Improving accuracy for RSSI based localisation

By
Qazi Mohsin Ijaz
00000171702

Supervisor
Dr. Arsalan Ahmad
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(August 2020)

Approval

It is certified that the contents and form of the thesis entitled **Improvin-gaccuracy for RSSI based localisation** submitted by **Qazi Mohsin Ijaz** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Arsalan Ahmad**

Signature: ______________

Date: 15-08-2020

Committee Member1: **Dr. Nadeem Ahmad**

Signature: ______________

Date: 15th August 2020

Committee Member2: **Dr. Asad Waqar Malik**

Signature: ______________

Date: 15-08-2020

Committee Member3: **Dr. Muazzam Khattak**

Signature: ______________

Date: ______________

Dedication

To my parents,teachers and friends
Without whom this success would not be possible.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Qazi Mohsin Ijaz**

Signature:

Acknowledgment

I am using this moment to show my appreciation to everyone who aided me out through entirety of my research. I am very grateful for such valuable assistance and friendly advises . I am wholeheartedly thankful to my Supervisor and Dept. of Computing SEECS for providing such an environment.

Qazi Mohsin Ijaz

Table of Contents

List of Figures

List of Tables

List of Symbols

Abbreviations

RSSI	Received Signal Strength Indicator
NS2	Network Simulator Version 2
WSN	Wireless sensor network
UN	Unknown node
MN	Mobile Node
FN	Fixed node
BN	Beacon Node
GPS	Global Positioning System

Abstract

Wireless technology has seen a sharp inclination towards low power wireless motes, micro sensor chips and microprocessors which are not only energy efficient but also have significant improvements in distributed signal processing, ad-hoc networking and pervasive computing. These low-powered multi-functional sensor nodes are capable of integrating vast information, faster data computation, optimal storage and efficient wireless communication. In other words wireless sensor networks are low cost, low power, short ranged, small size nodes that are deployed for data collection specifically at inaccessible places for sensing and processing information.

RSSI is a popular technique to measure the pathloss for wireless environments. Previous researches has utilised RSSI values to estimate the distance and subsequently do the localisation for unknown nodes in the topology. This research will focus on improving the localisation accuracy of RSSI based systems incorporating the estimation of pathloss coefficients. The proposed solution will be implemented by modifying the pre-existing 802.11 mac file in C++ and the scenario will be simulated in Network Simulator-2 (NS-2) through TCL scrips.

Chapter 1

Introduction

1.1 Motivation

The modern world is built on speedy communication. People need quick and reliable information about their health, the state of their environment, and current safety. Wireless Sensor Networks (WSNs) are a type of network that has been developed as an answer to this need for control. To improve the information and reliability of state of our environment tracking the location of objects and intruders in indoor, underground and outdoor environments has been the focal point of researchers in the past years with the rise in terrorism across the globe and unreliability of Global Positioning System (GPS) in closed/indoor environment, WSNs can potentially be used in effective reconnaissance, search and rescue operations, routing in deep jungles or mines or keeping track of target in unknown territories. The current solution require external hardware and software to achieve to track the objects or intruders, the main objective of this thesis is to calculate the distance of unknown nodes from known nodes and find out its estimated location in the topology using its RSSI value.

1.2 Problem Definition

RSSI is error prone, the error is transferred at close distance of about 1 meter, and the distance error varies between 0.5-2 meters. At 20 meters or more, the distance error may range from 10-40 meters. The error is caused by noise on the signal measurement, as well as signal reflections, humidity, temperature and obstacles [1]. In this thesis our goal is to develop an algorithm that does not require any modification to the existing wireless sensor motes, or require extra hardware and can easily perform with reduced communication delays.

While keeping all these factors in focus, this algorithm should effectively localize the unknown nodes.

1.3 Objectives and Research Goals

Our proposed solution is based on modifying the preexisting Mac 802.11 file to improve the localization thus eliminating the need of external hardware or software. As RSSI is error prone, the error is transferred to calculating distance which in turn is transferred to finding the location of the nodes. The proposed solution will be divided into two parts.

- Reduce error in calculating distance between Nodes
- Reduce error in finding the location of unknown Node

1.4 Thesis Organization

The work has been organized as follows. Chapter 2 covers the basic background information about the Wireless Sensor Networks. Chapter 3 covers the literature review and limitations. Chapter 4 explains the methodology used in the proposed mechanism. Chapter 5 unveils the evaluation of the results. Chapter 6 discusses the conclusion and future work.

Chapter 2

Background Information

Wireless sensor network (WSN) is a large group of low powered distributed sensors for monitoring and recording the physical environmental conditions such as temperature, sound, pollution levels, humidity and wind etc. They are similar to wireless ad-hoc networks as they too depend on wireless channels for communication. WSN is built of a few to thousands of connected nodes or motes, following the OSI architecture which has 5 layers and 3 cross layers. The 3 cross layers are for power, mobility and task management whilst the 5 layers are application layer, transport layer, network layer, data link layer, physical layer. The characteristics of WSN are [2].

- Resilience – The ability to cope with faulty nodes
- Energy Efficient - To run on lower power for a long period of time
- Mobility - The ability to move anywhere in the area with coverage
- Withstand Harsh Environmental Conditions
- Can accommodate new devices without the need to change the network infrastructure

Like any other network, WSN is also not free of its fair share of problems, which causes significant problems to the network if the conditions are not favorable.

- Fault tolerance: The environment conditions in the outside world cannot be controlled which makes the sensor nodes to become faulty and their readings are less and less reliable with the passage of time. [4]
- Traditional problems of wireless channels: Since the nodes communicate with each other in a traditional wireless channel, WSNs also carry these problems of high error rate, fading, noise, interference etc

- Geographical Landscape: The landscape causes numerous problems of scalability and coverage. Plain areas are comparatively easier to cover but hilly areas and underground areas require thousands of sensors to provide coverage. Scalability in non-plain area is a fundamental problem in WSN as it can affect the correctness of the data collected.
- Energy: The sensors while being energy efficient still operate on a battery with limited capacity so batteries have to be recharged or replaced.

