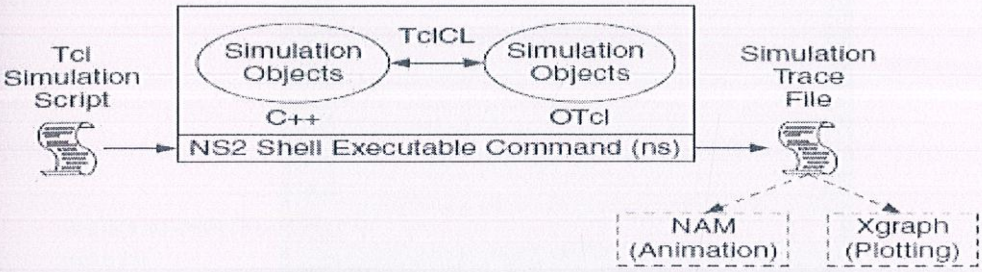
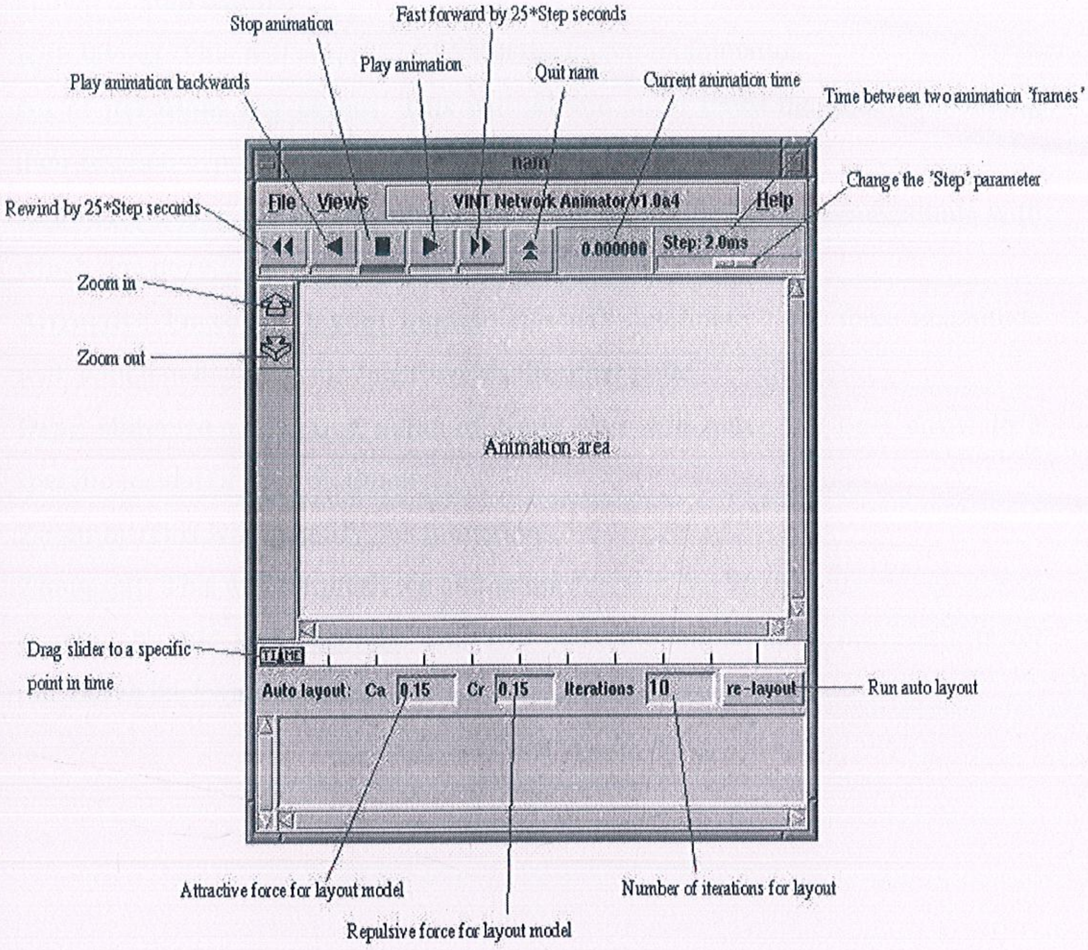


Fig. 2.1. Basic architecture of NS.

NS2 provides a large number of built-in C++ objects. It is advisable to use these C++ objects to set up a simulation using a Tcl simulation script. However, advance users may find these objects insufficient. They need to develop their own C++ objects, and use a OTcl configuration interface to put together these objects.

After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network AniMator) and XGraph are used. To analyze a particular behavior of the network, users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.

4.2 SCREENSHOT OF NAM WINDOW



Explanation of components of NAM window

Play animation: This will start the animation.

Stop animation: This will stop the animation

Fast forward: This will make the animation to move gain speed i.e. nodes moving at a faster rate.

Quit nam: This will close the nam window

Current Animation time: This will tell the current time where the current state of animation is

Time between two animation frames: This will determine the time difference between the current state of animation and previous state of animation. It is also known as **step count**

Run layout: This will automatically run the layout of animation.

No of iterations per layout: This will tell the no of times the node is changing their respective positions.

Repulsive Force for layout model: This will determine the force magnitude with which one node is moving away from the other node.

Attractive Force for layout model: This will determine the force magnitude with which one node is moving towards the other node.

Drag slider to respective point of time: This will move the time arrow to a specific instant of time on timeline

Zoom in: This will magnify the animation.

Zoom out: This will diminish the animation

Rewind by 25* step seconds: This will make the animation to move at a faster rate but in backward manner.

4.3 SIMPLE SIMULATION IN NS-2

- Creating a Simulator Object

```
set ns [new Simulator]
```

- Setting up files for trace & NAM

```
set trace_nam [open out.nam w]
```

This line will open the file **'out.nam'** for writing and gives it the file handle variable **'nf'**. NAM file is generally used for the displaying of what we have coded in tcl language.

```
set trace_all [open all.tr w]
```

we tell the simulator object that we created above to write all simulation data that is going to be relevant for nam into this file which handled by the **nf** file object .

- Tracing files using their commands

```
$ns namtrace-all $trace_nam
$ns trace-all $trace_all
```

- Closing trace file and starting NAM

```
proc finish { } {
        global ns trace_nam trace_all
        $ns flush-trace
        close $trace_nam
        close $trace_all
        exec nam out.nam &
        exit 0 }
```

This finish procedure is required to finish the simulation by closing all the trace_nam and trace_all variables which contain the simulation information.

- **$ns at 5.0 finish**

This line tells the simulator object to execute the 'finish' procedure after 5.0 seconds of simulation time.

