JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Improvement on Approximate Point In Triangle (APIT) Localization Algorithm in Large-scale Wireless Sensor Networks

By

Kelil Mohammed

April, 2019

Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Improvement on Approximate Point In Triangle (APIT) Localization Algorithm in Large-scale Wireless Sensor Networks

By

Kelil Mohammed

A thesis submitted to the School of graduate studies at Jimma University, Jimma Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Networking

Principal Advisor: Mr. Fisseha Bayu (Ph.D. Candidate)

Co-Advisor: Mr. Tesfu Mekonen (M.Sc.)

April, 2019

Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Improvement on Approximate Point In Triangle (APIT) Localization Algorithm in Large-scale Wireless Sensor Networks

By

Kelil Mohammed

Approved by Board of Examiners:

Mr. Fisseha Bayu (Ph.D. Candidate)	_____________	____________
Principal Advisor	Signature	Date
Mr. Tesfu Mekonen (M.Sc.)	______________	_____________
Co-Advisor	Signature	Date
Dr.-Ing Towfik Jemal	______________	_____________
External Examiner	Signature	Date
Dr. Girum Ketema	____________	_____________
Internal Examiner	Signature	Date
Mr. Worku Birhane (M.Sc.)	_____________	_____________
Chair Man	Signature	Date

April, 2019

Jimma, Ethiopia

DECLARATION

I, the undersigned, declare that this thesis entitled "Improvement on Approximate Point In Triangle (APIT) Localization Algorithm in Large-scale Wireless Sensor Networks" is my original work and has not been presented for a degree in this or any other universities, and all sources of references used for the thesis work have been appropriately acknowledged.

Name: Kelil Mohammed

Signature: ______________ Date: _________________

This thesis has been submitted for examination with my approval as a Principal advisor.

Advisor Name: Mr. Fisseha Bayu (Ph.D. Candidate)

Signature: _______________ Date: ___________________

This thesis has been submitted for examination with my approval as a Co-advisor.

Co-Advisor: Mr. Tesfu Mekonen (M.Sc.)

Signature: ______________ Date: _________________

Acknowledgement

First of all, I would like to say, ***"Alhamdulilahi Rabilalemin***". Next I would like to express my sincere gratitude to my principal advisor Mr. Fisseha Bayu (Ph.D. Candidate) and to my co-advisor Mr.Tesfu Mekonen for their valuable comments, guidance and encouragement while I was working on my thesis research. Unless and otherwise for their unreserved guidance and support, this work would not come to the reality.

Next I would like to express my heartfelt thanks to my wife Tigist Kebede, for her moral support , encouragement and taking more responsibility of up and downs of family life. Love of Abdi and Rahma is my especial energy to see the bright future.

I would like to take this opportunity to thank my friends, Gezahagn Gutema, Kedir wata and Jemal Abdulkadir for their moral support and advice in difficult situation of my life during this thesis work.

Last but not least, I would like to thank all teaching staff in school of computing and my especial thanks goes to Mr. Kebebew Ababu, chairperson of Networking and Operating System for his unreserved cooperation.

Dedication

To: my children (Abdi Kelil Mohammed & Rahmet Kelil Mohammed)

Table of Content

List of Tables

List of Figures

List of Acronyms

ADC	Analog-To-Digital Converter
ADH	Average Distance of Hop
ALS	Area Localization Scheme
AOA	Angle Of Arrival
AP	Anchor Percentage
APIT	Approximate Point In Triangle
CLA	Centroid Localization Algorithm
COG	Center Of Gravity
DOI	Degree Of Irregularity
DV_hop	Distance Vector hop
GPS	Global Positioning System
HTCRL	Homothetic Triangle Cyclic Refinement Location
IM_APIT	Improved Approximate Point In triangle
MAC	Medium Access Control
MATLAB	Matrix Laboratory
MEMS	Micro Electro-Mechanical systems
PIT	Point In Triangle
RF	Radio-Frequencies
RTD-APIT	Received Signal Strength Indicator and Triangle Deformation based APIT
TDOA	Time Difference Of Arrival
RSSI	Received Signal Strength Indicator
TOA	Time Of Arrival
WSNs	Wireless Sensor Networks

Abstract

In wireless sensor networks, the location of sensor nodes that generate information in application such as target tracking, geographic routing , environmental monitoring and forest fire monitoring are very essential.

Localization techniques are used to estimate the locations of the unknown nodes in a network using the available few anchors in network and inter-sensor measurements such as time different of arrival, angle of arrival and connectivity, and of which the three are categorized under range based localization algorithm and the last one is under range free localization algorithm.

In localization of unknown sensor nodes in wireless sensor networks, the straight forward solution is installing the GPS in every sensor nodes. But it is infeasible due to lack of line-of-sight in indoor applications, high network cost because of energy consumption and hardware cost.

In the localization algorithms, the range based algorithms are better in location accuracy but they need additional hardware requirement and energy which is the tradeoffs. In the case of range free localization algorithms, they are cost and energy efficient but with less localization accuracy. Since most of the wireless sensor network applications need coarse localization accuracy, range free localization algorithms are more effective than the range based ones, taking into consideration the tradeoffs. Among range free localization algorithms APIT, DV-hop and Centroid localization algorithms are the popular ones.

In this thesis, we proposed improved APIT localization algorithm by developing novel and simple Point in triangle test to minimize the probability of wrong judgement of whether the unknown sensor node is inside triangle formed by the three anchor nodes or outside the triangle.

We have evaluated the performance of both the improved APIT and the original APIT based on the localization error, and the simulation experiment result show that improved APIT localization algorithm has better performance than the original APIT. Accordingly localization error rate of improved APIT localization algorithm is 26%, while localization error rate of APIT localization algorithm is 31% when anchor node percentage is 25% of the total sensor nodes.

Chapter One: Introduction

1.1 Background

Wireless sensor network (WSN) is a new class of wireless network that is becoming very popular with a huge number of civilian and military applications. WSNs have gained worldwide research and industrial interest particularly with the rapid increase in wireless communication technologies and micro-electromechanical systems (MEMS) technology which has facilitated the development of smart sensors [1]. A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing [2]

A Wireless Sensor Network (WSN) is a network of independent, battery powered, low-energy, limited processing capability, low-cost, low storage, and low communication capacity tiny sensing stations known as sensor nodes [3]. Each sensor node consists of sensing unit, micro-controller unit or processing unit, transceiver or communication unit and power unit [4]. Upon deployment these sensor nodes organize themselves and form a network, which is typically consists of several to thousands of such sensor nodes. These sensor nodes are used to monitor the surrounding environment for some activity and send the sensor data to base station known as sink. The deployment of the sensor nodes can be on large scale or small scale depending upon the application to detect and collect the information from physical environment [3], [5]

WSNs have significant characteristics [6] such as low-cost, energy efficient, low computational power, low communication capabilities, low security and privacy, distributed sensing and processing, dynamic network topology, self-organizing, multi-hop communication, application oriented and robust operation.

The earlier applications of WSN were developed for military usage, from surveillance systems in the oceans for the detection of objects in different environments. However, the possibility of developing low cost sensors and with low consumption of energy, with wireless communication, has already allowed their usage in many civil and military applications. Since most applications depend on a successful localization of the nodes, it is necessary to compute their position in a network, for designing efficient localization algorithms. An efficient localization improves saving energy and reduces the communication cost among sensor nodes. Because of these advantages the precise knowledge of node localization in WSN is an active field of research [7]

Localization in wireless sensor networks (WSNs) is one of the central components of a variety of emerging applications including military, healthcare, environment monitoring, home and office automation, weather forecasting and so on. Many of these applications need location based services [8]. Location based services provide relevant (spatial) information to the end users through wireless networks and/or the internet. Applications that provide location based services can offer the context and the connectivity needed to dynamically associate the position of a user to context sensitive information about current environments and it send data by knowing the geographical location accessed by a mobile user [5]

Localization or positioning is the task of determining physical coordinates of sensor nodes in WSNs and is a key factor in today's communication systems to estimate the place of origin of events [5].

Localization techniques are used to estimate the locations of the unknown nodes in a network using the available few anchors in the network and inter-sensor measurements such as distance, time difference of arrival, angle of arrival and connectivity. Anchors locations can be obtained by using a global positioning system (GPS) or by installing anchors at points with known coordinates by some mechanism. Because of constraints on the cost and size of sensors, energy consumption, implementation environment, it is not accessible to embed all tiny sensor nodes with GPS in large-scale resource constrained wireless sensor networks [9], [10]. Therefore location of sensor node need to be estimated using localization algorithm.

Based on the positioning mechanism [10], [11], [12], the localization algorithms can be divided into two broad categories: (i) range-based localization techniques and (ii) range-free localization techniques. Range-based schemes have higher location accuracy than the range-free localization schemes, but are hardware intensive and high cost of sensor nodes. In range-free schemes, special hardware for distance estimation is not used and they are taken as a cost effective alternative to more expensive range-based approaches[9],[13] Approximate-Point-In-Triangle (APIT), Distance-vector-Hop (DV-Hop), and Centroid localization algorithms are the popular range-free localization algorithms[12]. Range-free localization algorithms even if they are cost effective, they have their own drawbacks like less accurate to localize sensor nodes.

Localization in non-line-of-sight like for range based localization scheme, node selection criteria for localization in energy-constrained network, scheduling the sensor node to optimize the tradeoff between localization performance and energy consumption, cooperative node localization, and localization algorithm in heterogeneous network are some of the challenges of localization in WSNs [2].

1.2 Statement of the Problem

Range-based localization algorithms are accurate relative to range-free algorithms but their implementation need sophisticated hardware that GPS embedded to and as the result of this it is infeasible in large-scale resource constrained WSNs [9],[13] . Therefore the alternative solution is the range-free (connectivity-based) localization algorithms and they do not need additional hardware, are cost effective but they are less accurate to estimate location of sensor nodes. Since most of the WSN applications need coarse accuracy localization, implementation of range-free algorithms is feasible. According to [14], APIT is a range–free localization algorithm which is simple and easy to implement. It is low in communication cost and energy consumption. It also has a low location error rate. However, the performance of the APIT algorithm is largely affected by the density of anchor nodes and is restricted for the In-to-Out Error and Out-to-In Error. Therefore there exists gap to improve this localization algorithm.

A key procedure in APIT localization algorithm is PIT test and the main shortcoming of this algorithm is also based on PIT test in which it is inevitable to make wrong judgments [14],[15],[16],[17],[18] that an internal node can be seen as outside of the triangle (In-To-out error) or judge an outside node as it is inside the triangle (Out-To-In error).

To illustrate these two errors, let A, B and C are three anchor nodes (or vertices of triangle ABC) that form triangle ABC and let M be the target node. As in [19] comparing received signal strength, in figure 1.1 (i) below as we can see M is inside ΔABC, the received signal strength of node 2 receive from anchors A, B and C is less than the received signal strength M receive from A, B and

C, and due to this neighbor information, node M misjudged to be outside of triangle ABC (In-To-out error occurred). Similarly in figure 1.1 (ii) node M is actually outside of ΔABC, the neighbor node 3 of M is nearer to anchor node A (vertex A of ΔABC) than anchor nodes B and C (vertices B and C of ΔABC) or node 4 is nearer to vertex B and further from vertex A and C, and this leads to wrong judgment that node M is inside triangle ΔABC (Out-To-In error).

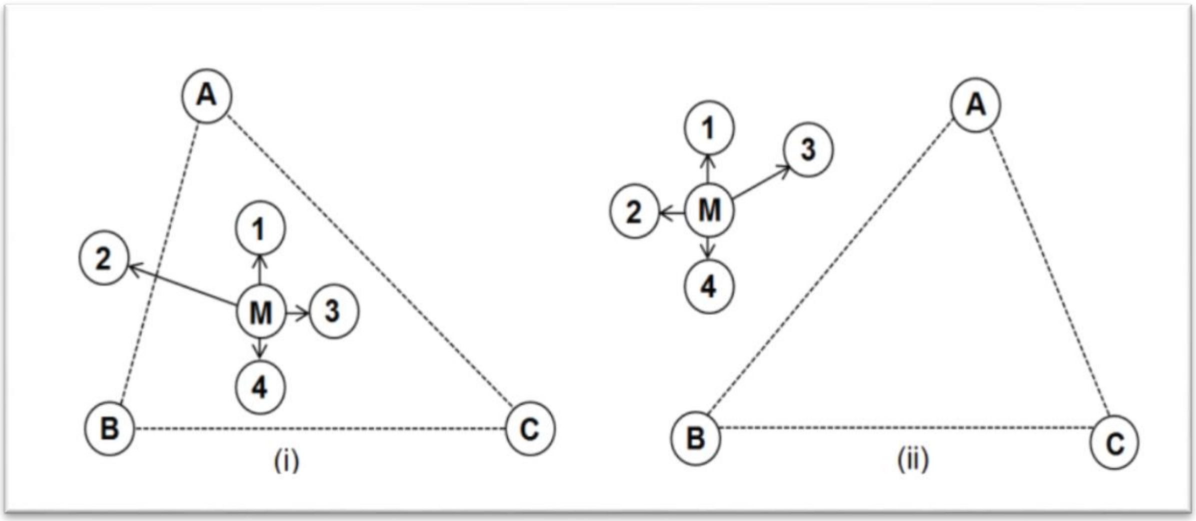


Figure 1.1: In-To-Out and Out-To-In Error situations [19]

To minimize these errors, this thesis is aimed to develop new side based PIT test method. The new side based PIT test method, compare the sum of distances from a target node to vertices of a triangle with perimeter of a triangle to take decision as a PIT test.

Finally the thesis will attempt to answer the following research question:

- To what extent localization error is minimized in comparison to original APIT localization algorithm as a result of new PIT test?