WSN can be used in numerous applications such as smoke and fire detection in forests and inside buildings; greenhouse monitoring to keep the temperature and humidity level constant; target tracking, as they can track something as small as another sensor to something as big as a train; monitoring health conditions of people and animal especially on big farms with thousands of animals, the diseases are detected in the earliest stages to avoid widespread among other animals [5].Localization or target tracking have been the main focus of researchers and military as it can help to identify unknown devices or find intruders in the area with the help of Received Signal Strength Indicator (RSSI) of the target/unknown node in both indoors and outdoors. RSSI is an estimated measure of power level a device is receiving from an access point or router in a wireless environment. As the distance increases, the signal gets weaker and the wireless data rates get slower, causing a low data throughput. RSSI vaulue information is often not available to the receiving devices, however the IEEE 802.11 protocols can be used to make this information available. The range of RSSI is from -50dBm (maximum power and ideal conditions for connectivity) to -102dBm (for minimum reception) [3]. The higher the value of RSSI, the stronger the connection and the least distance away from access point or beacon node. There is no standardized way to measure RSSI and the 802.11 protocols do not define a relationship between RSSI and power values received, each chipset maker has their own standards and relationship between RSSI and power and define their accuracy and range.Figure 2.1 shows different values of RSSI and signal strength for those values[5].

RSSI can be used to measure the distance between two nodes as the bigger the difference the lesser the RSSI value. Whilst it seems simple that RSSI is inversely proportional to distance but noise, obstruction and type of antenna and other environmental conditions greatly affect the results[6]. Once the distance is estimated then the localization algorithm can be used to get an estimated location as shown in figure 2.2 However, there is a need to develop an efficient localization algorithm that can efficiently use the RSSI value to work with a WSN based sensing infrastructure in indoor and outdoor area.

RSSI	Signal Strength
> -70 dBm	Excellent
-70 dBm to -85 dBm	Good
-86 dBm to -100 dBm	Fair
< -100 dBm	Poor
-110 dBm	No signal

Figure 2.1: Relationship between RSSI and Signal Strength

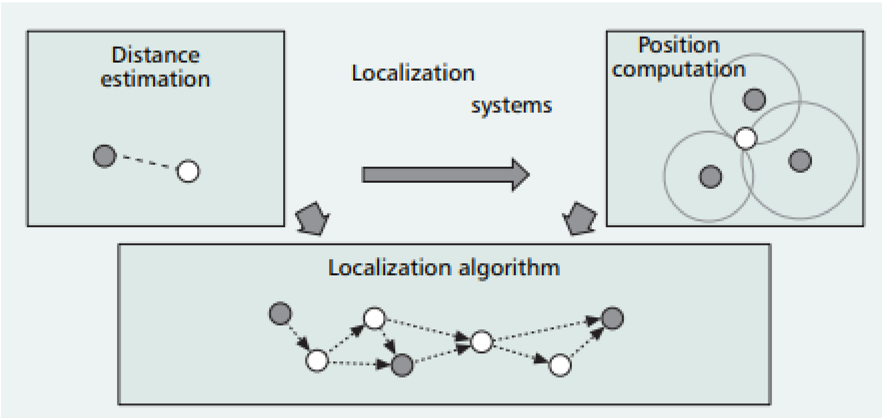

Figure 2.2: Localization through RSSI

The proposed techniques addresses the challenges of localization in both indoor and outdoor areas without the need of external hardware or changes to current WSN motes. The aim of the proposed technique is to minimize the localization error

Chapter 3

Literature Review

R. Dalce, et al. in their paper proposed the dv-hop algorithm to find the location of the unknown node [7]. The algorithm uses information from the neighbor node to find such as the hop count along with their coordinates to find the distance between two nodes. This way the unknown nodes know how many hops away it is from a certain node as the beacon has previous node information and hopcounts from the starting node. With this information each node can calculate their shortest distance to a certain node and so can the unknown node to find its location.
In [8] the authors, uses RSSI information to localize the unknown node in ZigBee, they use the trilateration method to calculate the distance of the node, they increase the antenna power of the node to avoid less interference from the environment such as humidity, temperature and electromagnetic pulses. They too used the dv hop algorithm to keep count of the hops and concluded that with increase in RSSI error the localization error increases too, hence the power antenna power was increased to reduce localization error.
In [9] the authors improve the localization in ZigBee using RSSI by placing fixed nodes at every 50 m distance and remoted nodes were carried by the miners. The fixed nodes have a big display so miners can see all the information on screen such as their location, temperature, humidity. Their solution, however very accurate but requires external hardware and software to achieve the goals with high energy consumption as the fixed nodes requires 93mA current and 24V. The remoted nodes, ran on lithium battery which lasted for 8 hours. Their localization error was reduced to 2.2m.
Jian et al. proposed a new weighted centroid localization algorithm in mine WSN [10], the algorithm uses RSSI algorithm for distance measurements along with the weighted centroid algorithm to find the location of the unknown node. In case of a damage to a node some security measures were

taken to ensure the node next to the damaged node will increase its transmission power to be able to communicate with the node after the damaged node. Due to the harsh underground environment, such as surface roughness and sloping of the tunnel walls, the wireless signals face scattering attenuation. The mobile node records the average RSSI from fixed nodes and selects three highest RSSI values to calculate the distance with a minimum error of 4.87m.

Shaoguo Xie [12] proposed another solution where the complexities related with pathloss and distance between the beacon nodes are completely ignored. It only uses the RSSI information to find the location of mobile node. Shaoguo Xie [12] reduced the localization error to 3.75m, their solution is faster and efficient then weighted centroid algorithm as it doesn't need to calculate distance and pathloss between the nodes, thus saving both time and energy.

Fan, Hongbo, et al proposed a new solution of weighted centroid algorithm to improve the RSSI range [13]. In their work, they used both the distance and RSSI and since the distance is already known, it removes the errors from the distance between the beacon nodes and the mobile node then selects the reciprocal of the distance as the weight. In order to improve the RSSI measurements, their solution with improved the localization by 10 percent in comparison with the traditional centroid algorithm.

In [14] the authors uses external radiolocation device that runs on IEEE 802.15.4 standard, it uses location estimation capability via RSSI. Unknown nodes collect data from all the nearby devices to calculate its position and sends it out to a monitoring application. Their scheme is used in a hybrid environment of indoor and outdoor. The device has 2 phases i) Deterministic phase ii) Probabilistic phase. In first phase the RSSI values of the referenced nodes are collected and in second, the distance and position is calculated with an error rate of 2.48m.