- **$ns run**

  This will run the simulation

- **Creating LINK & NODE topology**

  Creating NODES

  ```
  set n0 [$ns node]
  set n1 [$ns node]
  set n2 [$ns node]
  set n3 [$ns node]
  set r1 [$ns node]
  set r2[$ns node]
  ```

- **Creating LINKS**

  ```
  $ns duplex-link $N1 $R1 2Mb 5ms DropTail
  set DuplexLink0 [$ns link $N1 $R1]
  $ns duplex-link $N2 $R1 2Mb 5ms DropTail
  set DuplexLink1 [$ns link $N2 $R1]
  $ns duplex-link $R1 $R2 1Mb 10ms DropTail
  set DuplexLink2 [$ns link $R1 $R2]
  $ns duplex-link $R2 $N3 2Mb 5ms DropTail
  set DuplexLink3 [$ns link $R2 $N3]
  $ns duplex-link $R2 $N4 2Mb 5ms DropTail
  set DuplexLink4 [$ns link $R2 $N4]
  ```

This will make duplex link (which means that flow between the nodes can be in any direction) between two nodes with a bandwidth of 2 Mb , delay time of 5 ms and a queue which is used for the accumulation of packets.

- **Orientation of links**

  $ns duplex-link-op $N1 $R1 orient right-down

  $ns duplex-link-op $N2 $R1 orient right-up

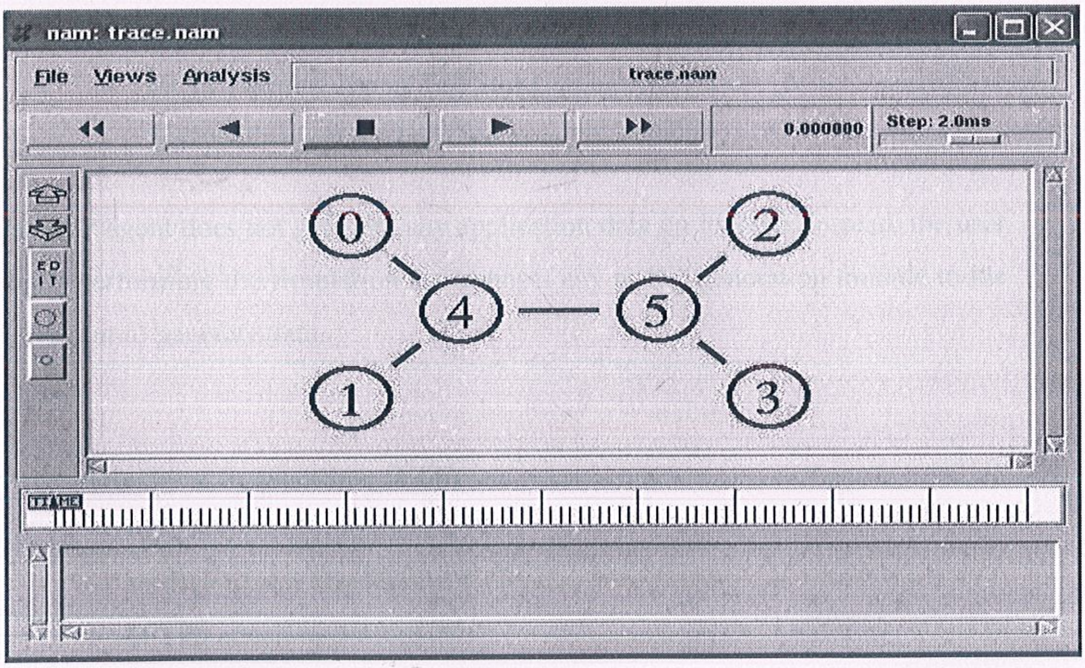
  $ns duplex-link-op $R1

  $R2 orient right

  $ns duplex-link-op $R2 $N3 orient right-up

  $ns duplex-link-op $R2 $N4 orient right-down

<u>FINAL TOPOLOGY CREATED</u>



4.4 GENERATING TRAFFIC

- **Attaching AGENT TCP to NODE 1**

```
set TCP1 [new Agent/TCP]
$ns attach-agent $N1 $TCP1
```

- **Attaching AGENT TCP to NODE 2**

```
set TCP2 [new Agent/TCP]
$ns attach-agent $N2 $TCP2
```

- **Attaching AGENT TCP to NODE 3**

```
set TCP3 [new Agent/TCPSink]
$ns attach-agent $N2 $TCP3
```

- **Attaching AGENT TCP to NODE 4**

```
set TCP4 [new Agent/TCPSink]
$ns attach-agent $N2 $TCP4
```

Attaching AGENT TCP to NODE 4

```
set TCP4 [new Agent/TCPSink]
$ns attach-agent $N2 $TCP4
```

The TCP agent does not generate any application data on its own. Instead, the user who is performing the simulation can connect any traffic generation module to the TCP agent to generate data.

- **Attaching Application (FTP)**

```
set FTP0 [new Application/FTP]
set FTP1 [new Application/FTP]
$FTP0 attach-agent $TCP0
$FTP1 attach-agent $TCP1
```

FTP(File transfer protocol): It is a type of the application of a TCP which is used to represent the bulk data transfer of large size. In NS2 , FTP does not need an input file. It simply performs the attached sending transport layer agent of file in bytes. Upon receiving demand , the agent creates the packets which can accommodate the file and forwards them to the connected receiving transport layer agent.

The following are methods of the Application/FTP class:

attach-agent: attaches an Application/FTP object to an agent.

start :start the Application/FTP by calling the TCP agent's send(-1) function, which causes TCP to

behave as if the application were continuously sending new data.

stop: stop sending.