1.3 Objectives

General objective

The general objective of this thesis is to improve judgment accuracy in PIT test phase of APIT localization algorithm of sensor nodes in large-scale wireless sensor networks, so that to improve localization accuracy of the algorithm.

Specific objectives

The specific objectives of this thesis include:

- To Design new side based PIT test method
- To set mathematical model that used as new PIT test method
- To compute perimeter of virtual triangle formed by three anchor nodes
- To calculate the sum of distances from a target node to the vertices of the triangle (three anchors).
- To compare perimeter of the virtual triangle formed by the three anchor nodes with the sum of distance from the target sensor node to the vertices of the triangle, to use as a PIT test.
- To implement the proposed improved APIT localization algorithm.
- To evaluate the performance of improved APIT localization algorithm.

1.4 Scope of the Study

This thesis work is focused on two dimensional network and sensor nodes can in general be moving or stationary. However this thesis is delimited to study localization problem of fixed sensor nodes in wireless sensor networks.

1.5 Methodology and Tools

To achieve the objective of the thesis the following steps will be used:

Literature Review: We will review literatures, published related works, books, journals and internet to gather information about study area relevant to the thesis.

Experimentation: experimentation on proposed APIT localization algorithm will be employed to achieve the objective of the study.

Tools: MATLAB software will be used as software tool of the study to obtain experimental results and to implement, to simulate and to analyze the performance of the proposed algorithm. MATLAB [41] relative to others simulation environment, its programming syntax is simple and it contain built-in complex mathematical expressions.

1.6 Thesis Organization

This thesis work consists of six chapters. The next chapter review literatures on the architecture of wireless sensor networks, components of a sensor node, characteristics of wireless sensor networks , protocol stack in wireless sensor networks, advantages and limitations of wireless sensor networks, application of wireless sensor networks and localization algorithms in wireless sensor networks. Chapter three reviews related works. Chapter four deals with design and proposed solution and chapter five discusses the simulation tools used for this work, experimentation and evaluation of proposed solution. Finally, chapter six summarizes the thesis by conclusion and future work.

Chapter Two: Literature Review

2.1 Overview of Wireless Sensor Networks

WSN is combination of sensing, computation, and communication into a single tiny device known as sensor node. It consists of hundreds and thousands of tiny battery powered, randomly deployed sensor nodes with limited processing, storage, and communication capacity. According to [15] the typical WSN consists of a number of tiny devices with microcontroller, a low-powered radio, and a number of sensors to perceive their surrounding environment. These devices are networked in a multi-hop fashion to enable cooperation among nodes and real-time delivery of sensed data to the user.

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, moisture, motion, pressure, heat or any interested environmental phenomena. The output is generally a signal that is converted to human readable display at the sensor location or transmitted electrically over a network for reading or further processing [2]. Each sensor node is able to sense the environment, perform simple computations and communicate with its neighbor other sensors or with the central unit. In wireless sensor networks sensor nodes are of two types, anchors and non-anchors (or simply sensor nodes). Anchors are those who are aware of their locations and non-anchors (or simply sensor node) are those who do not know their locations.

One way of deploying the sensor networks is to scatter the nodes throughout some region of interest. Once the sensor nodes are deployed into the target area, they collect data from the environment automatically and establish an ad hoc network to transfer their data to the base station (sink) [11]. A sink can store received packets and forwarded them to external network using reliable and most possibly wired communication link.

2.2 Architecture of Wireless Sensor Networks

The term architecture has been adopted to describe the activity of designing any kind of system, it is the complex or carefully designed structure of something; one of its common uses is in describing information technology, such as computer architecture and network architecture [20]. The architecture of wireless sensor network consists of sensor nodes, sink (base station) and user.

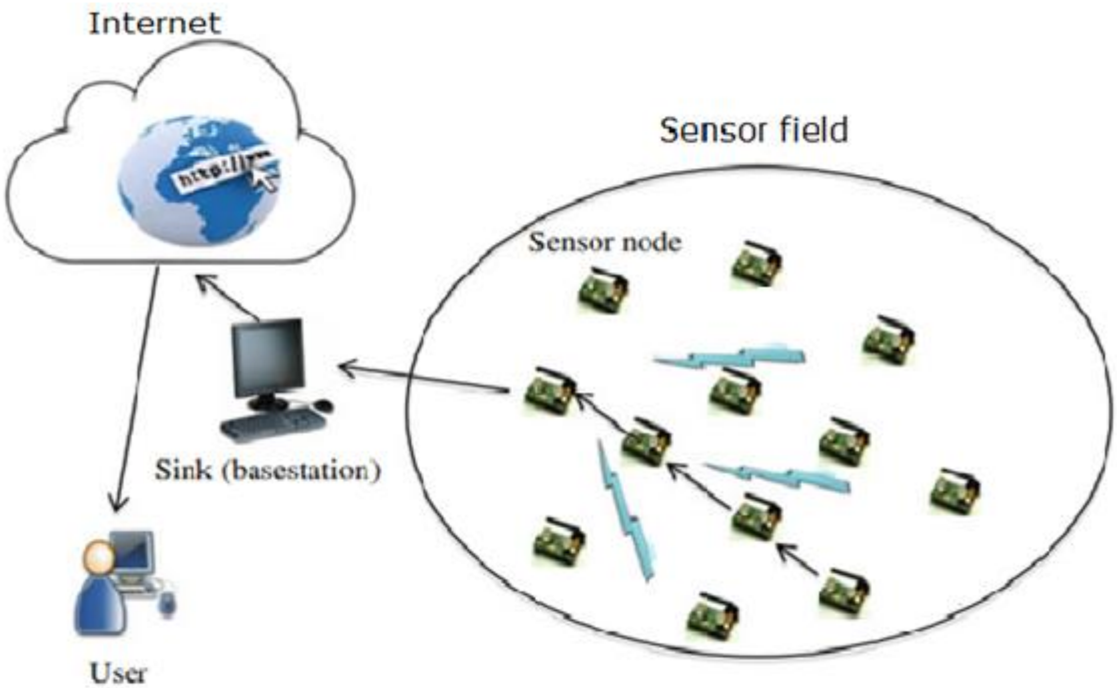


Figure 2.1: Architectures of WSNs

- ✓ The sensor nodes form the sensor network. Their main objectives are making discrete, local measurement about phenomenon surrounding these sensors, forming a wireless network by communicating over a wireless medium, and collecting data and routing data back to the user via a sink (base station).
- ✓ The sink (base station) communicates with the user via Internet or satellite communication. It is located near the sensor field. The collected data from the sensor field is routed back to the user by a multi-hop infrastructure less architecture through the sink.
- ✓ The user is one who is interested in obtaining information about a specific phenomenon to measure or monitor its behavior.

2.3 Components of a sensor node

Sensor node is one of the main components of any WSN. Sensor nodes have the following common characteristics: small physical size, low power consumption, limited processing power, short range communications, low cost and a small amount of memory storage [21]. A sensor node is made up of four basic components as shown in figure 2.2; sensing unit, processing unit, transceiver unit (communication unit), and power supply unit.

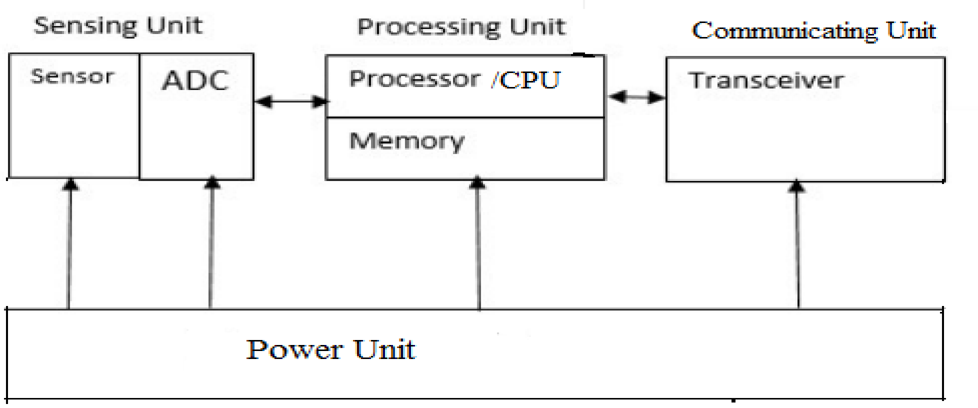


Figure 2.2: Basic Components of a sensor node

These basic components of a sensor node described as follows:

Sensing Unit: - It senses the environment through transceiver. Sensing unit [4] has two subunits, sensors and ADC. Sensors are considered as a vital part of any WSN. They are basically hardware devices that generate a signal proportional to the event or condition being monitored or measured. The sensed signal is converted to digital form using analog-to-digital converters (ADC) as the microcontrollers can only process digital data. Desired characteristics of a sensor node include small size, low power consumption, being adaptive to the environment, and being independent and able to work unattended.

Based on power or energy supply requirement, sensors [4] are categorized as; active sensors, passive and omnidirectional sensors, and passive and narrow-beam sensors, and are described as follows:

- ✓ **Active sensors**: - This category of sensors actively surveys the surroundings, for instance, a radar sensor or a seismic sensor that produces shock waves.
- ✓ **Passive and omnidirectional sensors**:-. These sensors monitor the data without really affecting the environment by active investigation. Normally, they are self-powered. Power is required only to magnify their collected analog data. Further, these sensors pick up the sensed data from all directions; they are not directional.

- ✓ **Passive and narrow-beam sensors: -** Sensors of this category are passive; however, they have distinct view of direction of measurement. Usually devices such as cameras are used. In addition to these units, a wireless sensor node may be equipped with extra units such as a global positioning system (GPS) device, and a motor to move the sensor node in specific directions, among others. Of course, such components must be built into a small module with low power expenditure and low fabrication cost.

According to [2], based on their applications, sensors are classified into the following groups and described as follows:

- ✓ **Accelerometers:** - These group of sensors are based on the Micro Electro Mechanical sensor technology. They are used for patient monitoring which includes pacemakers and vehicle dynamic systems.
- ✓ **Biosensors:** - sensors of this group are based on the electrochemical technology. They are used for food testing, medical care device, water testing, and biological warfare agent detection.
- ✓ **Image Sensors:** - These are based on the CMOS (COMPUT metal oxide semi-conductor) technology. They are used in consumer electronics, biometrics, traffic and security surveillance and PC imaging.
- ✓ **Motion Detectors** - These are based on the Infra-Red, Ultrasonic, and Microwave or radar technology. They are used in videogames and simulations, light activation and security detection

Microcontroller unit/ Processing unit: - The CPU (also called the electronic brain) of a sensor node [23] is composed of a microprocessor and a flash memory. In most of sensor nodes, it includes connectors to add external processing units and sensors to the main unit easily. Making decision and dealing with collected data are the crucial functions of the CPU. The CPU stores data in flash memory until enough data is collected. Once enough data is collected by the system, then microprocessor unit of the CPU puts the data in envelopes, because envelopes provide great efficiency in data transmission. Then, these envelopes are sent to the radio for broadcast. Meanwhile the CPU communicates also with other nodes in the same way it deals with data to maintain the most effective network structure. The CPU is connected to the base and it interacts with the sensors and radio. Flash memories [4] are employed due to their cost-effectiveness characteristics; they offer high capacity at low cost. Memory requirements are application dependent. The required memory type can be divided into two major classes: data memory and program memory.

Transceiver Unit (Communicating Unit):- The term transceiver signifies the combined functions of transmitter and receiver. Transceiver has mainly four operational states which are Receive, Transmit, Idle and Sleep. The Radios operating in the idle state of operation consume power roughly equal to that used in receive state. Hence, it is recommended to turn down the radios instead of running them in the idle state when not sending or receiving data. Moreover, a lot of power is spent switching from the sleep state to the transmit state in order to send a packet. Sensor nodes utilize the industrial, scientific and medical (ISM) frequency band, which is available free of charge and does not require a license. There are different choices of wireless transmission media including: infrared and radio-frequency. Infrared medium is inexpensive; however, it requires line of sight and cannot penetrate opaque objects and walls. Moreover, it is sensitive to atmospheric situations. Infrared does not require antennas, so it has limited broadcasting capacity. Communication based on radio-frequency (RF) is the most-often used means of communication in WSNs. Typically, WSNs use the communication frequencies between 433MHz and 2.4GHz [4] [23].

Power Unit: In WSNs [4] when a sensor node collects data/signal, it consumes power. Moreover, power is consumed even more for communication and data processing operations. Power is typically stored in batteries or capacitors. There are two major types of batteries that are used in sensor nodes: chargeable and non-rechargeable batteries. Moreover, batteries can be classified depending on the electrochemical material used for electrode as nickel–cadmium (NiCd), nickel–zinc (NiZn), nickel metal hydride (Nimh) and lithium-ion. Most WSNs employ power-saving policies in order to extend the life of the batteries in their nodes. Among such schemes that are used are the dynamic power management and dynamic voltage scaling schemes. The first technique shuts down parts of the sensor node that are not presently active. The second scheme changes power levels based on the unpredictable workload. This is performed by changing voltage besides the frequency, which allows achieving decrease in the power consumption. The power unit [26] usually consists of one or more batteries, providing 3V - 4.5V, generally with a capacity ranging between 1700mAh – 2700mAh. The node can be fitted with various sensors for acoustic, photo, temperature, pressure etc. based applications. Each node may also optionally be fitted with an interface for plugging-in an actuator for performing any mechanical actions on an application specific basis.

2.4 Characteristics of Wireless Sensor Networks

Wireless sensor networks have characteristics that are different from traditional wireless networks such as wireless ad hoc networks. The main characteristics of the wireless sensor network are summarized [26] as follows:

Unattended operation: In most cases, once nodes are deployed, wireless sensor networks have no human intervention. Hence the nodes themselves are responsible for reconfiguration in case of any changes.