In [15] the authors suggested on improving the dv hop algorithm by using average hops by approximating the distance in a weighted average algorithm. In a typical DV hop the unknown node accepts hop count information from the recent node only but in weighted average dv hop the unknown node accepts information from all the beacon nodes in the network and calculates the hops by taking the average of maximum hops and minimum hops. The average hop count is broadcasted to the network and by using the triangulation method the distance and position is calculated. In [15] improving the dv hop is also suggested by selecting 1 to 3 anchor nodes with hop of count of 1, if there is no anchor node with hop count of 1 then the original unaltered dv hop is used. If there is only one hop selected, position triangulation is used on the RSSI value to calculate the distance, If 2 nodes are selected then the

jump distance between the 2 selected nodes are calculated and position of unknown node is determined. In final case, when 3 anchor nodes are selected the distance is calculated directly by RSSI values. One of the limitation of this algorithm is that anchor nodes must be distributed evenly in a limited region.
All the aforementioned algorithms are using RSSI values to localize unknown nodes whether in WSN or ZigBee. Some techniques uses external hardware and software to improve the accuracy while others improve on the preexisting algorithm to decrease the localization errors in indoors, outdoors and mines environment. But issues with such techniques is that they require external hardware and a lot of power consumption which is not ideal for WSN.

Chapter 4

Methodology

The beacon nodes sent beacon messages with their id and coordinates, when, another node receives these message, the algorithm checks whether it's a fixed node or an unknown mobile node. The algorithm runs on all nodes regardless of their nature i.e. fixed or mbile. In case of a fixed node, the algorithm calculates the distance and stores the coordinates of the sender. In second case if the receiver is a mobile node, that doesn't know its coordinates, the receiver starts calculating RSSI for every message received and stores them along with sender's node id and coordinates. Furthermore, it uses the calculated RSSI value to calculate the distance from it, and stores it with the previously stored RSSI values and coordinates. Once the distance is calculated the array is sorted according to least distance. The least three values of distance are selected to calculate the coordinates of the unknown node to find it the X and Y coordinate of the mobile node. The rest of the chapter describes the detailed description of each step of research methodology that we are using in our thesis. Solution is also explained in detail.

4.1 Network Model

The simplest scenario in the simulation is based in indoor environment, the location of the four fixed nodes are already known and a fifth node is introduced to the network whose coordinates are unknown and the system has to localize it. However for further tests the area was increased to 50m x 50m with up to 45 fixed nodes.
Fig 4.1 shows the simplest scenario, when unknown node enters the network, the surrounding in range fixed beacon nodes sends it messages so the unknown node can localise itself through the beacon messages received and send out its coordinates in the future messages to the fixed nodes or other

mobile nodes.

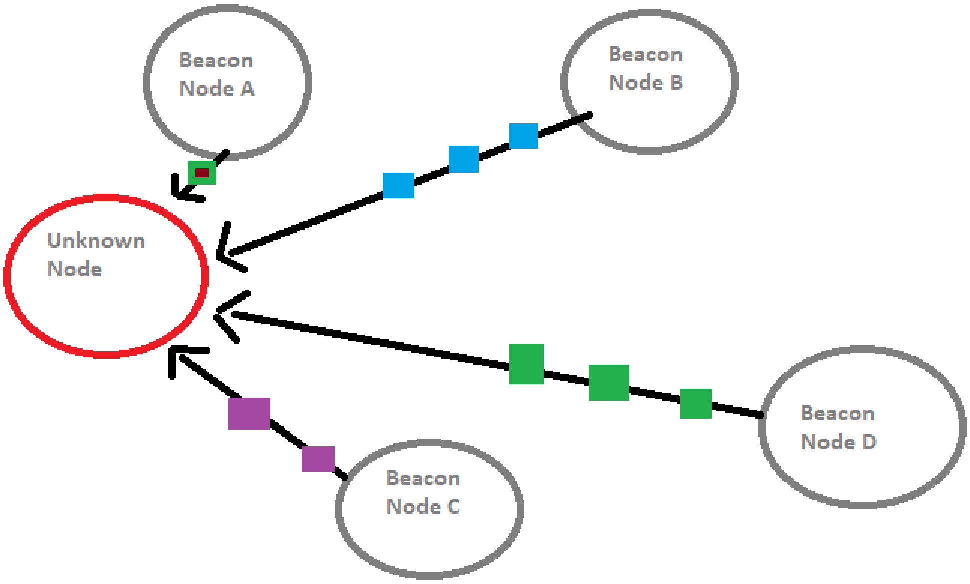

Figure 4.1: Proposed Network Model

4.2 RSSI Calculation

The fixed or beacon nodes are self-aware and when one beacon node receives a message from the other beacon node, it calculates its distance from the received beacon node as the beacon message includes the X,Y coordinates of the sender with the following equation

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \tag{4.1}$$

Where d is the distance between 2 beacon nodes A and B, XA and YA are X and Y coordinates of the beacon node A and XB and YB are the X,Y coordinates of the beacon node B.
Once every node calculates the distance for other nodes. these distances are stored so every fixed node in the network knows the distance and coordinates of its neighbour's fixed nodes coordinates. After determining each other's distances, beacon nodes calculate the RSSI values in dBm with Eq. 4.2

$$RSSI = -\left(10\log_{10}\left(P_r * 1000\right)\right) \tag{4.2}$$

Where P_r is the received power of signal by the unknown node and RSSI is received signal strength index for the fixed node.
After calculating distance and RSSI it is imperative that we find the pathloss factor, the pathloss provdies the information of signal decaying and attenuation.

$$n = \frac{-(RSSI - A)}{(10 \log_{10} d)} \tag{4.3}$$

Where n is the pathloss factor, d is distance from 4.1 and A is the RSSI at reference distance of 1m with a fixed value of -22.628.
When an unknown node enters the network, the fixed nodes sends it beacon frames with transmitting power of 100 milliwatt. The becaon frames include the information of $nodeID$, pathloss factor n along their X and Y coordinates. The mobile node calculates the RSSI vaules for all the in-range beacon nodes from the received power as by defualt the RSSI vaulues are not provided with Eq. 4.2

Algorithm 4.1 runs on the unknown node after receiving beacon frame from the fixed node, it gives the basic idea of the RSSI calculation.