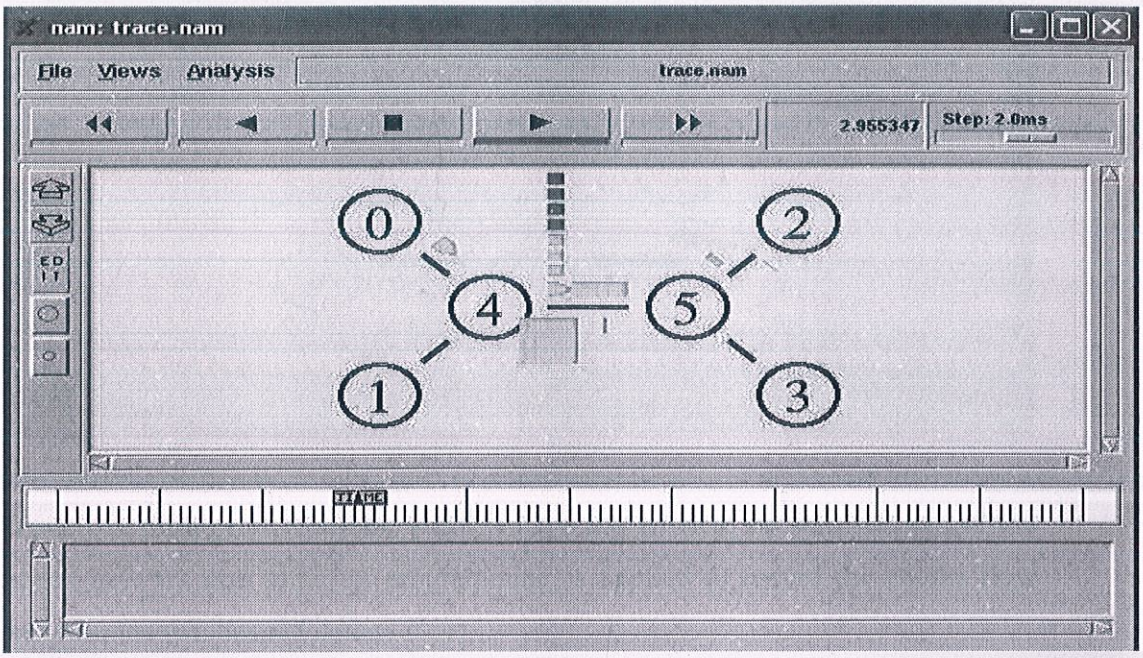
produce n :set the counter of packets to be sent to n.

producemore n: increase the counter of packets to be sent by n.

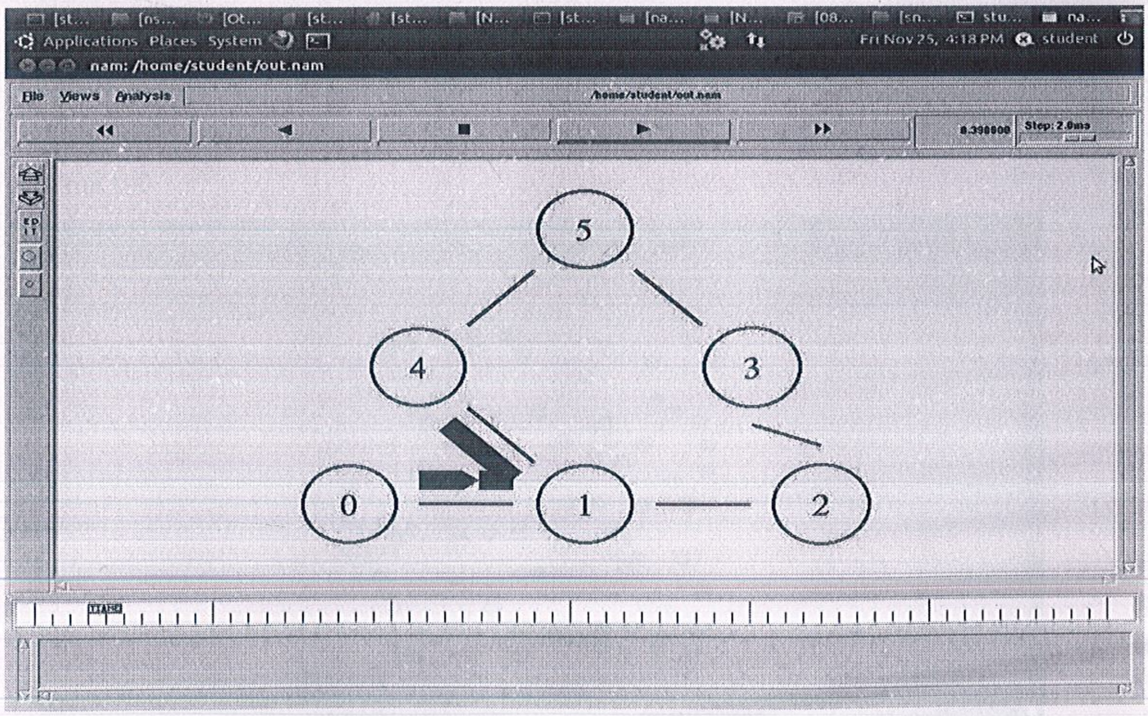
send n: sends n bytes instead of packets.