Dynamic network topology: It is an important aspect of the sensor networks. The lifecycle of a sensor network may be represented in three phases with respect to the topology and its maintenance. During the deployment phase, the nodes are dropped into their positions in an ad hoc manner. The nodes need to self-organize into a communicating network. The Post-deployment phase topology maintenance consists of topology changes induced due to the failure of the nodes, failure of radio links, or arrival of some mobile obstacles. The Re-deployment phase deals with the deployment of nodes to replace failed nodes. In each of the three phases, a sensor network should be capable of seamlessly organizing itself to stream data to the base-station.

Limited power: Sensor Networks are highly sensitive to energy usage. They may, probably, be deployed in hostile environments, where it may not be possible to refresh energy sources. Hence, energy consumption is a major issue, and energy-aware protocols or applications are desirable. Energy consumption is observed at three stages, node communication, sensing and processing. Optimizing the three processes will lead to a reduction in the energy consumed.

Distributed sensing and processing: the large number of sensor node is distributed uniformly or randomly. WSNs each node is capable of collecting, sorting, processing, aggregating and sending the data to the sink. Therefore the distributed sensing provides the robustness of the system.

Node mobility: Mobility of the nodes creates a dynamic network topology. Links will be dynamically formed when two nodes come into the transmission range of each other and are torn down when they move out of range.

Self-organization: the sensor nodes in the network must have the capability of organizing themselves as the sensor nodes are deployed in an unknown fashion in an unattended and hostile environment. The sensor nodes have to work in collaboration to adjust themselves to the distributed algorithm and form the network automatically.

According to [4] WSNs are unique in certain aspects that make them different from other wireless networks and these aspects are summarized as:

- ✓ small CPU with limited computation power,
- ✓ small memory in the node,
- ✓ limited power in the node as it is usually powered by a battery,
- ✓ limited bandwidth and data rate,
- ✓ weak security mechanisms,
- ✓ mobility of nodes,
- ✓ dynamic network topology,
- ✓ communication failures,
- ✓ heterogeneity of nodes,
- ✓ ability to survive in harsh environmental conditions, and
- ✓ ease of use and unattended operation.

2.5 Protocol stack in Wireless Sensor Networks

A protocol, code of conduct [20], is a set of rules that govern a certain behavior, in social or diplomatic activities, at work, when driving, etc. In communication networks, protocols govern, determine the functioning specifications and guidelines, and guarantee how networks fulfill their intended use.

A wireless sensor network is an ad hoc arrangement of multifunctional sensor nodes in a sensor field that are disseminated to gather information regarding some phenomenon. Sensor nodes can be densely distributed over a large and may be remote area and collaborate their efforts to the benefit of the network to the extent that even if a number of nodes malfunction, the network will continue to function. The algorithms developed for wireless ad-hoc networks cannot be used for wireless sensor networks for the following reasons [20]:

- ✓ The number of sensor nodes is typically much more than the number of nodes in a typical ad-hoc network.
- ✓ Sensor nodes, unlike ad-hoc nodes, are prone to permanent failure.
- ✓ Sensor nodes normally use broadcast rather than point-to-point communication with its limited power and memory.
- ✓ Unlike computer networks, sensor nodes do not have global ID since a typical packet overhead can be too large for them.

According to [23] [6] the architecture of protocol stack in WSN can be classified in different layers. In order to get the maximum efficiency with limited resources and low overhead, WSN does not adhere as closely to the layered architecture of OSI (Open System Interconnection) model of conventional network. Nevertheless, the layered model is useful in WSN for categorizing protocols, attacks and defense. So, in contrast to the traditional seven layers it is reduced to the five layers that include physical layer, Data link layer, network layer, transport layer and application later. The advantage of the layered model is conceptually similar functions are combined at one layer.

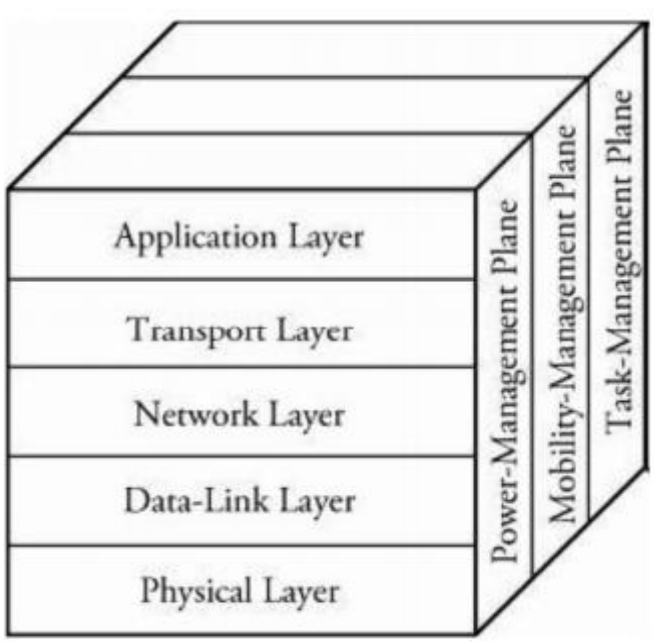


Figure 2.3: Protocol stacks in wireless sensor networks [23]

The physical layer addresses the hardware detail of wireless communication mechanism. This layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption.

The data link layer is concerned with the media access control (MAC) protocol. It ensures reliable end-to-end connections in a communication network. The Data Link Layer is designed to handle error control and the Medium Access.

The network layer manages routing the data supplied by the transport layer or between the nodes. Whereas the transport layer is able to maintain the data flow if the WSN application requires that. Various type of application can be implemented in the application depending on the physical environmental sensing.

Orthogonal to the five layers, there are three management planes; power management plane, mobility management plane, task management plane.

- ✓ The power management plane manages how a sensor node uses its power and also it decides on power consumption rates among three operations: sensing, computing, and communicating.
- ✓ The mobility management plane detects the movement of sensor nodes. It also registers the mobility of sensor nodes. Hence a route back to the user is always kept. Therefore, the nodes can manage their power to complete their tasks by considering this situation.
- ✓ The task management plane organizes the events which are mainly sensing and detecting events from a specific area. Thus, not all of the sensor nodes perform the sensing tasks at the same time and at the same area.

2.6 Advantages and Limitations of Wireless Sensor Networks

The advantages of wireless sensor networks can be summarized as follows [24]:

- ✓ Network setups can be carried out without fixed infrastructure.
- ✓ Suitable for the non-reachable places such as over the sea, mountains, rural areas or deep forests.
- ✓ Flexible if there is random situation when additional workstation is needed.
- ✓ Implementation pricing is cheap.
- ✓ It avoids plenty of wiring.

- ✓ It might accommodate new devices at any time.
- ✓ It is flexible to undergo physical partitions.
- ✓ It can be accessed by using a centralized monitor.

Some of the limitations of wireless sensor networks [25] are hardware constraints, power consumption, node unit costs, environment and standards. Some other limitations of wireless sensor networks are also summarized as follows [24]:

- ✓ Less secure because hackers can enter the access point and obtain all the information.
- ✓ Lower speed as compared to a wired network.
- ✓ More complicated to configure compared to a wired network.
- ✓ Easily troubled by surroundings (walls, microwave, large distances due to signal attenuation, etc.).
- ✓ It is easy for hackers to hack it because cannot control propagation of waves.
- ✓ Comparatively low speed of communication.

2.7 Application of Wireless Sensor Networks

According to [4] the growth of wireless sensor networks was originally motivated by military applications; however, wireless sensor networks are now used in all kinds of civilian and industrial applications. They are meant to identify or observe various physical parameters or conditions including pressure, pollutants, temperature, humidity, air quality, water quality, soil structure, and characteristics of an object in terms of its weight, height, size, position, speed or direction. Wireless sensor networks have an advantage over fixed sensor networks as they can be deployed in hostile territories, battlefields, outer space or under seas, rivers, and oceans.

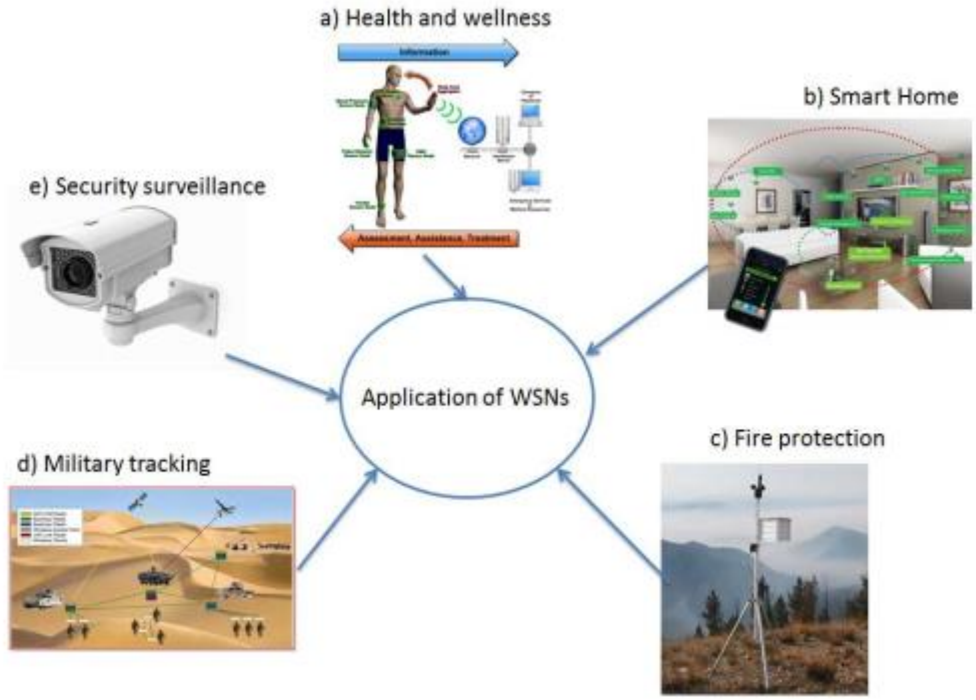


Figure 2.4: Some Applications of WSNs [28]

Military applications: Wireless sensor networks [4] have many applications in the military domain. Wireless sensors can be deployed rapidly in any battlefield or hostile territory without the need for any infrastructure. They can be used for military situation awareness, detection of an enemy unit's movements on land or sea, chemical or biological threats, offering logistics in urban warfare, battlefield surveillance.

Environmental monitoring applications: They can be broadly categorized into indoor and outdoor monitoring [20]:

- ✓ Indoor monitoring applications typically include buildings and offices monitoring. These applications involve sensing temperature, light, humidity, and air quality. Other important indoor applications may include detection of fire and civil structures.
- ✓ Outdoor monitoring applications include chemical hazardous detection, habitat monitoring, traffic monitoring, earthquake detection, volcano eruption, flooding detection and weather forecasting. Sensor nodes have also found their applicability in

agriculture; soil moisture and temperature monitoring is one of the most important applications of WSNs in agriculture.

In environmental applications, sensors are deployed to monitor different environmental factors and conditions, and identify the following cases [4].

- ✓ To observe wild animals or plants in forests and wild habitats as well as to monitor environmental factors of these habitats.
- ✓ To monitor water or air quality by deploying sensors on the ground or under water. Air quality monitoring can be used for air pollution control while water quality monitoring can be used in hydrochemistry areas.
- ✓ Sensors can be deployed in forests to detect fires or in rivers to detect floods. In addition, seismic sensors can be installed in a building to find out the path and degree of earthquakes and offer an estimation of the safety of the building.

Health care applications: In this application, WSNs can be employed to observe and trace patients. Sensors can be installed in patients' homes in order to monitor their behavior or movement; this is especially useful for elderly patients. Such systems can alert the responsible nurse or physician if a patient falls or needs immediate attention. This also can provide real-time health status and updates as well as location of the patients so that they can be reached or saved, if needed. This is achieved by having wearable sensors that can be incorporated into wireless body area networks (WBANs) in order to monitor vital signs in real time [4].

The medical applications of WSNs aim to improve the existing healthcare and monitoring services especially for the elderly, children and chronically ill. Numerous benefits are achieved with these systems [20]

- ✓ Remote monitoring capability; with remote monitoring, the identification of emergency conditions for at-risk patients will become easy and the people with different degrees of cognitive and physical disabilities will be assisted to have a more independent and easy life.
- ✓ Real-time identification and action taking. In healthcare applications, a real-time system is actually a soft real-time system, in which some latency is allowed

- ✓ Being able to identify the context is another benefit achieved with pervasive healthcare systems. Context-awareness enables understanding the conditions of the people to be monitored constantly and the environments in which they are.

Agriculture: Wireless sensor networks are being used more and more in the agricultural industry. Gravity-fed water systems can be observed via pressure transmitters in order to check water tank levels. Pumps can be managed by using wireless I/O devices, and water use maybe estimated wirelessly. Irrigation automation can help in better water consumption and so reduce wastage. In addition, WSNs can be used to control the temperature and humidity levels inside greenhouses so that, when they fall below threshold levels, the greenhouse guard should be notified by using a cell-phone text message or e-mail message [4].

2.8 Localization in Wireless Sensor Networks

Localization is the process of determining the position of unknown sensor node based on the known node position by some mechanism algorithm [27]. The location discovery service enables a device to know its location. The use of additional devices is not suitable for wireless networks, especially the resource-constrained ones, because of the power-hungry nature of such devices. The information provided by wireless sensor networks is highly correlated with respect to space and time. In applications such as target tracking, geographic routing, environmental monitoring, structural health monitoring, and forest fire monitoring, the location of information generating sensors are very important [4]. Location of sensor nodes can be obtained either by placing the sensor nodes at points with known coordinates manually or by deployment of global positioning systems (GPS) on every sensor node. Manual configuration is impractical in large-scale deployment and adding GPS [7] to all nodes in the network is infeasible because of high energy consumption, cost and line-of-sight nature of signal of GPS satellites.