Algorithm 4.1 – RSSI Calculation

$UN \leftarrow$ Unknown Node
$FN \leftarrow$ Fixed Node
X_A X coordinate of Node A
Y_A Y coordinate of Node A
$flag \leftarrow 0$

while ($flag = 0$) **do**
    FN sends beacon frames for interval $timer$
    if ($FN.Found$) **then**
        Extract X_A,Y_A coordinates
        $distAB$ = using eq. 4.1
        Calculate RSSI using eq. 4.2
        Calculate pathloss using eq. 4.3
        store pathloss
        Advertise pathloss in future messages
    else
        Calculate RSSI for the fixed node using Eq. 4.2
        goto Algorithm 4.2
    end if
end while

4.3 Pathloss Estimation

In previous phase, pathloss was successfully calculated for all the fixed nodes. However, the distance between fixed node and mobile node is not known, therefore it is has to be adjusted in the following ways.

- Average Pathloss: The mobile node uses average pathloss for calculating its distance from the fixed as shown in eq. 4.4

$$n_{avg} = \frac{(n_1 + n_2 + n_3)}{3} \tag{4.4}$$

  Where n_{avg} is the average pathloss of the 3 fixed nodes and n_1, n_2, and n_3 are their respective pathlosses.

- Highest pathloss: The mobile node uses the pathloss n of the fixed node with the highest value of RSSI.

- Pathloss of Sender: The mobile node can use the pathloss received in the beacon frame from the fixed node. Each node will have a different pathloss.

However as the distance between unknown node and fixed nodes is not known so taking highest pathloss is not feasible, for our solution we considered the average pathloss.

4.4 Distance Calculation

After calculating RSSI values and taking the average pathloss for each beacon node in the range of the unknown node, the unknown node stores the received RSSI values in an array and selects the three highest values. Once the RSSI value is selected it calculates distance by the following formula.

$$D = 10^{\frac{(RSSI-A)}{(10*n)}} \tag{4.5}$$

Where D is the distance between the fixed node and unknown node,RSSI value is from eq 4.2 and A is the RSSI constant at 1m distance, whilst n is the pathloss factor of the sender node from eq. 4.3
Algorithm 4.2 runs on the unknown node and shows the basic working of distance calculation between unknown and known nodes.

Algorithm 4.2 – Distance Estimation

$UN \leftarrow$ Unknown Node
$FN \leftarrow$ Fixed Node
$RSSI \leftarrow$ RSSI from 4.1
if $(RSSI\_Found)$ **then**
    Calculate distance between $UN.Dist$ and $FN.Dist$
    Store distance Sort distance array
    select 3 least distances
    goto Algorithm 4.3
else
    goto Algorithm 4.1
end if

4.5 Position Estimation

In previous phase, the unknown node successfully calculates the distance with 3 of nearest beacon nodes, to find the coordinates of unknown node, two methods were selected

- Vanilla Triangulation
- Weighted Centroid Triangulation

4.5.1 Vanilla Triangulation

In vanilla triangulation method, the distances of the fixed nodes do not matter in a sense that they all play equal role in the estimations of the position. To find the X-axis coordinate, the 3 fixed nodes X coordinates are divided by their respective distances to mobile weight and summed up, similarly for Y-axis coordinate of mobile node, their Y coordinates are divided by their respective distances as shown in Eq. 4.6 and Eq.4.7

$$x_U = \left(\frac{x_1}{d_1} + \frac{x_2}{d_2} + \frac{x_3}{d_3} \right) \tag{4.6}$$

The X coordinate of the unknown node is represented by x_U, x_1 , x_2 and x_3 are the X coordinates of the 3 selected nodes, d_1 , d_2 and d_3 are their respective distances.

$$y_U = \left(\frac{y_1}{d_1} + \frac{y_2}{d_2} + \frac{y_3}{d_3} \right) \tag{4.7}$$

The X coordinate of the unknown node is represented by y_U, y_1 , y_2 and y_3 are the X coordinates of the 3 selected nodes, d_1 , d_2 and d_3 are their respective distances.

4.5.2 Weighted Centroid Triangulation

In weighted centroid triangulation, the the closer the fixed node is the more influence it has on the final outcome. Before estimating the position, it is important to assign the weights, which is reciprocal of the distance so that the nearest distance has the most influence on position estimation of the mobile node as shown in Eq. 4.8.

The weights to distances are given to reduce the position estimation error as without the weights, each node has the same influence on the localization process which results in huge errors. By giving them weights, the closer the fixed node is the more influence it has on the final outcome.

$$w = \frac{1}{d} \tag{4.8}$$

Once the weights are given, we use eq. 4.9 for the X-axis coordinate and eq. 4.10 for the Y-axis coordinate.

$$x_U = \frac{\left(\frac{x_1}{d_1} + \frac{x_2}{d_2} + \frac{x_3}{d_3}\right)}{w_1 + w_2 + w_3} \tag{4.9}$$

The X coordinate of the unknown node is represented by x_U, x_1 , x_2 and x_3 are the X coordinates of the 3 selected nodes, d_1 , d_2 and d_3 are their respective distances while w_1 , w_2 and w_3 are the weights given to the distances in eq 4.8

$$y_U = \frac{\left(\frac{y_1}{d_1} + \frac{y_2}{d_2} + \frac{y_3}{d_3}\right)}{w_1 + w_2 + w_3} \tag{4.10}$$

The Y coordinate of the unknown node is represented by y_U, y_1 , y_2 and y_3 are the X coordinates of the 3 selected nodes, d_1 , d_2 and d_3 are their respective distances while w_1 , w_2 and w_3 are the weights given to the distances eq 4.8
Algorithm 4.3 is run on the mobile node, and gives the basic idea of position estimation.

Algorithm 4.3 – Position Estimation

```
UN ← Unknown Node
D_A ← Distance from fixed Node A
D_B ← Distance from fixed Node B
D_C ← Distance from fixed Node C
W.A= Perform Eq.4.8 on D_A
W.B= Perform Eq.4.8 on D_B
W.c= Perform Eq.4.8 on D_C
UN_X = Calculate from Eq.4.9
UN_Y = Calclate from Eq. 4.10
return UN_X and UN_Y
```

Chapter 5

Results & Discussion

5.1 System Specification

The experiments for the evaluation of the proposed techniques are performed on system with the specification given in Table 5.1

Table 5.1: System specification

Name	Specification
CPU	Intel Core (TM) i5-7200 CPU 2.7 GHz
RAM	8 GB - DDR4 2133 MHz
GPU	2 GB - GeForce 920 MX
OS	Ubuntu 16.04.03 LTS
Simulator	NS2

In this section, we have evaluated the proposed mechanism on NS2 with the updated Mac 802.11 file. platform and is compared with the tradition centroid solution. NS-2 is an object oriented simulator, the language used is C++, with TCL script for front-end network design. The TCL script is used for configuring the network, a traffic pattern to declare data transmission from source to destination and a mobility pattern to specify the movement of the nodes inside the grid .