Snapshots of the packet flow between the nodes in the network

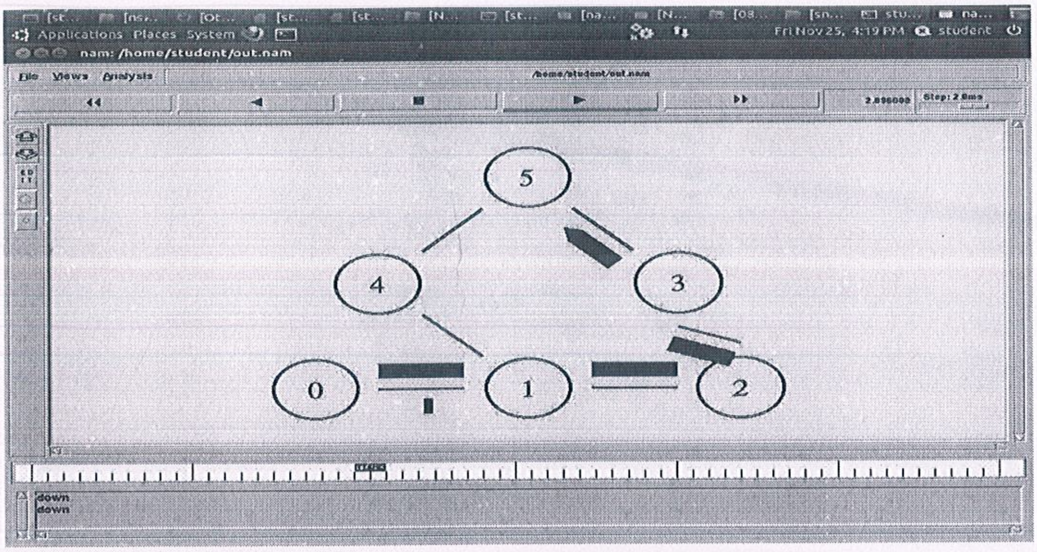
At time t=0



At time t=0.5



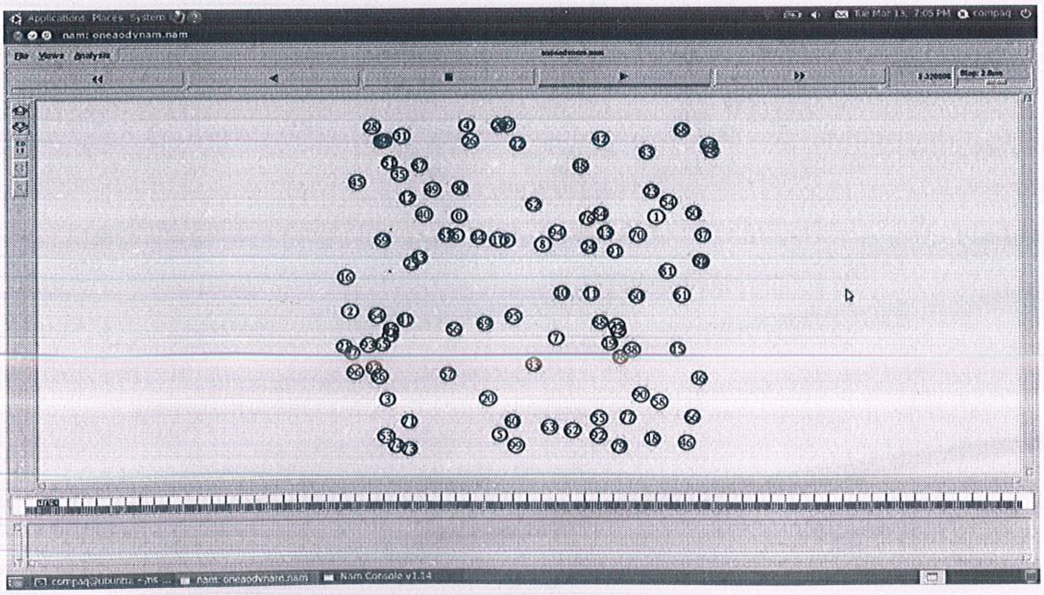
At time t=1



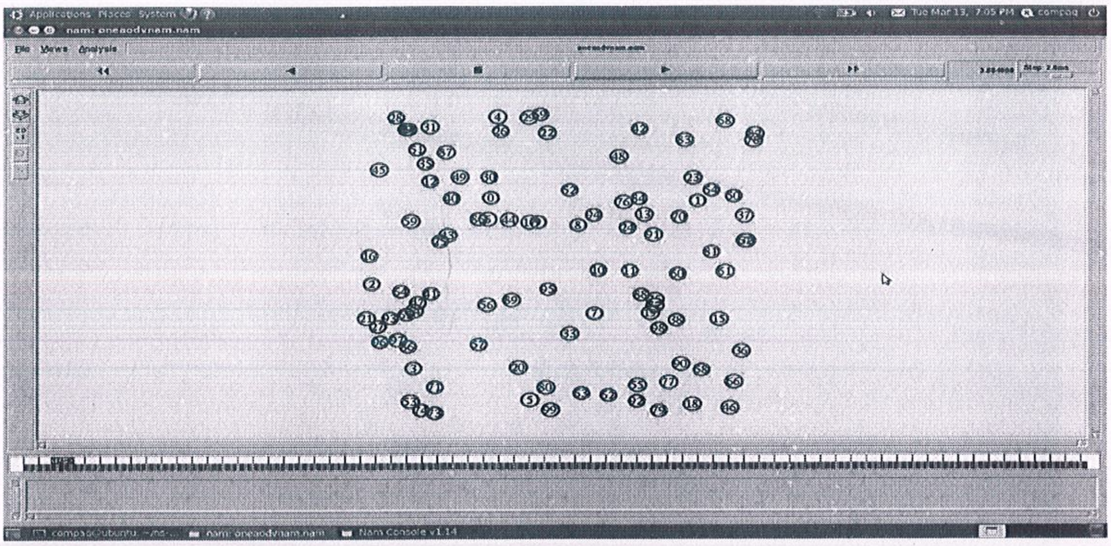
Since our **toplogy** is **random**, we will have the wireless network instead of the wired network and same tcl programming is applied to ensure the packet flow between the two desired nodes.

Snapshots of the wireless nodes at different instants of time and packets flowing between the two nodes passes through the intermediate nodes while mobility is achieved after some instant of time in the network.