Practically, the use of GPS in WSNs involving thousands of nodes is not feasible because of the following main reasons [4]:

- ✓ GPS receiver is expensive and the cost of deployment of WSN increases if GPS is built inside every node.
- ✓ The use of GPS with every node is not an energy-efficient solution in WSNs, as these networks are intrinsically energy constrained.

- ✓ GPS does not work in indoor environments, and environmental factors such as large buildings affect GPS performance.

For these reasons an alternate solution of GPS is required which is cost effective, rapidly deployable and can operate in diverse environments. It is localization algorithm.

Localization in wireless sensor networks [8] is one of the central components of a variety of emerging applications including military, eHealth, environment monitoring, home and office automation, weather forecasting and so on. Many of these applications need location based services. Although GPS is a direct solution to the localization problem, the high cost, high power consumption, and poor performance of GPS inside an indoor environment have necessitated the research on localization algorithms.

2.8.1 Localization Algorithms in Wireless Sensor Networks

Literatures classify localization algorithms based on different parameters in to different main categories. According to [4], [28], [29] localization algorithms based on computational capability at each anchor are categorized as centralized and distributed.

In centralized algorithms, a single powerful central node calculates the positions of unknown node. An unknown node sends its measured information, such as beacon position and distance to the central node, and the central node sends back the estimated position to the unknown node. This approach involves a lot of communication overhead for the unknown nodes, which in turn, would require these nodes to have more battery power for sending and receiving. The centralized approach also introduces a single point of failure, if the central node fails for some reason, the entire localization process fails.

In distributed localization algorithms all the relevant computations are done on the sensor nodes themselves and the nodes communicate with each other to get their position in a network. In large WSNs, distributed localization algorithms are more efficient than centralized algorithms.

Comparing a distributed localization algorithm with a centralized localization algorithm, a distributed localization algorithm is inherently more robust than a centralized localization algorithm, such as less possible of link failures. Distributed localization algorithms are also far more scalable in practical deployment and may be the only way to achieve the large scales needed for some applications [28].

Based on the number of anchors [28] that is used to localize a node, the localization algorithms can be classified as anchor-based algorithms and anchor-free algorithms. In anchor-based algorithms, several anchor nodes are used to localize nodes with unknown locations. In this scheme, the accuracy of the estimation highly depends on the number of anchors and the performance is improved when more anchors are added to the network. In the anchor-free scheme, there is no anchor node with perfectly known location. Nodes communicate with each other to estimate relative locations instead of computing absolute locations. Comparing the anchor-based scheme with the anchor-free scheme, the anchor-based scheme provides more accurate results than the anchor-free scheme. However, since GPS receivers are expensive, to obtain the accurate anchor locations, the hardware cost for the anchor-based scheme is much higher than the anchor-free scheme.

Based on the location determination mechanism, localization algorithms are broadly categorized in to two classes: range-based and range-free localization algorithms [28], [30], [31], [8].

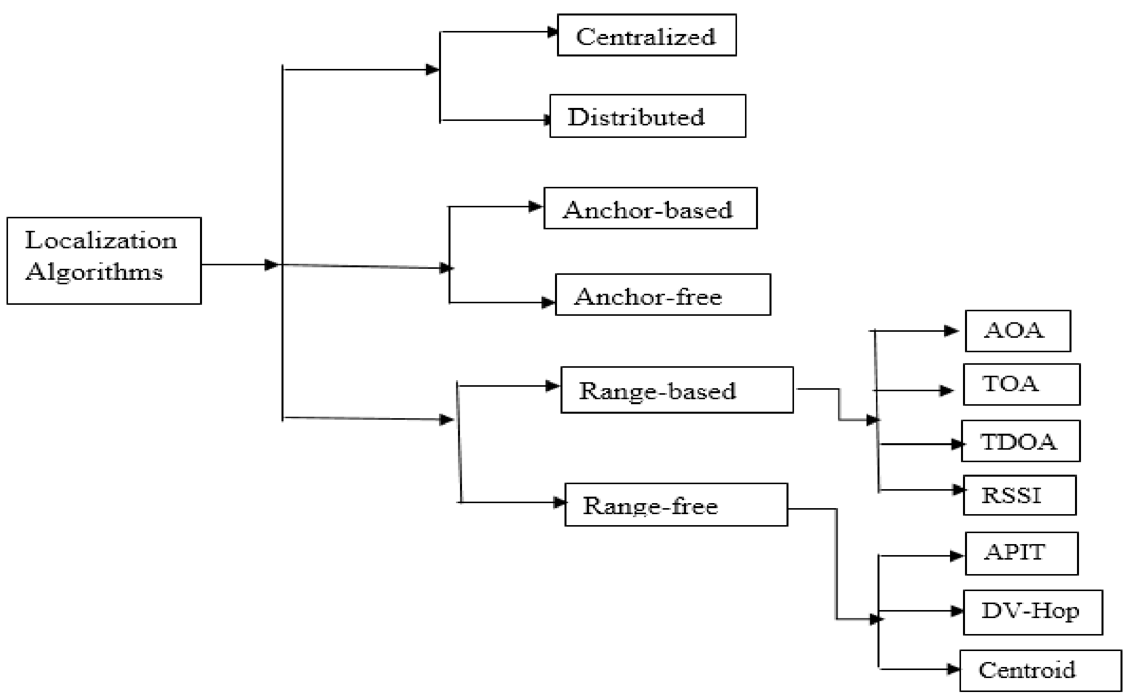


Figure 2.5: Classification of localization algorithms

2.8.2 Range-based localization algorithms

Range-based localization algorithms require more units such as GPS and directional antennas to measure angles, and are defined by protocols that use absolute point-to-point distance (range) estimates or angle estimates for calculating the location. These methods utilize the measurement such as angle of arrival (AOA), time of arrival (TOA), time difference of arrival (TDOA) and received signal strength indicator (RSSI). Normally range-based schemes have higher location accuracy than the range-free localization schemes, but are hardware intensive and high cost of sensor nodes [10], [28], [30].

Angle of arrival (AOA) means the angle at which the receiver received the signal from the transmitter, and is [4] a method in which the angles between unknown node and a number of anchors are used to estimate the location of unknown sensor node. Using AOA, each node can estimate the bearing to the neighboring nodes with respect to its own axis and Bearing is an angle with the respect to another object. AOA use antenna array for detecting angles [21]. The antenna array consists of multiple antennas separated by certain distance. To perform localization with AOA [21], two angle measurements are required, as shown in the figure 2.6. The signal sending from the node M is received by anchor A1 and anchor A2. The antenna array of A_1 can detect the signal's AOA denoted as α, while A_2 can measure the AOA as β. Then the two anchors send the angle information α and β as well as their positions (x_1, y_1) and (x_2, y_2) to M. From the positions of anchors, M can calculate the distance between anchors, denoted as d. Finally, M estimates its positions (x_M, y_M) through the triangulation approach.

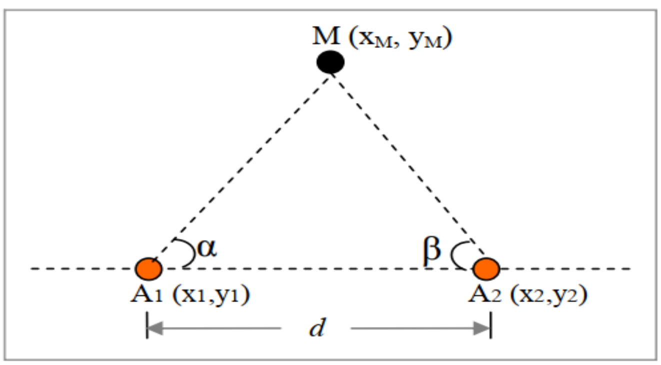


Figure 2.6: Example of AOA localization [21]

AOA has the following two practical problems:

- ✓ It is not practical to implement antenna array on tiny sensor nodes considering the constraints on sensor nodes in terms of size, cost and energy.
- ✓ The accuracy of AOA measurement is affected by a combination of factors including multi-path reflections and background noise, which lead to large errors in angle estimation.

Time of Arrival (TOA): Time of Arrival (TOA) estimates the distance between the anchor node and a particular sensor node based on the time required to propagate and the speed of the transmitted signal [3]. The sensor nodes transmit a signal to their neighbors at a predefined speed and wait for answers. Their neighbors, send a signal back to them. Inter-node distance is calculated by measuring the difference between the time when a signal is transmitted by a node and the time when the same signal is re-transmitted by another node is received at the original node. As TOA is based on the accurate measurement of time at the receiver and transmitter, synchronization has to be considered.

Time Difference of Arrival (TDOA): In the TDOA [4], the propagation time can be directly translated into distance, based on the known signal propagation speed. In this approach, each node is equipped with a speaker and a microphone. In the transmitter initiated approach, the transmitter first sends a radio message. After sending the message, it delays for some fixed interval of time, t_{delay}, and then generates some fixed patterns of sound by its speaker. When a receiver listens to the radio message, it records the receiving time, t_{radio}, and waits for the sound signal. It records the receiving time of sound signal as t_{sound}. The receiver then calculates the distance d from the sender, by using equation [4]. TDOA able to achieve high ranging accuracy but need extra hardware and consume more energy.

$$d = (S_{radio} - S_{sound}) \times (t_{sound} - t_{radio} - t_{delay}) \quad \text{(EQ.2.2)}$$

Where S_{radio} and S_{sound} are the speed of radio signal and sound signal, respectively.

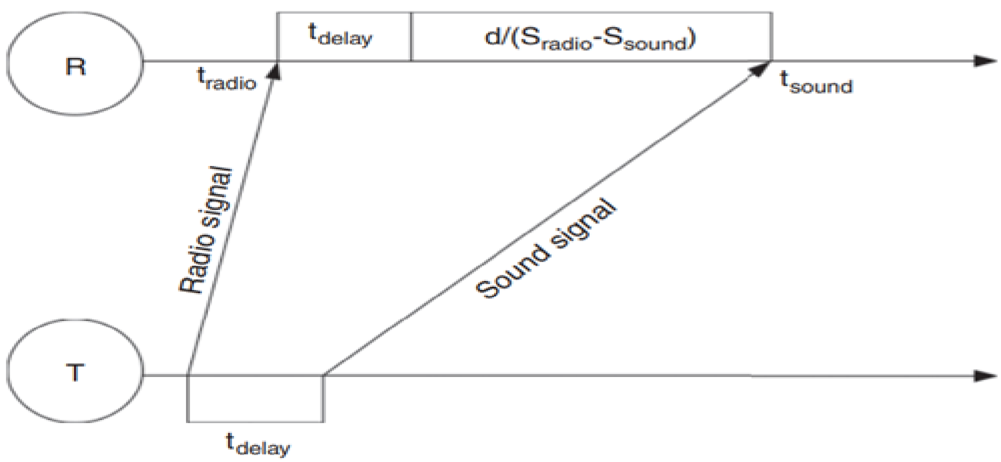


Figure 2.7: Example of time difference of arrival (TDOA) [4]

Received signal strength indicator (RSSI): Received signal strength indicator [21] is considered as the most popular technique for range estimation in wireless sensor networks. That is, because almost every sensor node in the market has ability to analyze the strength of a received message, then every RSSI information can be obtained at most with no additional cost. The intensity of an emitted signal decreases as the distance from the emission source increases .This decrease relative to the original intensity is attenuation. The signal strength decays with respect to distance in a polynomial manner. In the most ideal circumstances, signal power attenuation is proportional to $d^{2,}$ where d denotes the distance between the transmitter and the receiver. This effect is sometimes referred to as free space loss.

RSS [33],[42] transforms radiated signal strength to a meaning full distance using models of wireless channels and it is commonly made in both theoretical studies and experimental studies, the power loss can be formulated by using the lognormal model shown in EQ.2.3, which is the widely used radio propagation model in the absence of multipath effects. Given a function correlating attenuation and distance [21], it is possible to estimate the distance between two nodes by measuring the strength of the signal. The widely used radio propagation model is the log-distance path loss model (without multipath effects):

$$pt = pl(do) - 10 \times \mathrm{n} \times \log_{10} \frac{d}{do} \qquad \textit{(EQ. 2.3)}$$

In Equation EQ.2.3 pt is measured in dBm, which is a logarithmic measurement of transmit signal power. *d* is the distance between emitter and receiver, pl(do) is the signal power value at reference distance *d*o, *n* is the attenuation constant (rate at which the signal decays). Usually, *n* is obtained through empirical data, n is around 2 in a free-space environment, but its value increases if the environment is more complex (walls, large metallic objects, etc.). In environments with many obstructions such as an indoor office space, an approximation of *n* is between 3 and 6. The model of RSS measurement is shown in the figure 2.8

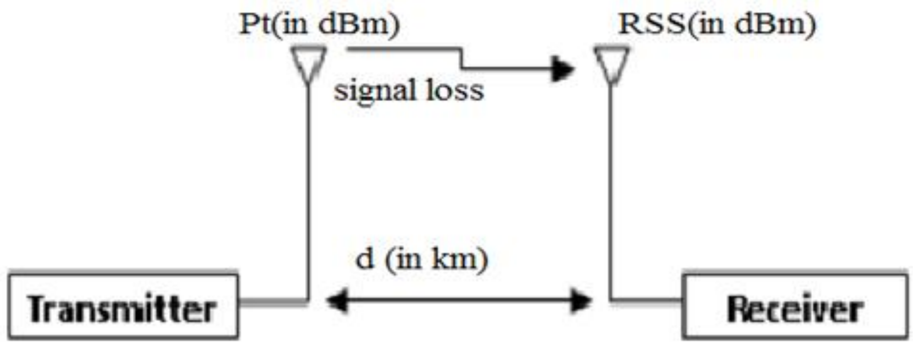


Figure 2.8: The Model of RSS measurement modified from [33]

The above RSS model is given by the following relation;

RSS = Transmitted power – Path loss (signal loss), i.e.

$$RSS = pt - pl(do) - 10 \times \mathrm{n} \times \log_{10}\frac{d}{do} \qquad \textit{(EQ. 2.4)}$$

Where;

pt is transmitting power,

pl(do) is the path loss for a reference distance of do,

n is the path loss exponent,

d is the distance between sender and receiver node.