5.2 Simulation Scenario

The solution is simulated in a 50x50m area, where beacon nodes are fixed, self-aware of their coordinates and transmitting power is 100mW. The un-

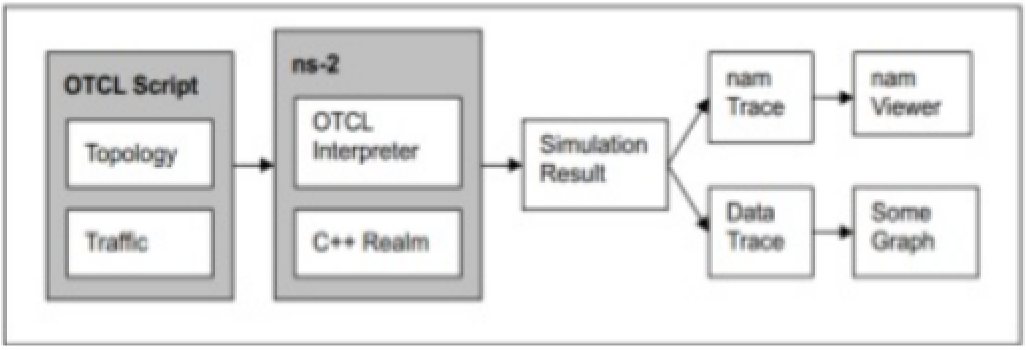

Figure 5.1: Architecture of Network Simulator

known node is placed randomly on the topology and moves around in a straight line at speed of 2m/s. The density of fixed nodes can vary from 4 to 50 depending on the size of area, in this particular case it was set to 45.

5.3 Performance metrics

The following metrics are used to evaluate the scenario for the given two algorithms.

- Distance Estimation error between mobile and fixed node
- Position Estimation error of the mobile node.

5.4 Results

Figure 5.2 shows the comparison of number of nodes and distance estimation error with actual distance, as the number of node increases, the error decreases until number of nodes reach 40. Afterwards, the distance error doesn't decrease. The green line shows the actual distance between the closest fixed node and mobile node, the red line shows the estimated distance by the weighted centroid algorithm between the closest fixed node and mobile node, and the blue line shows the distance between mobile and fixed node with vanilla triangulation method.
The error was reduced to less than 50 centimeter as number of nodes were increased. However, reducing the fixed nodes greatly increases the distance estimation error, below 3 beacon nodes the algorithm cannot detect the unknown node. Reducing the distance estimation error is of grave importance as shown by the red and blue lines because this error is carried forward to the

position estimation. The bigger the difference in actual and estimated distance the bigger will be the difference between actual and estimated position of the mobile node.

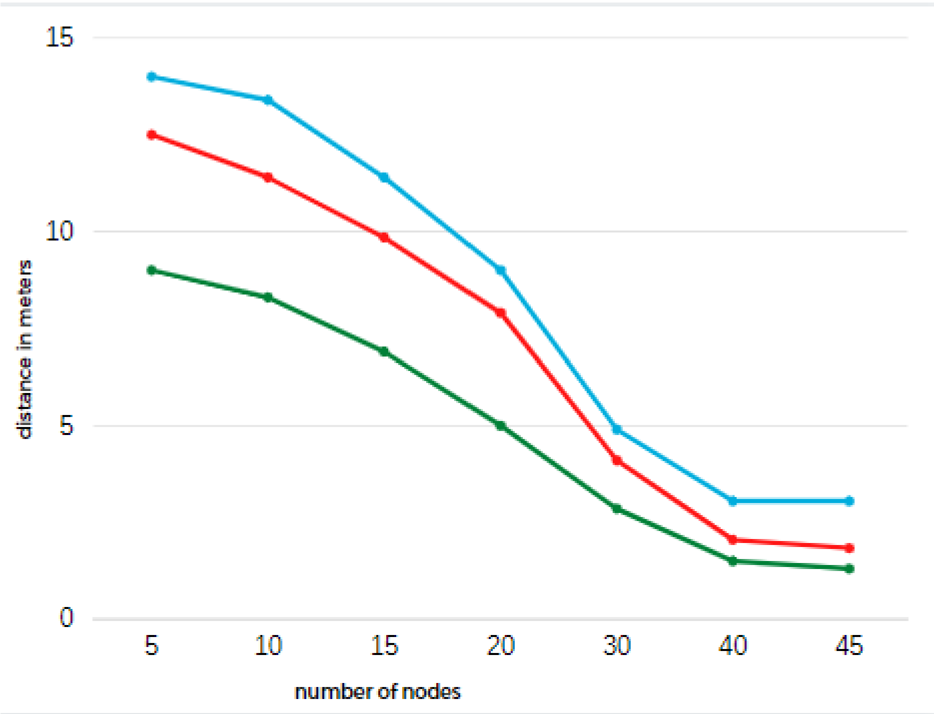

Figure 5.2: Comparison of number of fixed nodes with distance error

Figure 5.3 shows the comparison of traditional weighted centroid algorithm and the proposed algorithm with the position calculation for unknown node in a closed area with all the conditions such as time, speed, and pathloss the same for both algorithms.It can be seen that proposed solution gives better results and since the is ignored interference/noise in indoor environment, the localisation error is very small. However as the mobile node moves the error increases and when it is stationary the algorithm can localise with error less than 50 cm.

Fig 5.4 shows the comparison of traditional weighted centroid algorithm and the proposed algorithm with the position calculation for unknown node in a closed area with all the conditions such as time, speed, and pathloss same for both algorithms. It can be seen that initially the traiditional centeroid algorithm works better but as time passes by the proposed algorithm adjusts the distance to nearest known nodes and performs better. The error is reduced as the mobile node spends more and more time in the network.