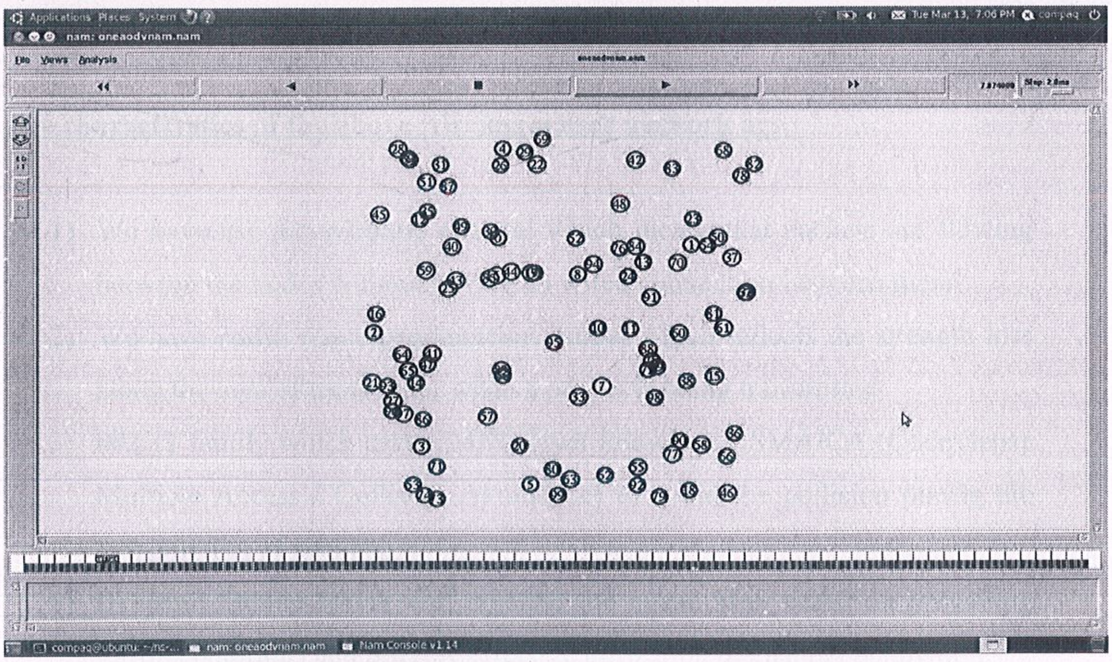
At time t=0



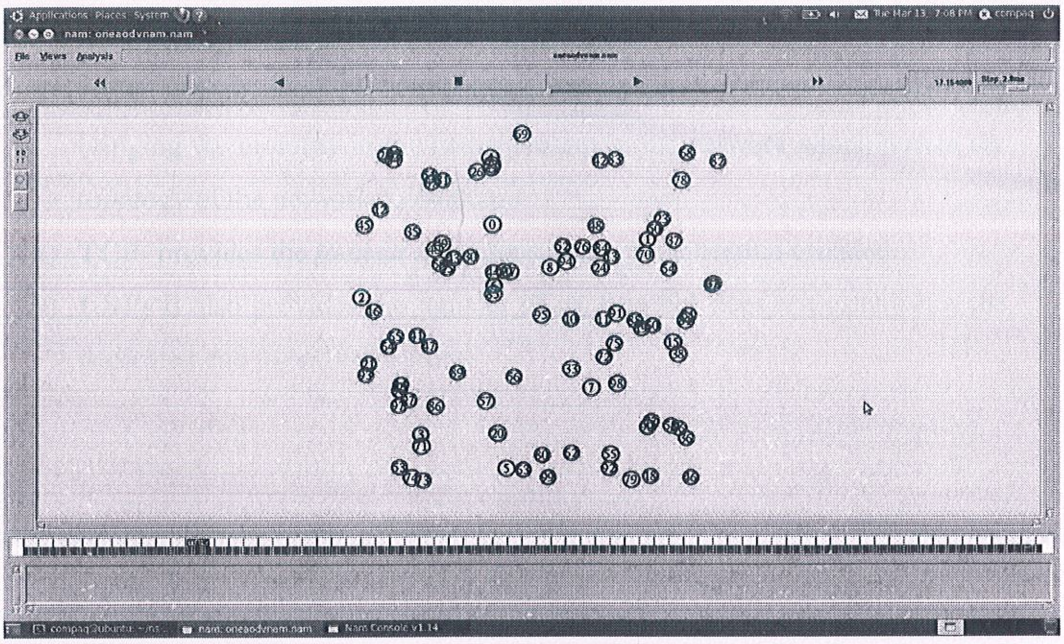
At time t=0.5



At time t=1



At time t=1.5



<u>The characterstics of the above wireless sensor network are :</u>

1) We have use the **wireless** channel which means that packets are flowing between the nodes without having an wired connection between them.
2) We have **radio wave propagation model** which reflects the strength loss along the area of transmitter when a packet is being transmitted.
3) **802.11** family which uses a MAC layer known as CSMA/CA (Code Sense Multiple Access / Collision Avoidance) to avoid the collision during the transmission of packets.
4) **Droptail Queue** which will regulate the queue size, based on prevailing traffic conditions and provide effective utilization of channel bandwidth.
5) **AODV (Adhoc On demand Distance Vector)** protocol which is a reactive protocol which establishes the route to destination point only on demand. Moreover, it sends the packets only on demand by the destination host.

6) **Cwnd (Congestion Window)**- Determines the no. of bytes that can be outstanding at a time and moreover it stops the link between two nodes from getting overloaded with too much traffic.

7) **GOD** object(**General Operations Descriptor)** Object : is used for bringing the mobility in the nodes present in the network which makes the topology of the network as random.

8) **TCP-** provides the transfer of packets and is a connection oriented.

9) **UDP-** It also provides the transfer of packets but does not establishes the dedicated end to end connection.

4.5 THE RSSI CONFIGURATION : THE SIMULATION SCENARIO

In Tcl code is a type of an antenna defined in a node configuration as

$ns node-config –antType Antenna/OmniAntenna

Omni means present everywhere, so OmniAntenna is is an antenna which radiates radio wave power uniformly in all directions in one plane.

In RSSI, we proceed by setting up the carrier sensitivity threshold and the received value of signal strength which is obviously less in magnitude as compared to the carrier sensitivity.

Let's assume two wireless nodes in certain coordinates at the beginning of the simulation. One of the nodes starts to transmit UDP and TCP packets through its wireless interface with given transmit power and antenna gain. We specify the propagation model as a Free Space Model and thresholds for carrier sense sensitivity and receive sensitivity. These thresholds define probability of successfully received packet. The receiver gain and transmitter gain are also associated with the carrier sensitivity and received sensitivity.

The **OTcl configuration** is as follows:

set opt(valif) Phy/WirelessPhy

$opt(valif) set RXThresh_ 2.2752e-7 #receive sensitivity threshold

threshold -m FreeSpace -r 0.95 30

which means that 95% packets is correctly received at the distance of 95m.

$opt(valif) set CXThresh_ 5.9852e-6 #carrier sense threshold

$opt(valif) set Gt_ 1 #transmitter antenna gain

$opt(valif) set Gr_ 1 #receiver antenna gain

$ns_ node-config -energyModel EnergyModel \ #node configuration

**-rxPower 0.281 **

**-txPower 0.281 **

**-phyType $opt(valif) **

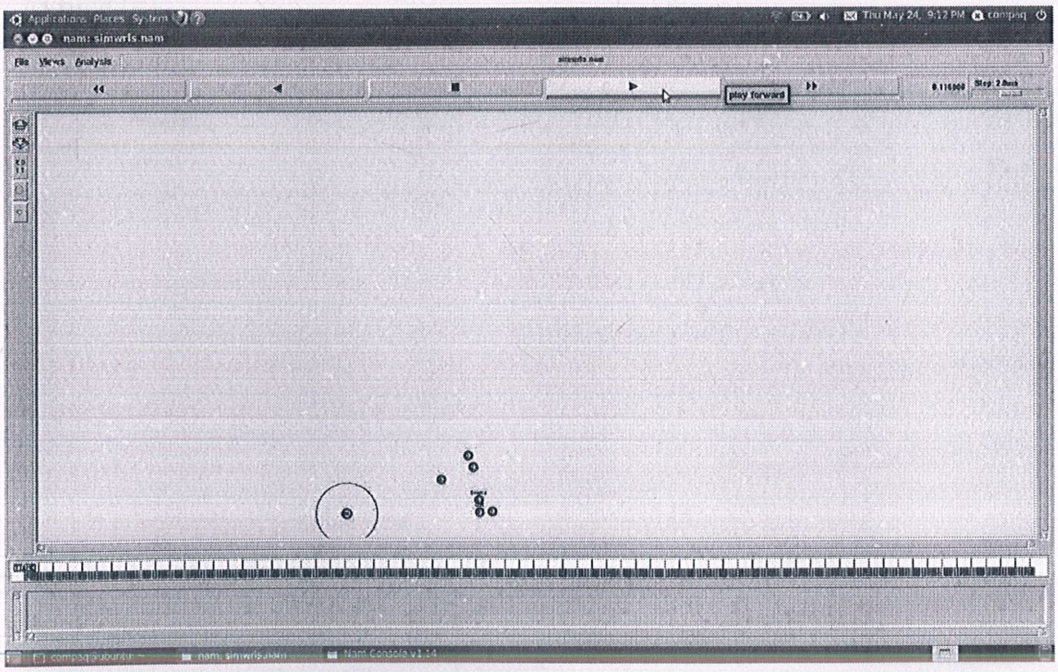
-propType Propagation/FreeSpace

4.6 THE AOA CONFIGURATION

To implement the AOA concept, what we have done is that we have created the wireless sensor network and we have assigned different colours to different nodes which are attained by the nodes after some instance of time when they start achieving their respective mobilities and hence we can say that the particular node is at this point in the network with the respective colour being achieved by it after some instance of time. Colours are obtained by writing **add-mark** and **delete-mark nodename colourname circle.**