Based on EQ.2.4 a commonly used model for calculating the distance *d* is given in (EQ.2.5) which pl(do) is measured at do = 1m.

$$d = do \times 10^{\frac{pt-pl(do)-RSS}{10n}} \qquad \textit{(EQ. 2.5)}$$

The drawbacks of range based [21] localization algorithms:

- The range information is very easily affected by multipath fading, noise and environment variations for the case of RSSI.
- Additional ranging devices are needed, which consume more energy and increase the overall cost for the case of AOA,TOA and TDOA

2.8.3 Range-free localization algorithms

The Range-free localization algorithms need less sophisticated hardware. They use estimated distance instead of metrical distance to localize, so they have much advantage in power and position cost. The range-free localization algorithms are cost effective due to the fact that no distance or angle measurement between communicating nodes is required. As far as the tradeoff between location precision and deployment cost is concerned, range-free localization schemes are more suitable for WSNs than the expensive range-based ones [4].

In range-free localization schemes; centroid localization algorithm (CLA), APIT, and DV-Hop are the popular localization algorithms [12].

DV-hop Localization algorithm:- DV-hop is a range free localization algorithm based on hop count method in which [15], [34] the anchor broadcasts a packet in its transmission range in the network that contains the coordinate data of this anchor node in addition to a hop count with initialized value equals 1. The packet is broadcasted in the network and the hop-count increment by 1. The receiver saves the packet with small number of hops and discards the packet with a large number of hops.

The DV-Hop implementation includes three steps.

- First, every anchor broadcast its coordinate data and hops count in the network, which ensure that all anchor should get the location information and minimum hop to all anchor node in the network topology.

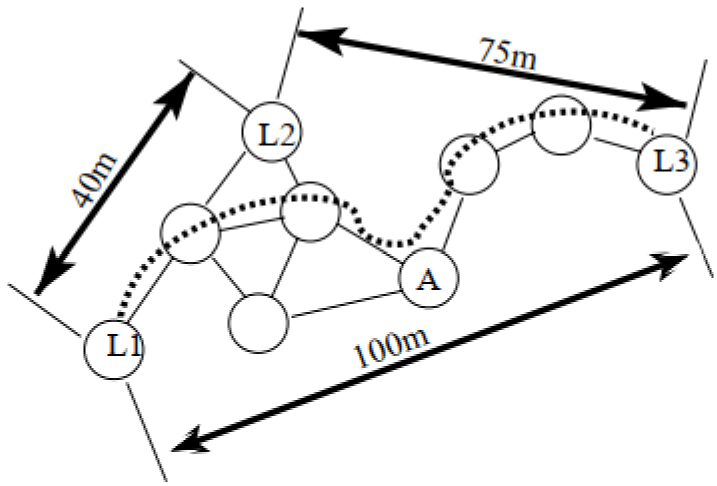


Figure 2.9: Example of average distance per hop (ADH) in DV-HOP implementation

- Second, each anchor node computes the average distance per hop using equation EQ.2.6

$$ADH = \frac{\sum j \in S, j \neq i \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum j \in S, j \neq i, h_{i,j}} \quad (EQ.2.6)$$

where (x_i , x_j) and (y_i , y_j) are the actual and calculated coordinates of anchor i and j, hi,j is the hop metric value between two anchor nodes i and j and S represents the set of anchors. Then each anchor broadcast their average distance of hop (ADH) in its transmission range. At the end of this step, each beacon node contains a routing table include the destination node ID, x coordinate, y coordinate and minimum hop count, so that each beacon node contain the information of all beacons in the network. The unknown node receives the first ADHs from the beacons and discards the later ones. After that, the unknown node uses the received ADHs to calculate the distances to the first three anchors by multiplying ADH with a number of hops to its anchor.

In the figure 2.9, L1, L2 and L3 are anchor nodes and A is an unknown node and L1 has both the Euclidean distance to L2 and L3, and the path length of 2 hops and 6 hops respectively. L1 computes its ADH as (100+40)/(6+2) = 17:5, which is the estimated average size of one hop, in meters. In a similar manner, L2 computes ADH as (40+75)/(2+5) 2+5 = 16:42 and L3 ADH is also computes its ADH as (75+100)/(6+5) = 15:90. Then after the anchors broadcast their respective ADH in the network and node A gets an update from one of the anchors, and it is usually the closest one, that is L2 in this case which is 2 hops from A. Now the ADH of L2 is taken as ADH of the entire network. Node A estimate distance to the three anchors as: to L1; 3×16.42 = d_1, to L2; 2×16:42 = d_2, and to L3; 3×16:42 = d_3.

- Third, the unknown nodes positions are computed using trilateration method as follows:

$$d_1^{\ 2} = (x - x_1)^2 + (y - y_1)^2$$
$$d_2^{\ 2} = (x - x_2)^2 + (y - y_2)^2 \qquad \textit{(EQ. 2.7)}$$
$$d_3^{\ 2} = (x - x_3)^2 + (y - y_3)^2$$

Centroid Localization Algorithm (CLA): - The centroid localization algorithm [44], is range-free localization algorithms and its core idea is to use the connectivity relationships among nodes to calculate the unknown node's position information. In this algorithms, the anchor periodically broadcasts its coordinates information, which contains anchor's ID and location information, to the neighboring unknown nodes. An unknown node locates [45] its position using the intersection of the connectivity of the regions as defined by the radio range of each anchors. The estimate location of the unknown node is the centroid of the polygon which is formed by several anchor nodes. The centroid localization algorithm [34] uses the location (x_i, y_i) of anchor nodes participated in localization of unknown node position and mathematical expressed as follows:

$$(x_{est}, y_{est}) = (\frac{x_1 + ... + x_N}{N}, \frac{y_1 + ... y_N}{N}) \qquad \textit{(EQ. 2.8)}$$

Where $N \geq 3$, $x_{est}\, y_{est}$, is the estimated coordinate of the sensor node be localized.

As a fundamental similarity between CLA and APIT algorithms [45], they are both proximity based algorithms. While centroid localization algorithm assumes equal weight to the received signal strength, APIT through a process, weighs the quality of the received signal strength within the coverage area. APIT uses the best weighted triangles to estimate the most dominant region.

2.8.4 More on Approximate-Point-In-Triangle (APIT) localization algorithm

APIT [35] [29] is a distributed, area-based range-free localization algorithm and segments the whole area into triangular regions among anchors. A sensor node may be present inside or outside these triangles. A sensor node can estimate its location by narrowing down the area of intersection of triangles in which it possibly reside.

The method used to estimate the possible narrow area in which the sensor node can be found is known as point-in-triangle (PIT) test. In this test a sensor node chooses any three anchors and tests whether it is inside the triangle made by chosen anchors or not, by comparing the signal strength of neighbor sensor node. APIT repeats PIT test with different combinations of neighboring anchors until all the combinations are exhausted or the desired accuracy is achieved. After PIT testing, APIT calculates the center of gravity (COG) of the intersection of all the triangles to estimate the unknown sensor node position.

The basic pseudo code [36] for APIT algorithm:

```
    Receive location beacons (xi,, yi) from N anchors.
  InsideSet = Φ // the set of triangles in which an unknown node i reside
    For (each triangle Ti∈ (N 3) triangles) {
        If (Point-In-Triangle-Test (Ti) ==TRUE)
            InsideSet = InsideSet ∪ {Ti}
        If (accuracy (InsideSet)> enough) break;
    }
  /* Center of gravity (COG ) calculation */
Estimated Position = COG(∩Ti ∈ InsideSet);
```

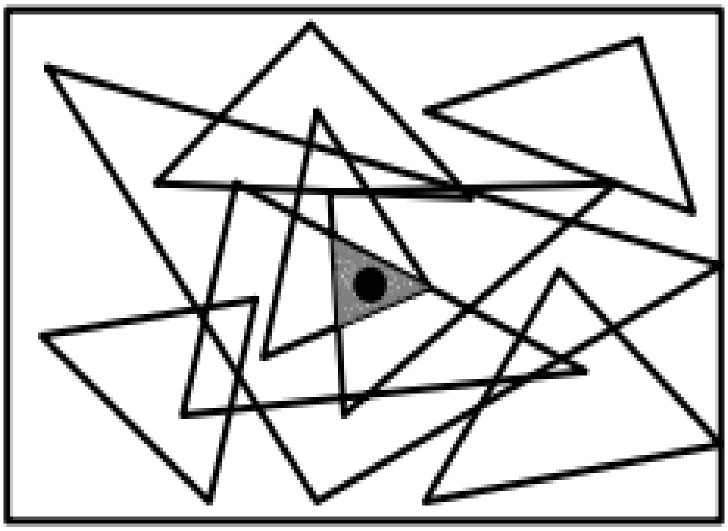


Figure 2.10: APIT Localization Algorithm overview [36]

According to [14], [15], [16], [17], [18] APIT localization algorithm has four phases:

Phase 1: A sensor node receives messages from all anchors where the distance between the sensor node and anchor is less the communication range of a sensor node.

- After receiving messages from anchors, each sensor node maintains a table where entries are anchor ID, anchor position and received signal strength.
- A sensor node exchanges the table with all neighboring sensor nodes and builds a supper table using the information from all neighbors.

Phase 2: PIT testing is done for predefined number of anchor combinations.

The principle of the PIT [14] is, for any triangle ABC made by anchor nodes A, B and C and M is an unknown sensor node on the two dimensional plane, if there exists direction the unknown node move along, and the unknown node moves close to one of the anchors A, B and C, it will be far away from the other two anchors, then we can deduce that the unknown node M is inside triangle ABC. On the other hand if the unknown node M moves along a direction far away from the three anchors A, B and C simultaneously, then we can deduce that M is outside triangle ABC. This is named as perfect PIT (PPIT) test. However, the nodes are deployed randomly in WSN and most of them are static, so the nodes cannot move.

To perform PIT test without the need of node movement [19], approximate PIT test method has been proposed that takes advantage of high node density in WSNs. To emulate the movement of a node in the PPIT, node uses neighbor information, exchanged via received signal strength (RSS).

If neighbor of node M is closer to one of the three anchors and far away from the remaining other two anchors, then it is deduced that node M is inside triangle ABC formed by the three anchors A, B and C. On the other hand, if neighbor of node M is closer to or far away from the three anchor nodes A, B and C simultaneously, then it is assumed that M is outside triangle ABC.

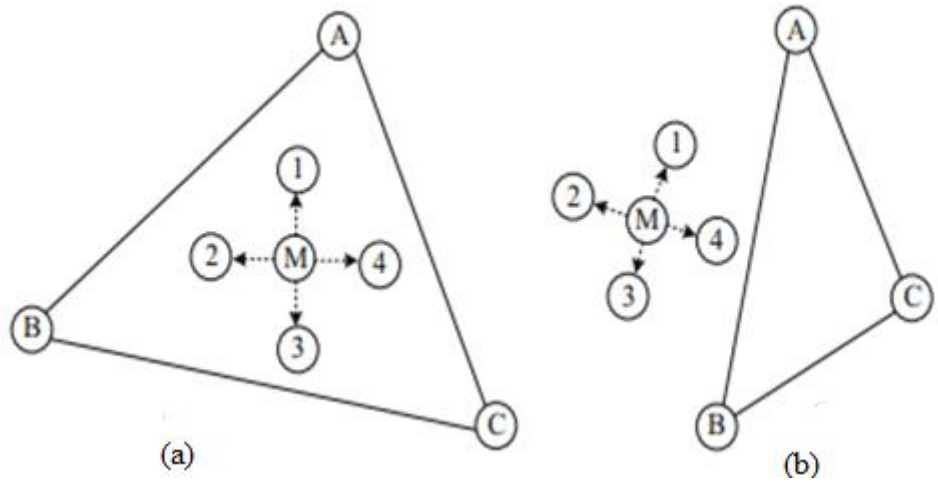


Figure 2.11: Approximate PIT test, (a) inside scenario, (b) outside scenario

Phase 3.This step is called APIT aggregation in which the intersection area of all the triangles, estimated at step 2 found.

Phase 4. The center of gravity (COG) of the intersection of all triangles is calculated to estimate the sensor node position.