The results from this work have confirmed that estimation of path loss and

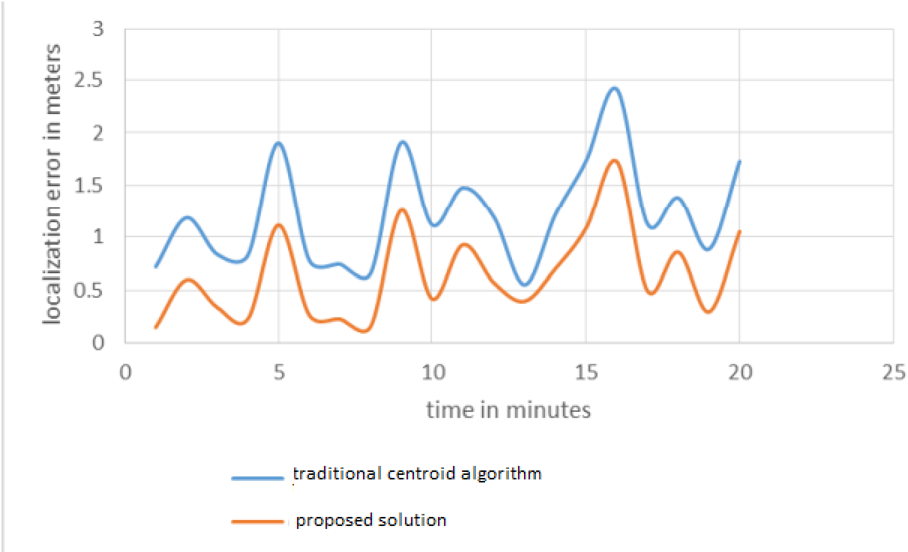

Figure 5.3: Comparing proposed solution with traditional centroid solution in indoor area

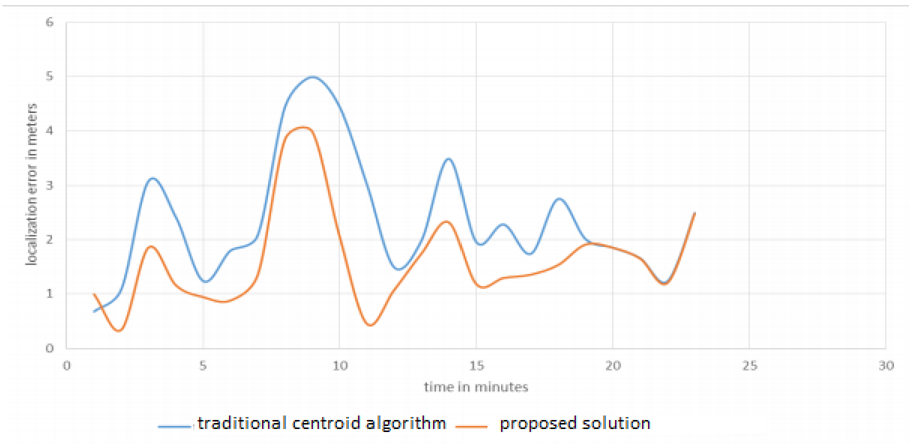

Figure 5.4: Comparing proposed solution with traditional centroid solution in open area

use of centroid weighted algorithm for localization is a viable technique as previously investigated in [30].

Chapter 6

Conclusion & Future Work

Lastly, this chapter concludes the presented research work. In which, section 6.1 presents the conclusion of this research work and 6.2 covers the future directions and some other research challenges that need to be addressed and section .

6.1 Conclusion

In conclusion,we were aiming to reduce localization error so that the unknown node/target node can be localized as accurately as possible by first reducing the distance estimation error and then position estimation error. The distance estimation error is carried forward to the position estimation, so to reduce the position estimation error the distance estimation error was reduced to 55cm.

Distance estimation and position estimation are recalculated as soon as the location of the mobile node changes, so its location is known at all times. Once the location of mobile node is estimated, it is forced to include its X,Y coordinates in future messages to other nodes. The results show that the localization algorithm works better in higher node density as more nodes are in range of the unknown mobile node and it can localize itself with less error in position estimation.

6.2 Future Work

There is still a room for improvement for reducing the localization error, as conditions like humidity, temperature and other physical obstructions were not factored. As these conditions can increase the pathloss factor and thus decreasing the RSSI value and often nondelivery of information to nearby nodes. Factoring in these conditions will provide more real-environment RSSI values and further reduce the localization error. All the simulations were run in a 2d plane so position estimation can be further reduced if instead of 2d plane, 3d plane is considered. 3D plane will provide the Z coordinates which will make it possible to localise in real world.

References

[1] Ion G. Tudorache, Andrew Kemp "Indoor RSSI-Based Ranging Consistency and Error Factors in Wireless Sensor Networks." 20th Telecommunications forum TELFOR 2012.
[2] Ahmad Ali, Yu Ming, "A Comprehensive Survey on Real-Time Applications of WSN", Future Internet 9(4):77 · November 2017
[3] Mirko Ivanić, Ivan Mezei, "Distance Estimation Based on RSSI Improvements of Orientation Aware Nodes" 2018 Zooming Innovation in Consumer Technologies Conference (ZINC).
[4] Indu, Sunita Dixit, "Wireless Sensor Networks: Issues Challenges", IJCSMC, Vol. 3, Issue. 6, June 2014.
[5] http://blog.industrialnetworking.com
[6] Bhisham Sharma ; Deepika Koundal, "Cattle health monitoring system using wireless sensor network: a survey from innovation perspective." IET Wireless Sensor Systems (Volume: 8 , Issue: 4 , 8 2018)
[7] Jungang Zheng , Yue Liu, Xufeng Fan and Feng Li, "The Study of RSSI in Wireless Sensor Networks" 2nd International Conference on Artificial Intelligence and Industrial Engineering, 2016.
[8] R. Dalce, T. Val and A. Bossche, "Comparison of Indoor Localization Systems Based on Wireless Communications." Wireless Engineering and Technology, Vol. 2 No. 4, 2011, pp. 240-256. doi: 10.4236/wet.2011.24033.
[9] Xu, Huping, Feng Li, and Yancheng Ma. "A ZigBee-based miner localization system." Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on. IEEE, 2012.
[10] Yang, Dongxuan, Yan Chen, and Kedong Wang. "A Coal Mine Environmental Monitor System with Localization Function Based on ZigBee-Compliant Platform." Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on IEEE, 2011.
[11] Jian, Li, and Liu He-ping. "A new weighted centroid localization algorithm in coal Mine wireless sensor networks." Computer Research and Development (ICCRD), 2011 3rd International Conference on. Vol. 3. IEEE, 2011.