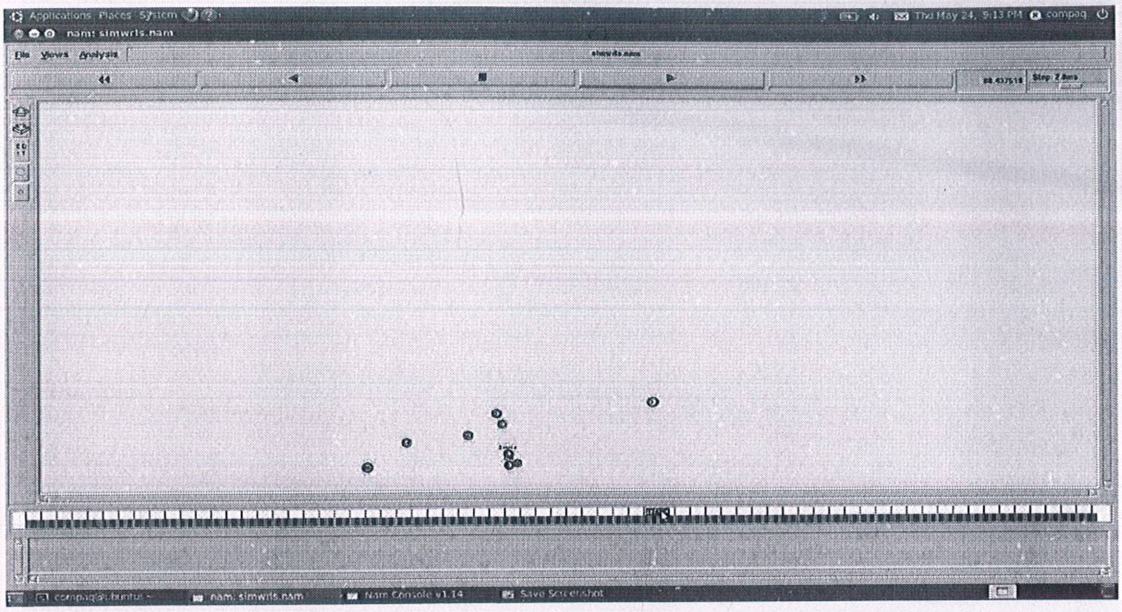
Snapshots of the wireless network showing the nodes attaining different colours after different instants of time