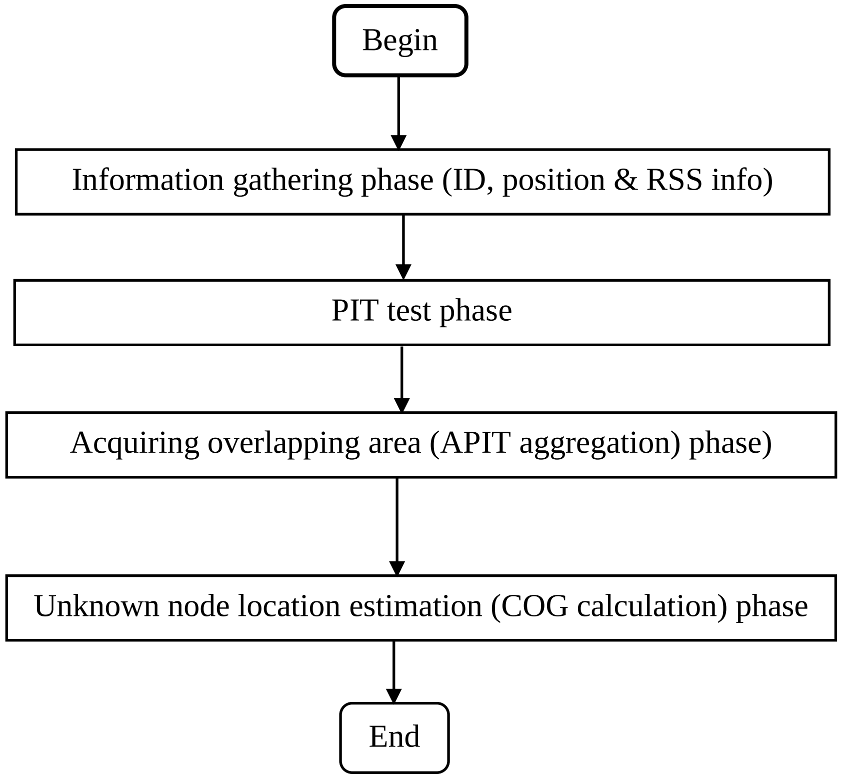


Figure 2.12: The Main Phases of APIT Localization Algorithm

As discussed in literatures a key step in APIT algorithm [12], [14], [16],[17], [35] is PIT testing, which check whether the unknown sensor node is inside or outside triangle formed by three neighboring anchors. If the unknown sensor node is out of the triangle, then it is not localized.

2.8.4.1 Shortcomings of APIT Localization Algorithm

Mainly there are two inevitable errors that are based on PIT test of APIT localization algorithm. The two scenarios where incorrect decisions are made are illustrated in figure 2.13 .Let M be a target unknown node and let A, B and C be anchor nodes that form triangle ABC. As shown in figure 2.13 (a), M is actually inside the triangle ABC, but conclusion on the PIT test that it is outside the triangle ABC and this is called In-to-Out errors. This can happen when M is near the edge of the triangle, while some M's neighbors (node 4 in this case) are outside the triangle and further from all vertices of the triangle ABC, in relation to node M. As a result, M mistakenly thinks it is outside of the triangle ABC due to this edge effect.

On the Other hand, the irregular placement of neighbors can result in Out-to-In error. Figure 2.13 (b) depicts the scenario where M is outside of the triangle ABC and some of its neighbors is closer to one vertex of triangle ABC and further from the remaining two vertices. Neighbor node 2 is closer to anchor A, and far away from anchor B and C, neighbor node 3 is closer to anchor B and far away from anchor A and C. This makes M mistakenly assume it is inside triangle ABC.

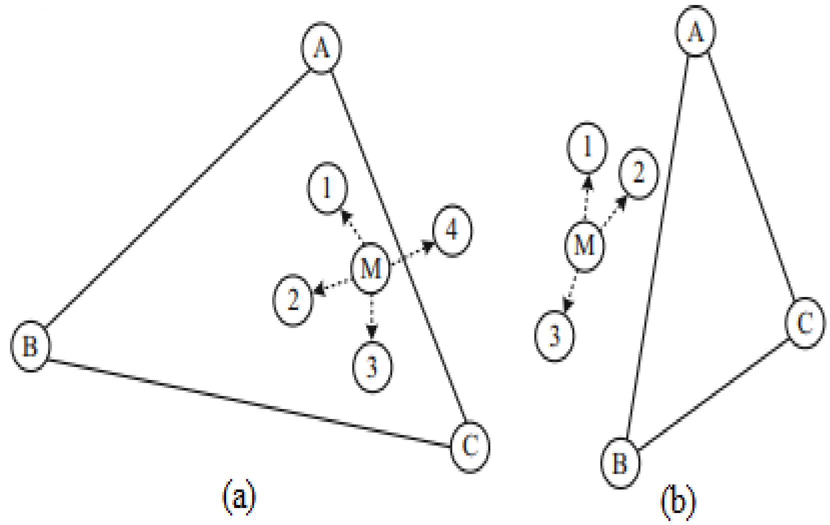


Figure 2.13: Error Scenarios of the APIT test, (a) In-to-Out error, (b) Out-to-In error

2.9 Summary

This chapter provided an overview of WSNs and description of architecture of WSNs, components of sensor node, characteristics of WSNs, protocol stack of WSNs. The application of WSNs in different areas was also reviewed. The concepts of localization and localization algorithms and related issues were introduced in order to provide the theoretical background knowledge that is necessary for the study area.

Chapter Three: Related Work

In this chapter, we summarize related works that are published on the improvement of APIT localization algorithm of wireless sensor networks. Finally we conclude the chapter by showing the gaps and the alternative improvement of the APIT algorithm intended by this thesis work.

Literatures show that there have been a number of works that have done on the improvement of rang free localization algorithms. Here we have summarized works those that are relevant and related to our study.

In the work [14] an improved APIT algorithm HTCRL (Homothetic Triangle Cyclic Refinement Location) is proposed, which is based on the principle of the homothetic triangle. According to this work two triangles are homothetic if a pair of corresponding angles is equal and the corresponding edges are proportion. Another definition given in the paper that related with homothetic is center of triangle. The angular bisectors of the three angles of a triangle meet at a point. The point is the center of a triangle's inscribed circle, which is called the center of a triangle.

The proposed improved APIT localization algorithm adopts the homothetic triangles to narrow down the target area. According to the theory of homothetic triangle, the area of the triangle that hold the unknown node will be narrowed down gradually. This improved algorithm has six steps. First initialize the network configuration. Second each anchor node in the area broadcasts the messages including their own location information and ID. The nodes (includes other anchor nodes and all unknown nodes) collect the messages (ID, position, RSS) and exchange the message with their neighbor unknown nodes. Third the unknown nodes judge whether they have received the message from more than three anchor nodes. If the number is more than three, it will go to PIT test stage, if the number is less than three, it will go to end. Forth PIT test stage. It will judge whether they are inside the triangle composed by the three neighbor anchors. If the unknown node is inside the triangles, HTCRL algorithm begins to narrow down these triangles until the localization error rate is satisfied for the applications. Fifth acquire triangles overlapping area. Sixth, use the homothetic triangle cyclic refinement algorithm to calculate the centroid of the overlapping area in fifth step which is the estimated coordinates of the unknown nodes. This research work took all APIT algorithm procedures as they are and add HTCRL in a place of the center of gravity (centroid of overlapped area of polygon) step, but the key procedure for APIT algorithm, the possible errors associated with PIT test is not taken into consideration.

Authors in [18] propose improvement on APIT localization algorithm by improving judgment accuracy using the method of angle judgment as PIT test. According to this work, if the unknown node is in the triangle say triangle ABC, then the sum of the degree measure of the angles formed between the line-segments drown from the three vertices of a triangle ABC to an unknown node is equal to 360°. Otherwise, it is less than 360°.

In the work [32] the authors introduced RSSI to revise the location accuracy so that to solve the problem that exists in APIT localization algorithm and the authors suggested that to read the signal power of the receiving node from the CC2431 RSSI register. To do this, two main stages were considered. First for the anchor nodes within one unit of the unknown node, use RSSI to measure the distances between the unknown nodes and the anchor nodes, select three anchor nodes with smaller distances, and calculate the estimated position 1 by trilateration measurement. Second according to APIT algorithm, the center of mass of the overlapping region was calculated as estimated position 2. Finally the average value of these two positions was calculated as the final position coordinates. For those unknown nodes that are unable to be located by APIT algorithm, the authors proposed RSSI ranging quantitative model to find the estimated coordinates of the unknown node. Experimentation was used for the verification of the proposed APIT localization algorithm.

Fanzhen *et al*. [16] propose improvement on APIT localization algorithm based on cosine theorem according to a basic idea of APIT to prevent In-To-Out and Out-To In errors and address RSSI model to locate the unknown nodes which are judged outside of the triangles through trilateral measurement method to improve localization coverage.

To apply cosine theorem angles formed by the line segments joining each vertices of the triangle formed by the three anchors say A,B and C , and the target point (target node) were considered and by finding the cosine value of each angles and then for the negative value , the angle will be obtuse. According to this work if there are two or three angles are obtuse, and then it determines that the target node is inside the triangle. Otherwise, the target node is outside of the triangle.

In our solution we intended to fulfill the following issues which are not given focus in reviewed works.

Design new and simple PIT test method which is a key procedure in APIT localization algorithm and take into consideration computational complexity of the algorithm.

Chapter Four: Design of the Proposed Solution

4.1 Introduction

As described in previous chapter, the key procedure in APIT localization algorithm is PIT test and the main shortcoming of the algorithm is also associated with PIT test in which wrong judgments are made on the presence of unknown node inside triangle formed by three anchor nodes or outside the triangle. In this work we propose new and simple approach on PIT test which is based on perimeter of triangles formed from three anchor nodes and the distance between the target node and the three anchor nodes (vertices of the triangle formed by three anchors).

Following this introduction, in section 4.2 we present mathematical model in which one theorem from plane geometry which is helpful for the proposed IM_APIT localization algorithm is stated and proved, and the detail application of the theorem as a mechanism for PIT test is discussed. The flow description of the proposed improved APIT (IM_APIT) localization algorithm is given in section 4.3. Finally section 4.4 summarizes the chapter.

4.2 Mathematical Model

To minimize probability of wrong judgment that unknown node whether it is inside triangle composed by three anchor nodes or not, which leads to In-To-Out and Out-To-In errors of APIT localization algorithm, modifying the condition of point-in-triangle test (PIT test) is point of improvement of the APIT localization algorithm in this thesis. To do this, the following theorem from the concept of plane geometry is set and proved. This theorem is very helpful for the proposed improvement APIT localization algorithm as PIT test to minimize wrong judgement of unknown node to be inside triangle formed from three anchor nodes or to be outside.

Theorem: If the sum of distance from target node M to vertices of the triangle ABC (anchor nodes A, B and C) is greater than perimeter of the triangle ABC, then the target node M is outside triangle ABC. Otherwise M can be inside triangle ABC

Proof

To prove this theorem, we want to show the relation between the sum of distance from a point to the three vertices of a triangle and perimeter of a triangle.

To do this we consider different scenarios of position of a point with respect to ΔABC.

As a notation, let dist = sum of distances from a point to the three vertices of a triangle and let perimeter = perimeter of a triangle ABC.

Scenario i. Let M be a point inside ΔABC.

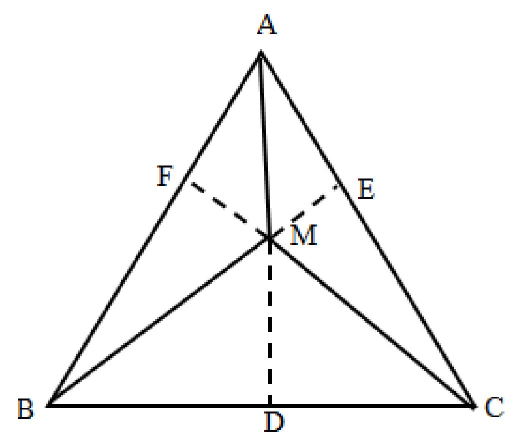
Let's consider the following construction.

Construct line segments AM, BM and CM

Extend line segment AM and let it meet BC at D,

Extend line segment BM and let it meet AC at E

Extend line segment CM and let it meet AB at F



1) BE < AB+AE …………triangle inequality (in ΔABE)

2) MC < ME+EC ………triangle inequality (in ΔCME)

3) BM+MC < AB+AC ……… from step 1 and 2

4) AD < AC+CD ………triangle inequality (in ΔACD)

5) MB < MD+BD ……..triangle inequality (in ΔBDM)

6) AM+MB < AC+BC ……from step 4 and 5

7) CF < BC+BF ….....triangle inequality (in ΔBCF)

8) AM < AF+MF ……..triangle inequality (in ΔAFM)

9) AM+CM < AB+BC ….from step 7 and 8

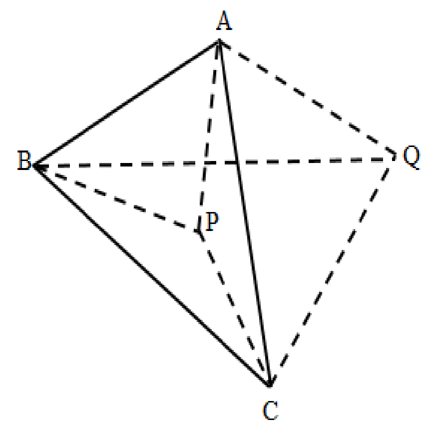
10) 2(AM+BM+CM) < 2(AB+BC+AC) ….from step 3, 6 and 9

11) AM+BM+CM < AB+BC+AC…step 10

Therefore the sum of distance from vertices of a triangle to a point inside the triangle is less than perimeter of the triangle, i.e. dist < perimeter.

Let's consider an arbitrary point in the plane of a triangle ABC.

Let P and Q be points in the plane of ΔABC.



Scenario ii. Let's consider point P and construct line segment AP, BP and CP

1) AP+BP > AB …. Triangle inequality (in ΔABP)
2) AP+CP > AC …. Triangle inequality (in ΔACP)
3) BP+CP > BC …. Triangle inequality (in ΔBCP)
4) 2(AP+BP+CP) > AB+AC+BC ….. from step 1, 2 and 3
5) AP+BP+CP > $\frac{1}{2}$(AB+AC+BC) … from step 4

Therefore the sum of distance from vertices of a triangle to a point inside the triangle is greater than semi-perimeter of the triangle, i.e. dist > $\frac{1}{2}$ perimeter.