[12] Wang, Yang, Liusheng Huang, and Wei Yang. "A novel real-time coal miner localization and tracking system based on self-organized sensor networks." EURASIP Journal on Wireless Communications and Networking 2010.1 (2010): 1-14.
[13] Shaoguo Xie, Yanjun Hu and Yi Wang, "Weighted centroid localization for wireless sensor networks," Consumer Electronics - China, 2014 IEEE International Conference on, Shenzhen, 2014, pp. 1-4.
[14] Fan, Hongbo, et al. "Weighted centroid localization algorithm based on improved RSSI ranging." Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on. IEEE, 2013
[15] Xu, Dengyuan, Benniu Zhang, and Zhongwei Hou. "Novel PIT localization algorithm based on coverage of anchors in WSN." Journal of Networks 7.9 (2012): 1349-1354.
[16] Lau, Erin-Ee-Lin, et al. "Enhanced RSSI-based high accuracy real-time user location tracking system for indoor and outdoor environments." International Journal on Smart Sensing and Intelligent systems 1.2 (2008): 534-548
[17] Liu, Junjie. "Survey of Wireless Based Indoor Localization Technologies." Dept. of Science Engineering, Washington University (2014).
[18] Zhonghua Zhang , Di Chen. "An improved RSSI-based centroid localization algorithm in wireless sensor networks." 2011 International Conference on Computer Science and Service System (CSSS)
[19] Chong Zhao , Bo Wang. "A MLE-PSO indoor localization algorithm based on RSSI" 2017 36th Chinese Control Conference (CCC)
[20] Jianqiao Xiong , Qin Qin , Kemin Zeng. "A Distance Measurement Wireless Localization Correction Algorithm Based on RSSI." 2014 Seventh International Symposium on Computational Intelligence and Design
[21] Akhand Pratap Singh , et al. "New proposed RSSI method for the distance measurement in WSNs." 2015 1st International Conference on Next Generation Computing Technologies (NGCT)
[22] Ann Whitney , et al. "RSSI informed phase method for distance calculations." 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)
[23] Xiaoyong Yan , et al. "A high accuracy localization based on RSSI measurements." 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)
[24] Joanna Janicka ; Jacek Rapinski. "Application of RSSI Based Navigation in Indoor Positioning." 2016 Baltic Geodetic Congress (BGC Geomatics)
[25] Bo Guan , Xin Li. "Research on Wireless Sensor Networks Localization Algorithm." 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)

[26] Wangsheng Fang , Guangyu Yang. “Improvement based on DV-Hop localization algorithm of wireless sensor network.” 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)
[27] Zhang Dengyi, et al. “Improvement of DV-Hop Localization Algorithms in Wireless Sensor Networks.” 2012 International Symposium on Instrumentation Measurement, Sensor Network and Automation (IMSNA)
[28] Xu Yingxi , Gao Xiang ,Sun Zeyu , Li Chuanfeng. “WSN Node Localization Algorithm Design Based on RSSI Technology.” 2012 Fifth International Conference on Intelligent Computation Technology and Automation
[29] Wang Suzhe , Li Yong. “Node localization algorithm based on RSSI in wireless sensor network.” 2012 6th International Conference on Signal Processing and Communication Systems
[30] Nazish Tahir, ”Localization issues in wireless sensor networks in underground mines”, MS Thesis SEECS-NUST May 2016.

Annex

Installation of NS2 in ubuntu

This section provides step by step installation process of NS2 to run a basic simulation in Ubuntu 16.04.

Download and Extract ns2

Download NS2 package from
“http://sourceforge.net/projects/nsnam/files/latest/download ”
The package downloaded will be named ”ns-allinone-2.35.tar.gz”. Copy it to the home folder. Then in a terminal use the following two commands to extract the contents of the package
cd / tar -xvzf ns-allinone-2.35.tar.
All the files will be extracted into a folder called ”ns-allinone-2.35”

Building the dependencies

Ns2 requires a few packages to be pre-installed. It also requires the GCC-version 4.4 to work correctly. So install all of them by using the following command: **sudo apt-get install build-essential autoconf automake libxmu-dev**
sudo apt-get install gcc-4.4
Navigate to the folder ”linkstate”, use the following command. Here it is assumed that the ns folder extracted is in the home folder of your system.
/ns-allinone-2.35/ns-2.35/linkstate
Now open the file named ”ls.h” and scroll to the 137th line. In that change the word ”error” to ”this-¿error”.

Installation

Running the command **sudo su cd /ns-allinone-2.35/./install** will start the installation process of ns2.
Once the installation is completed (it can take from 10 to 30 minutes), we need to set up the environment path

Setting the Environment Path

To set the environment path using the ".bashrc" file. In that file, we need to add a few lines at the bottom. The things to be added are given below.
XGRAPH=/home/user/ns-allinone-2.35/bin:/home/user/ns-allinone-2.35/tcl8.5.10/unix:/home/user/ns-allinone-2.35/tk8.5.10/unix NS=/home/user/ns-allinone-2.35/ns-2.35/ NAM=/home/user/ns-allinone-2.35/nam-1.15/ PATH=$PATH$:**XGRAPH:**NS :**NAM**
Save the file and restart the system.