At time t=0



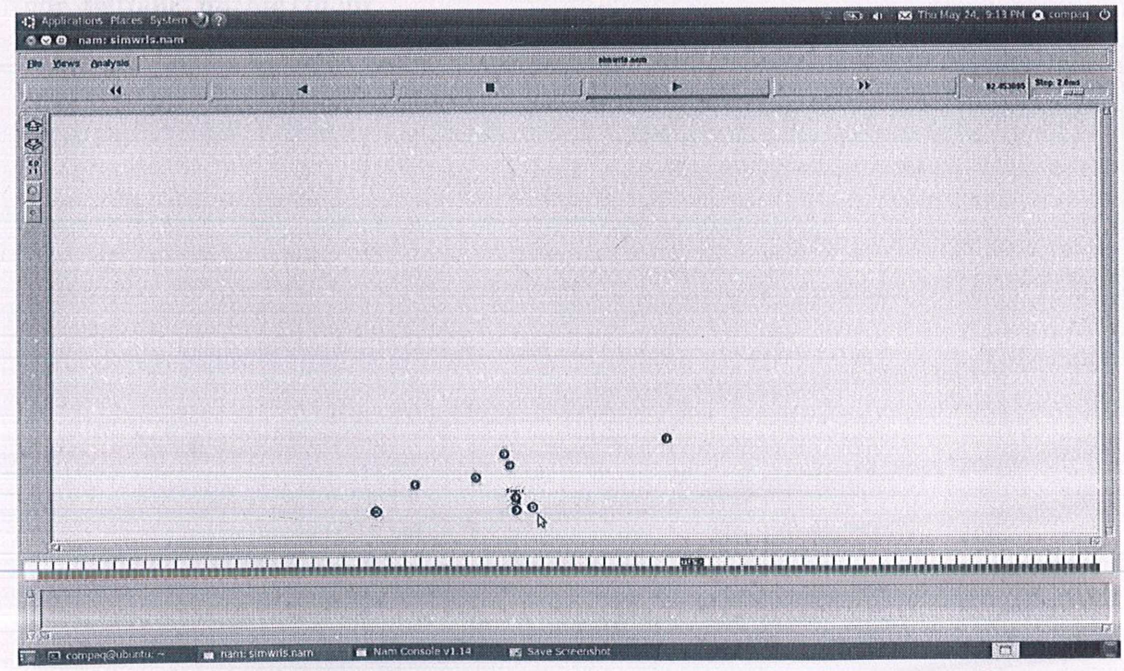
At time t=0.5

Node 1 attains yellow colour



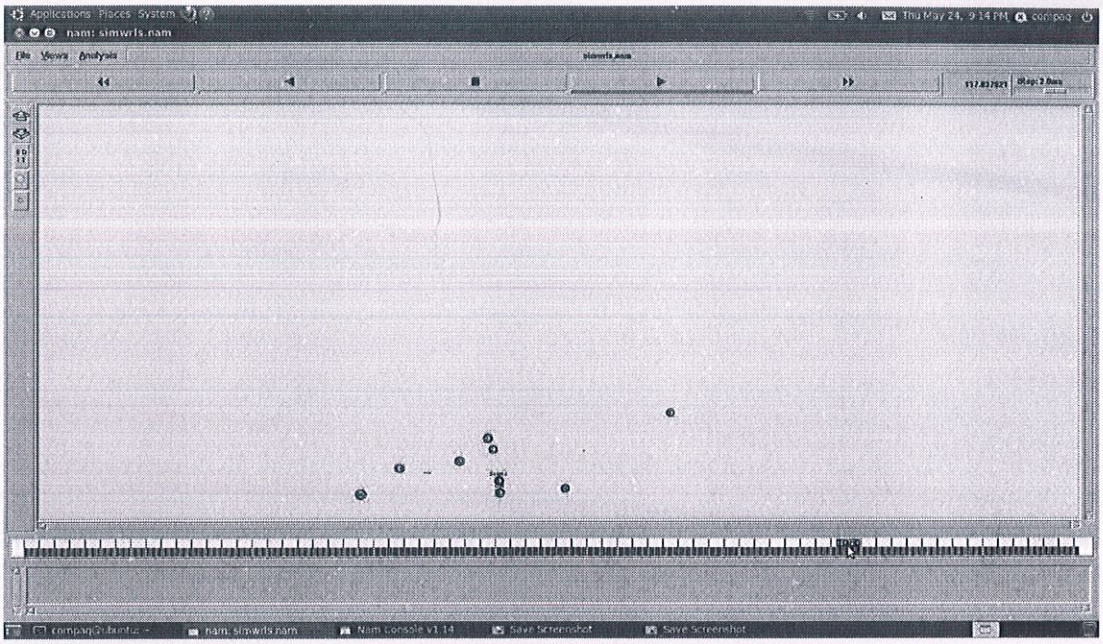
At time t=1

Node 2 attains green colour



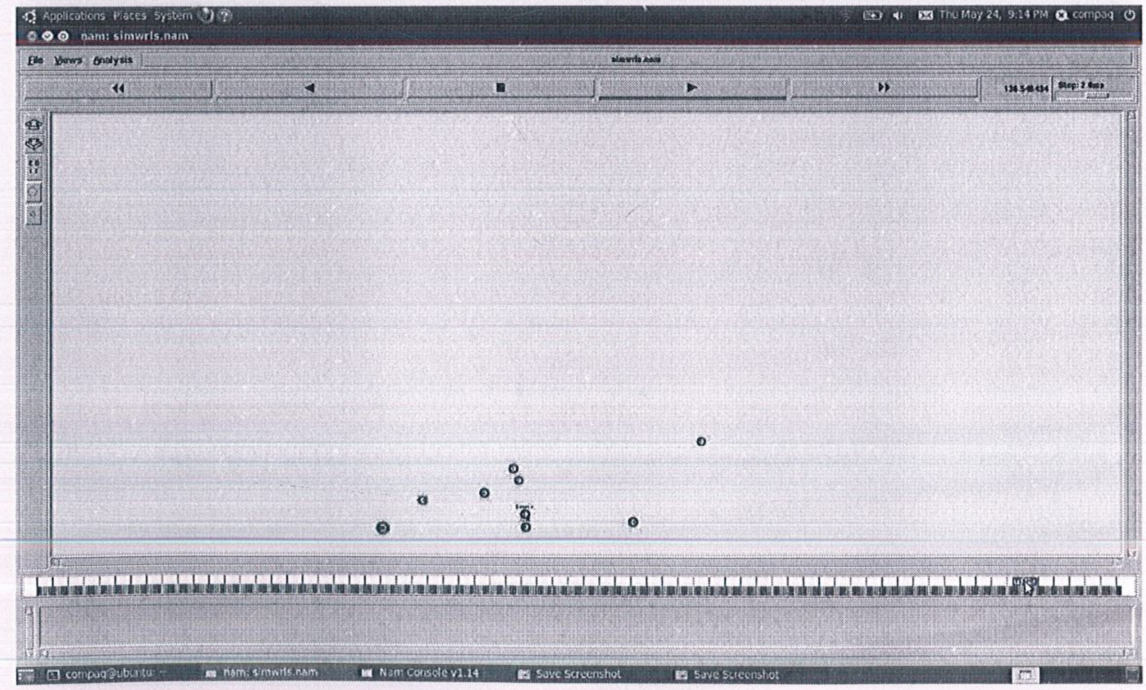
At time t= 1.5

Node 3 attains pink colour



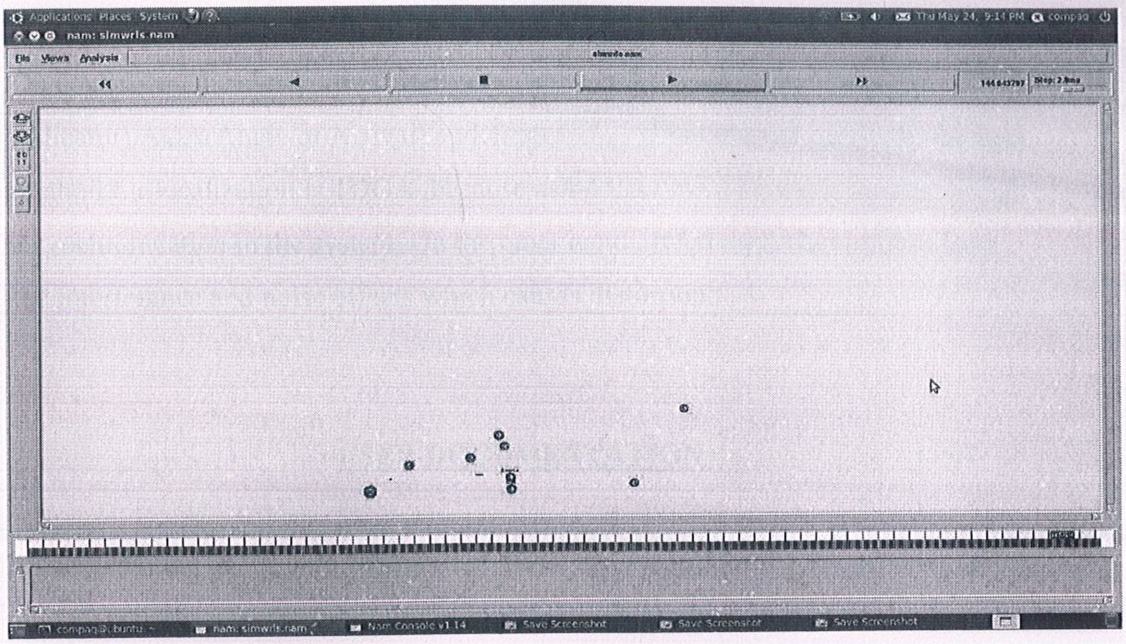
At time t=2

Node 4 attains purple colour



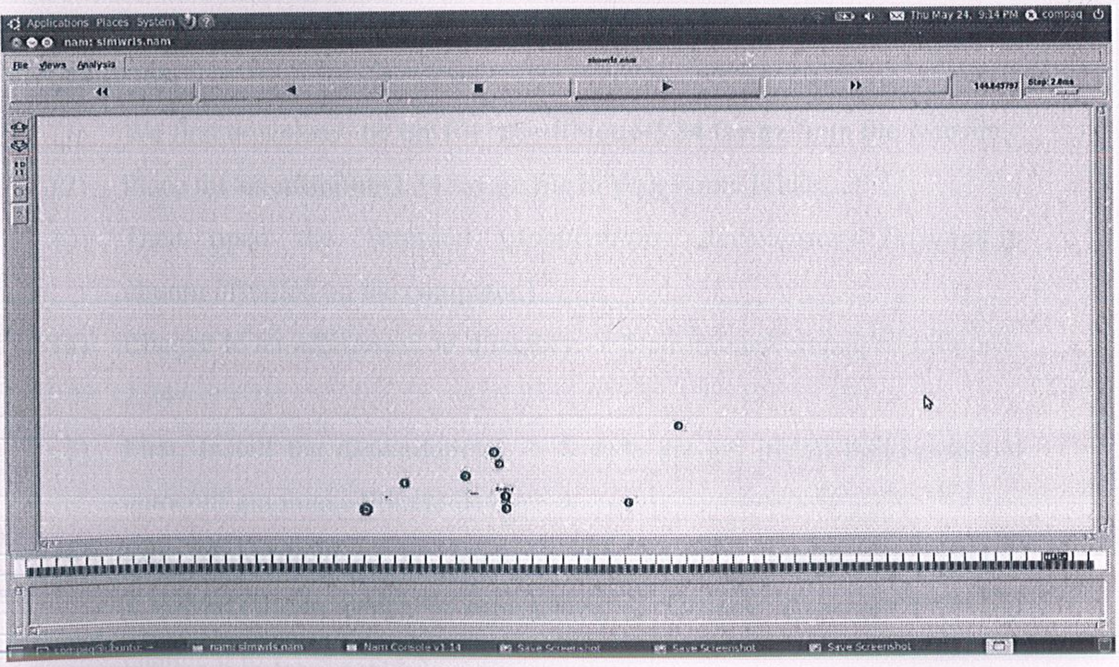
At time t=2.5

Node 5 attains white colour



At time t=3

Node 6 attains black colour



CHAPTER-5

]CONCLUSION

On comparing the above three methods i.e. RSSI(Received Signal Strength Indicator),AOA(Angle of Arrival) and TDOA(time difference of Arrrival), the best method for localization is TDOA because using the time difference concept brings a synchronization in the system. On the other hand , RSSI and AOA suffers from the interference and noise effects which causes distortion.

USER DOCUMENTATION

Documentation is the information that gives the description of the product to the user. It consists of the product manuals and online information etc.

In the user documentation, we ponders upon the tool or technology used to achieve the project implementation. In this tool used is Network Simulator (NS2) version 2.34.

To **install** this,

(1) We first download the **tar** file **ns-allinone-2.34.tar.gz** from the website.

(2) Place the **ns-allinone-2.34.tar.gz** file in your home folder.

(3) Then open the Terminal (*Applications–>Accessories–>Terminal* in ubuntu installed on the computer.)

(4) Change to ***ns-allinone-2.34*** directory -> $ cd /home/compaq/ns-allinone-2.34.

(5) First, Install the **dependencies** -> $ sudo apt-get install build-essential autoconf automake libxmu-dev gcc-4.

(6) Edit **Makefile.in** to be found at this location ns-allinone-2.34/otcl-1.13/Makefile.in which is represented as follows: **CC= @CC@** and change it to **CC= gcc-4.3**

(7) Begin the NS2 installation: **$ sudo su ./install**

(8) Once the installation is successful ,then we need to add the path information to the file **~./bashrc $ gedit ~./bashrc**

(9) For changes to take place immediately , **type $ source ~./bashrc**

(10) Then type ns to see % and **nam** to see **nam** window

REFERENCES

[1] Amitangshu Pal: **Localization Algorithms in Wireless sensor networks: Current Approaches and Wireless Sensor Networks.**

[2] Azzedine Boukerche, A.B.F Oliveara, Eduardo F. Nakamura : **Localization System for Wireless Sensor Networks**

[3] Azzedine Boukerche, University of Ottawa: **Localization systems for wireless sensor networks**