Scenario iii. Let's consider point Q and construct line segment AQ, BQ and CQ

1) AQ+BQ > AB …. Triangle inequality (in ΔABQ)
2) AQ+CQ > AC …. Triangle inequality (in ΔACQ)
3) BQ+CQ > BC …. Triangle inequality (in ΔBCQ)
4) 2(AQ+BQ+CQ) > AB+AC+BC ….. from step 1, 2 and 3
5) AQ+BQ+CQ > $\frac{1}{2}$(AB+AC+BC) … from step 4

Therefore the sum of distance from vertices of a triangle to a point outside the triangle is greater than semi-perimeter of the triangle, i.e. dist > $\frac{1}{2}$ perimeter.

From the proof, the following points can be deduced:

1. The sum of distances from a point inside a triangle to its vertices is less than perimeter of the triangle.

   dist < perimeter

2. The sum of distances from a point inside a triangle to its vertices is greater than semi-perimeter of the triangle.

   dist > $\frac{1}{2}$ perimeter

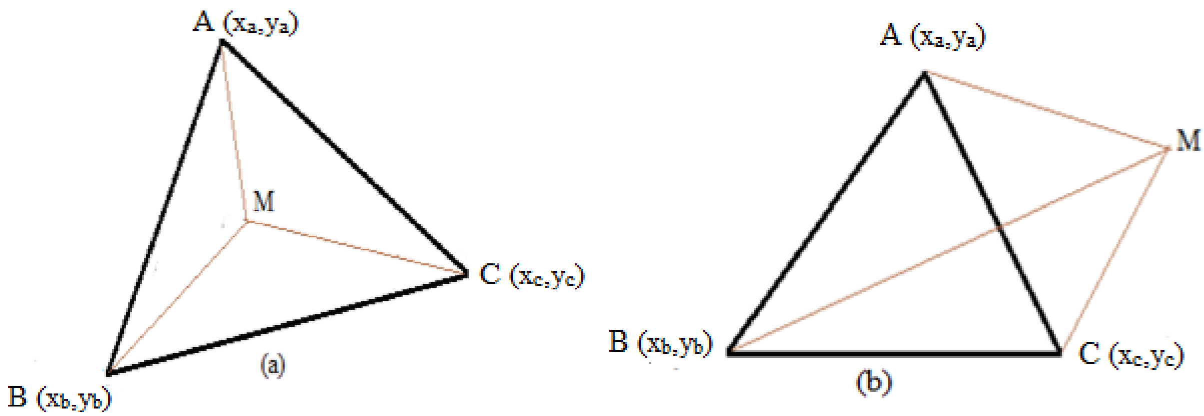
3. The sum of distances from a point outside a triangle to its vertices is greater than semi-perimeter of the triangle.

   dist > $\frac{1}{2}$ perimeter

From 1 and 2 possible inside scenario, we can have; $\frac{1}{2}$ perimeter < dist < perimeter, it mean that if dist is greater than semi-perimeter and less than perimeter, then the point is inside the triangle. But from the scenario 3 the point have possibility to be outside.

Therefore we conclude that if the sum of distance from a point to vertices of the triangle is greater than perimeter of the triangle, then the point is outside a triangle without any possibility to be inside.

To make clear this idea, let A, B and C be three anchor nodes, as vertices of triangle ABC and let M be the target node as shown in figure 4.1.



(a) Inside scenario (b) Outside scenario

Figure 4.1: The New PIT test scenario

To apply this theorem, first we need to compute distances mentioned in the theorem. The distance between three anchor nodes A, B and C (AB, BC and AC) can easily be calculated using the following Euclidean distance formula of two dimensional plane since the anchor nodes are aware of their position.

$$\begin{cases} AB = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \\ BC = \sqrt{(x_b - x_c)^2 + (y_b - y_c)^2} \\ AC = \sqrt{(x_a - x_c)^2 + (y_a - y_c)^2} \end{cases} \qquad \textit{(EQ. 4.1)}$$

Where, (x_a, y_a) is x,y coordinate of anchor node A, (x_b, y_b) is x,y coordinate of anchor node B, and (x_c, y_c) is x, y coordinate of anchor node C.

The value of AB+BC+AC is equal to perimeter of the triangle ABC.

To compute distance between target node M and each vertices of triangle ABC (or anchor node A, B and C) the concept of RSSI techniques is used.

Assuming that the received signal strength (RSS) a target node M received from anchor node A, B and C be RSS_{AM}, RSS_{BM} and RSS_{CM} respectively are known and then the corresponding distance will be estimated by converting the RSS value found and using the parameters of the RSS model by the formula shown in *EQ. 4.2*.

$$d = do \times 10^{\frac{pt - pl(do) - RSS}{10n}} \qquad \textit{(EQ. 4.2)}$$

Typical values of these parameters are:

Pt = 0 to 4dBm (max),

pl(do) = 55dBm at do = 1m ,

n = 2 to 4.

Then according to the theorem set for the improvement of APIT localization algorithm ,the key procedure of the algorithm; Point-in-Triangle test (PIT test) is performed by comparing the value of perimeter of the triangle ABC and the sum of distances from target node M to the vertices of triangle ABC (anchor nodes A, B, C),i.e. if AB+AC+BC >AM+BM+CM then it can be deduced that a target node M is reside inside triangle ABC, otherwise M reside outside triangle ABC.

4.3 The Flow Description of the Proposed Improved APIT (IM_APIT) Algorithm

Improved APIT (IM-APIT) localization algorithm based on modifying mechanism of PIT test (point-in-triangle test) proposed in this thesis work can reduce probability of wrong judgment of unknown node whether it is inside triangle composed of three anchor nodes or outside and accordingly decreases localization error rate efficiently. In APIT localization algorithm among its four major phases, the second phase, PIT test is the core part of the APIT localization algorithm which has a great impact on its performance.

The main process of IM_APIT localization algorithm is shown in figure 4.2 in which each steps are performed at individual nodes in a purely distributed fashion.

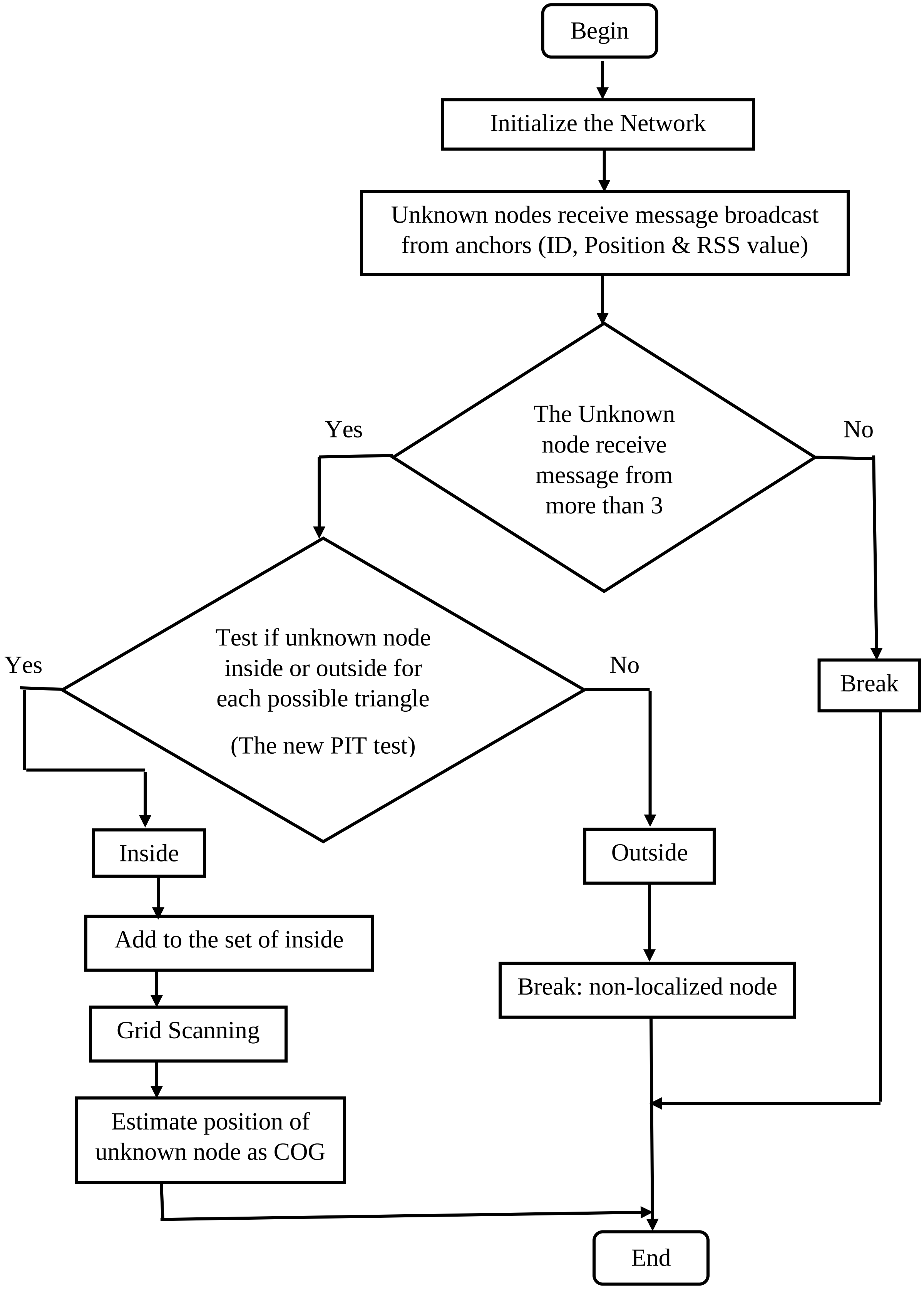


Figure 4.2: The proposed IM_APIT Localization Algorithm flow chart

Step 1. After having completed the deployment of WSN, the network is initialized.

Step 2. All of the anchor nodes in the network area start broadcasting messages. The messages contents are the unique identifier of the anchor node (Node ID), the location coordinates of the anchor node and the received signal strength (RSS) information of anchor node when accepting the messages. Unknown nodes store the messages after received them from the anchor nodes and then exchange the message with their neighbor unknown nodes.

Step 3. The unknown nodes judge whether they have received message from more than three anchor nodes. If the number of anchor nodes is more than three, then it will go to the PIT test phase (step 4), if the number of anchor nodes is less than three , it will go to end step.

Step 4. Point-in-triangle (PIT) test stage. In this step unknown nodes will judge whether they are inside the triangle composed by the three neighboring anchor nodes. This is done by the theorem set as PIT test of IM_APIT localization algorithm. According to the theorem, assuming A, B and C are anchor nodes and let M be unknown node, if perimeter of triangle ABC (AB+BC+AC) is less than AM+BM+CM ,then unknown node M is judged to be outside triangle ABC. On the other hand if semi-perimeter of triangle ABC is less than the sum distance from target node M to vertices of the triangle ABC (anchor nodes A, B and C) and it is again less than perimeter of the triangle ABC, then target node M has high possibility to be inside the triangle ABC. If M is outside of triangle ABC, it will go to end. If M is inside triangle ABC then it is added to the set of inside and go to step 5. This method applied repeatedly with different combinations of three anchor nodes until all combinations are tried.

Step 5. Grid Scanning: Once the individual PIT test is finished, APIT aggregates the results (inside or outside decision) through grid scanning. In grid scanning algorithm the area in which the unknown node distributed is divided into grids and a grid array is used to represent the maximum area in which an unknown node is likely reside. For each PIT-test inside decision, the values of the grid regions over which the corresponding triangle resides are incremented. Assuming count is the number of each grid covered. And the initial value is 0. Each determines node in triangle, the value of grid's count will be p1us 1 in the corresponding area, and the largest value as shown in Figure 4.3, is 2, indicate that maximum overlapped and narrow area that used to estimated location of unknown nodes by calculating COG of most overlapped area. The basic idea of Grid scan algorithm [38] is that dividing the entire network into several grids according to certain steps, and using the APIT algorithm to determine the relationship between the unknown nodes and the triangles. If the unknown node is inside the triangle, the value of all the grids in the triangle is plus 1; and if the unknown node is outside the triangle, the value of all the grids in the triangle is minus 1. After traversing all the triangles, the area with the maximum value is the area where the node is most likely to reside.

The pseudo code for APIT aggregation [36] is as follows:

```
For (each triangle Ti∈ (N 3) triangles) {
        If (APIT (Ti) == Out)AddNegtiveTriangle(Ti);
      If (APIT(Ti) == In)AddPositiveTriangle(Ti);
  };
```

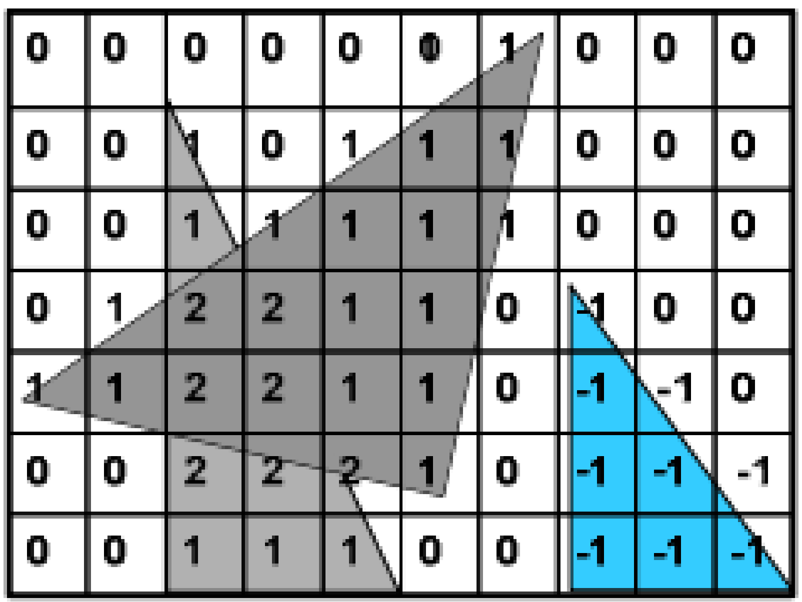


Figure 4.3: Grid Scan approach [36]

Step 6. The APIT algorithm calculates the Centre of Gravity (COG) of the intersections of all the overlapped triangles in which the node resides to determine its location.