Running NS2

Open the terminal and run the command **ns**

Code Snippets

Snippet 6.1 shows the edition in NS2 **mac-802_11.h** file, where extra parameters are introduced for X, Y coordinate and the pathloss factor. All the beacon frames will include tese information in their headers so that receiving devices can extract this information for distance estimation and localization. Snippet 6.2 shows all the information receiving node will store upon getting a beacon frame, regardless of its nature, whether fixed or mobile. These changes are also in the NS2 mac-802-11.h file. Snippet 6.3 shows the intilaiz ation of nodeDB array, each node is given an ID. The pathloss and distance are set to 0 and -1 respectively as they will be calculated later. These changes are in the NS2 **mac-802_11.cc** file. Snippet 6.4 shows the RSSI calculation for fixed and mobile node. The fixed node further calculates the distance between itself and sender and the calculates the pathloss factor.However, the mobile node does not need to calculate pathloss factor as it is already advertised by the becaon node. These changes are in the NS2 **mac-802_11.cc** file. Snippet 6.5 shows the position estimation for the mobile node.These changes are also in the NS2 **mac-802_11.cc** file. Snippet 6.6 shows the code for our network model, it sets the channel type, number of nodes routing protocols,

```
struct beacon_frame {
    struct frame_control    bf_fc;
    u_int16_t        bf_duration;
    u_char           bf_ra[ETHER_ADDR_LEN];
    u_char           bf_ta[ETHER_ADDR_LEN];
    u_char           bf_3a[ETHER_ADDR_LEN];
    u_int16_t        bf_scontrol;
    double           bf_timestamp;
    double           bf_bcninterval;
    u_int8_t         bf_datarates[1];
    u_char           bf_fcs[ETHER_FCS_LEN];
    // Adding XY and pathloss info to Beacon frame
    u_int16_t bf_X;
    u_int16_t bf_Y;
    float bf_pathloss;
};
```

Figure 6.1: Changes in NS2 mac-802_11.h file

```
//Soring beacon info base on eash node
struct nodeInfo {
    u_int32_t nodeID;   //index of array and this value should remain same
    float X;     //ERROR: Beacons transmit them in u_int16_t
    float Y;
    float powerRecved;  // for RSSI value calculation in dbm
    float distance; //distance estimation
    float distanceTrad; //Distance computed from traditional algo
    float beta; //pathloss of the soruce node
};
```

Figure 6.2: Information stored on each node

data rate etc. This is our custom topoly file in tcl format that runs on the NS2. Snippet 6.7 shows the tcl script code for arranging some of the fixed nodes horizontally in the coverage area.

```
//Initializing the whole nodeDB to null values:
for (int i = 0; i < 100; i++) {
    nodeDB[i].nodeID = 999;
    nodeDB[i].pathloss = 0;
    nodeDB[i].distance = -1;
    nodeDB[i].powerRecved = -1;
}
//except the last one for reference purpose (farthest node);
nodeDB[99].beta = 999;
nodeDB[99].distance = 999;
nodeDB[99].powerRecved = -100;
lS = 0;

// chk if basic/data rates are set
// otherwise use bandwidth_ as default;

Tcl& tcl = Tcl::instance();
tcl.evalf("Mac/802_11 set basicRate_");
if (strcmp(tcl.result(), "0") != 0)
bind_bw("basicRate_", &basicRate_);
else
```

Figure 6.3: Intilazing nodes array in mac-802_11.cc file

```
//Calculating RSSI
MobileNode *thisnode = (MobileNode *)(Node::get_node_by_address(index_));
double power = p->txinfo_.RxPr;
double RSSI = 10*log10(power*1000);
double A= 22.628;
// for fixed node
if (nodeDB[src].nodeID == 999) {
    nodeDB[src].nodeID = src;
    nodeDB[src].X = bf->bf_X;
    nodeDB[src].Y = bf->bf_Y;
    nodeDB[src].dist = sqrt(pow(nodeDB[src].X-nodeDB[dist].X) +
        pow(nodeDB[src].Y-nodeDB[dist].Y));
    nodeDB[src].pathloss = -(RSSI-A)/10*log10(nodeDB[src].dist);

} // for MobileNode
else {
    nodeDB[src].X = bf->bf_X;
    nodeDB[src].Y = bf->bf_Y;
    nodeDB[src].RSSI = RSSI
    nodeDB[src].pathloss = bf->bf_pathloss;
}
```

Figure 6.4: Changes in mac-802_11.cc file for RSSI and pathloss calculation

```
//position estimation of unknown node

struct LocationCoords{
                float value1;
                float value2;};
public LocationCoords positionEstimation(nodeDB){
    if (nodeDB[a].distance > 0 && nodeDB[b].distance > 0 && nodeDB[c].distance > 0 ) {

    float eq1 = (nodeDB[a].X/nodeDB[a].distance) ;
    float eq1 += (nodeDB[b].X/nodeDB[b].distance) + (nodeDB[c].X/nodeDB[c].distance);
    float eq2 = (nodeDB[a].Y/nodeDB[a].distance) + (nodeDB[b].Y/nodeDB[b].distance) ;
    float eq2 += (nodeDB[c].Y/nodeDB[c].distance);
    float eq3 = (1/nodeDB[a].distance) + (1/nodeDB[b].distance) + (1/nodeDB[c].distance);
    localizedX = eq1/eq3;
    localizedY = eq2/eq3;

    printf("%d %d \t", localizedX, localizedY);
    LocationCoords location = {localizedX, localizedY}
    return location;
    }
    else retun null;
}
```

Figure 6.5: Position Estimation for mobile node

```
set val(chan)           Channel/WirelessChannel     ;#Channel Type
set val(prop)           Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)          Phy/WirelessPhy             ;# network interface type
set val(mac)            Mac/802_11                  ;# MAC type
set val(ifq)            Queue/DropTail              ;# interface queue type
set val(ll)             LL                          ;# link layer type
set val(ant)            Antenna/OmniAntenna         ;# antenna model
set val(ifqlen)         50                          ;# max packet in ifq
set val(nn)             40              ;# number of mobilenodes
set val(rp)             DumbAgent                   ;# routing protocol
set val(x)      5
set val(y)      100
set val(nam)        beacon.nam;
Mac/802_11 set dataRate_ 11Mb

Phy/WirelessPhy set CSThresh_ 10.00e-12
Phy/WirelessPhy set RXThresh_ 10.00e-12

Phy/WirelessPhy set Pt_ 0.008
```

Figure 6.6: Declarations in TCL file for network model and topology

```
# Create channel
set chan_1_ [new $val(chan)]


$ns_ node-config -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -topoInstance $topo \
        -agentTrace OFF \
        -routerTrace OFF \
        -macTrace ON \
        -movementTrace ON \
        -channel $chan_1_

# Horizontal arrangement of nodes
        $node_(1) set X_ 2
        $node_(1) set Y_ 1


        $node_(2) set X_ 1
        $node_(2) set Y_ 5
```

Figure 6.7: Arranging the fixed nodes in the 2D plane area.