[4] J.R Douceur: **The sybil attack. In IPITS '01 Revised papers from the first International Workshop on Peer to Peer systems, pages 251-260,2002**

[5] Mohamed Youssef, Nasir El-Shiemy: **Robust Cooperative Localization Technique for Wireless Sensor Networks**

[6] Murat Demirbas, Youngwhan Song: **An RSSI- Based scheme for Sybil Attack Detection in wireless sensor networks**

[7] Murat Uney, Mujdat Cetin: **Target Localization in sensor networks using factor graphs**

[8] Patrik Moravek, Dan Komosny: **NS2 simulator capabilities in Nodes localization in Wireless Sensor networks.**

[9] Prasan Kumar Sahoo, Shi-Yao Lin, I-Shyan Hwang : **A Distributed Localization Scheme for Wireless Sensor Networks**

[10] M. McGuire, K.N. Platanotis: **Location of mobile terminals using time measurements and Survey points" IEEE transactions on Vehicular Technology.**

[11] Rong Peng, Mihial L. Sichitiu : **Angle of arrival localization for wireless sensor networks**

[12] Zenon Chaczko, Ryszard Klempous, Jan Nikodem, Michal Nikodem: **Methods of Sensors Localization in Wireless Sensor Networks**

[13] Zenon Chaczko, Ryszard Klempous, Jan Nikodem, Micheal Nikodem: **Methods of Sensor localization in wireless sensor networks**

BRIEF BIO-DATA OF PROJECT GROUP MEMBERS

Priya Puri (081424)

Exam	Name of School/University	Year of Passing	CGPA/ Percentage Obtained
10th (C.B.S.E.)	Bhatnagar International School, Vasant Kunj ,New Delhi	2006	89.8%
12th (C.B.S.E.)	D.A.V Public School, New Shimla, H.P	2008	83.2%
B.Tech (I.T.)	Jaypee university of information technology,Waknaghat ,H.P	2012	6.6 (till 7th semester)

Anshul Dhingra (081434)

Exam	Name of School/University	Year of Passing	CGPA/ Percentage Obtained
10th (C.B.S.E.)	Kulachi Hansraj Model School, Ashok Vihar ,New Delhi	2006	89.0%
12th (C.B.S.E.)	Kulachi Hansraj Model School, Ashok Vihar, New Delhi	2008	80.4%
B.Tech (I.T.)	Jaypee university of information technology, Waknaghat ,H.P	2012	7.8 (till 7th semester)

Anubhuti Verma (081452)

Exam	Name of School/University	Year of Passing	CGPA/ Percentage Obtained
10th (C.B.S.E.)	A.S.N. Senior Secondary School, Delhi	2005	93.0%
12th (C.B.S.E.)	A.S.N. Senior Secondary School, Delhi	2007	91.2%
B.Tech (I.T.)	Jaypee University of Information Technology, Solan	2012	8.1 (till 7th semester)

Deepali Gupta(081459)

Exam	Name of School/University	Year of Passing	CGPA/ Percentage Obtained
10th (C.B.S.E.)	D.A.V Public School	2006	95%
12th (C.B.S.E.)	D.A.V Public School	2008	90.4%
B.Tech (I.T.)	Jaypee University of Information Technology, Solan	2012	8.1 (till 7th semester)