From step 5, grid scanning we can find the maximum count of several grids and then calculate the centroid of the polygon formed by the grids in formulas shown in EQ.4.3 and EQ.4.4.

$$x = \frac{x_1+x_2+x_3+\cdots+x_n}{n} \qquad \textit{(EQ. 4.3)}$$

$$y = \frac{y_1+y_2+y_3\ldots+y_n}{n} \qquad \textit{(EQ. 4.4)}$$

Where, in EQ.4.3 and EQ.4.4 *n* represents the number of vertices of the polygon of which going to be calculated , x represents the horizontal coordinate of the unknown node , y represents the vertical coordinate of the unknown node and then accordingly (x_i, y_i) for i =1,2,3,….,*n* represents the coordinates of the girds with the maximum count.

4.4 Summary

In this chapter, we described one theorem from the concept of plane geometry that is helpful for the proposed Improved Approximate-Point-in-triangle (IM_APIT) localization algorithm which is directly used as PIT test and the detail discussion of its application is presented as mathematical model. Following this, we described RSS model for meaningful estimation of distance between anchor nodes and unknown nodes. Lastly, we presented the flow description of the proposed IM_APIT localization algorithm.

Chapter Five: Prototype Implementation and Evaluation

5.1 Introduction

Approximate-Point-In-Triangle (APIT) localization algorithm is a range free area based localization algorithm in wireless sensor network and it performs location estimation by dividing the environment into triangular regions among anchor nodes in which an unknown node's presence inside or outside these triangular regions allows this unknown node to narrow down the area it can reside and the main shortcoming of APIT rises from the method of judging whether the unknown node reside inside or outside the triangular region. To minimize the probability of wrong judgement, we implement a new and simple Point-In-Triangle (PIT) test method to improve APIT localization algorithm.

To accomplish this task, the development environment, programming language and simulation tool for implementation and evaluation of both APIT and IM_APIT localization algorithms performance is described in section 5.2. The implementation details of improved APIT is described in section 5.3. The simulation parameters and evaluation metrics of performance of both APIT and IM_APIT localization algorithms is described in section 5.4. Finally we summarize the chapter in section 5.5.

5.2 Development and Simulation Tools

The development environment, programming language and the simulation tools that were used for the implementation and evaluation of the proposed IM_APIT localization algorithm is described in this section.

According to [39] among several simulators and programming languages which are frequently used in the development of the localization algorithm in wireless sensor networks; NS-2, NS-3, OMNET++ and MATLAB are the popular ones. NS-2, NS-3 and OMNET++ are the most common simulators of WSNs used by localization algorithm developer, are capable of integrating WSN with other wireless communication but none of the current network simulators offers a complete localization algorithm library.

MATLAB (matrix laboratory) [41] has a very high performance varied feature rich computation and visualization tool comprising of thousands of features was built by Mathworks Inc. The most important feature of MATLAB is its easy platform for users to develop their own custom functions. MATLAB has various toolboxes like aerospace, control system design, Fuzzy Logic, Symbolic computations, statistics, communication and many others. The simplicity of the MATLAB's programming syntax and the availability of build-in complex mathematical expressions made the development and assessment of the localization algorithm effortless [39].

As it can be observed from the surveys [39], [40] MATLAB is more preferable even if it has no localization algorithm library like other WSNs simulators, because of its simplicity relative to others, and availability of built-in complex mathematical expressions and computational intelligence (CI). Therefore we select MATLAB simulator for implementing and evaluating our proposed IM_APIT localization algorithm.

5.3 Prototype Implementation

In previous section, the necessary simulation tool is identified for implementing and evaluating our proposed localization algorithm. This section describes the implementation detail of the IM_ APIT localization algorithm which is implemented in MATLAB R2013a software that can be purchased or downloaded from [41] and running on Intel(R) core (TM) i5-5200U CPU 2.20 GHz windows 8.1 professional.

Node deployment: to describe implementation detail of our localization algorithm, we begin with description of node deployment step in which 100 sensor nodes are randomly distributed in two dimensional space (2D) 100m by 100m square network area and anchor percentage and communication radius of sensor nodes are 25% and 125m tentatively. Figure 5.1 shows node distribution, in which red * indicates anchor nodes and blue O indicates unknown sensor nodes.

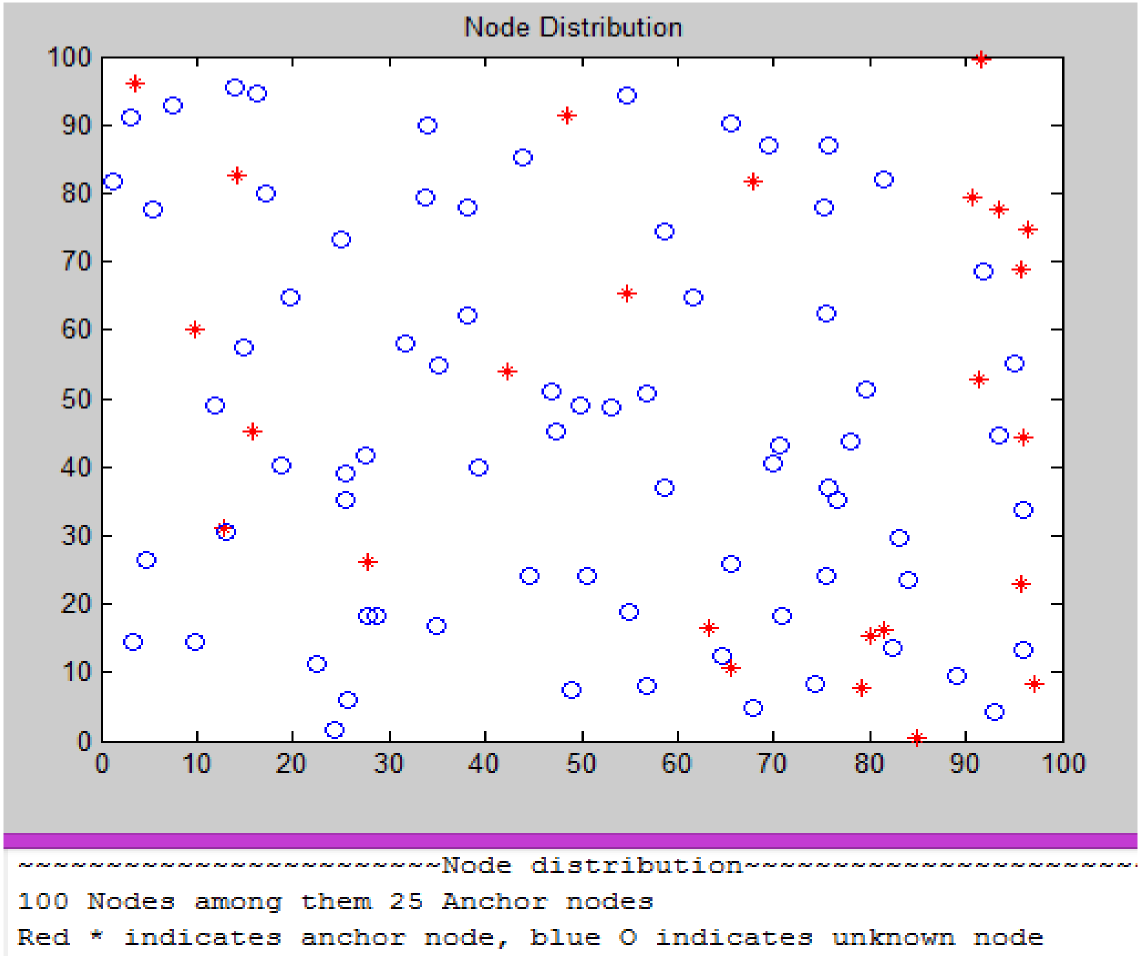


Figure 5.1: Node distribution example screen shoot

The initialization section: After node deployment have completed, the network is initialized and then the anchor nodes start broadcasting message that contain anchor identification number (ID), their x y coordinates and the received signal strength (RSS); the unknown nodes receive information coming from the anchors, and forward those received messages to the whole network in the format shown in table 5.1 and at the same time, store the received signal strength (RSS) values of the anchors as shown in figure 5.2

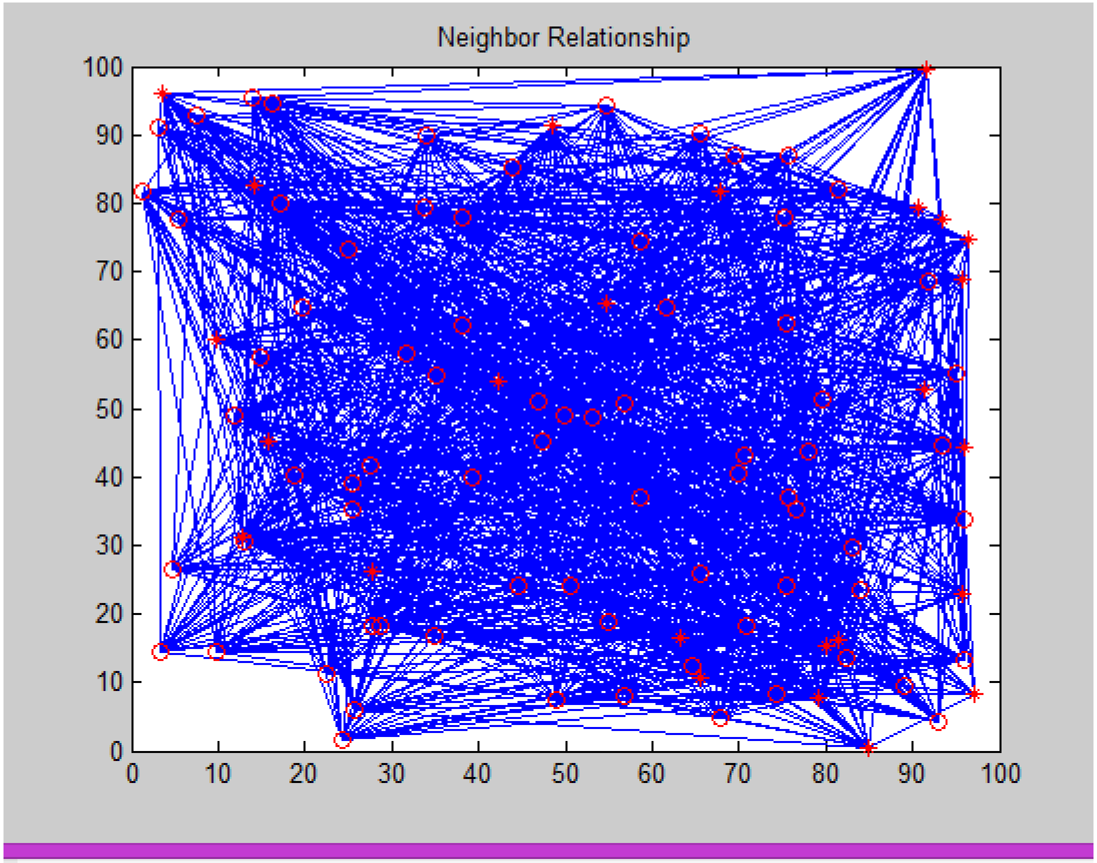
Table 5.1: Message format table of sensor nodes

Node ID	x , y coordinate	RSS value

```
Node ID: 26  x,y coordintate: 75.774    86.8695   RSSI value: -93.9528
Node ID: 27  x,y coordintate: 74.3132    8.4436   RSSI value: -73.4438
Node ID: 28  x,y coordintate: 39.2227    39.9783   RSSI value: -48.4322
Node ID: 29  x,y coordintate: 65.5478    25.987   RSSI value: -148.5555
Node ID: 30  x,y coordintate: 17.1187    80.0068   RSSI value: -79.0514
Node ID: 31  x,y coordintate: 70.6046    43.1414   RSSI value: -82.688
Node ID: 32  x,y coordintate: 3.1833    91.0648   RSSI value: -23.0974
Node ID: 33  x,y coordintate: 27.6923    18.1847   RSSI value: -62.0957
Node ID: 34  x,y coordintate: 4.6171    26.3803   RSSI value: -142.3595
Node ID: 35  x,y coordintate: 9.7132    14.5539   RSSI value: -63.3121
Node ID: 36  x,y coordintate: 82.3458    13.6069   RSSI value: -134.8071
Node ID: 37  x,y coordintate: 69.4829    86.9292   RSSI value: -96.8795
Node ID: 38  x,y coordintate: 31.7099    57.9705   RSSI value: -75.3464
Node ID: 39  x,y coordintate: 95.0222    54.986   RSSI value: -19.9755
Node ID: 40  x,y coordintate: 3.4446    14.4955   RSSI value: -54.8874
Node ID: 41  x,y coordintate: 43.8744    85.3031   RSSI value: -34.1205
Node ID: 42  x,y coordintate: 38.1558    62.2055   RSSI value: -140.9691
Node ID: 43  x,y coordintate: 76.5517    35.0952   RSSI value: -76.5388
Node ID: 44  x,y coordintate: 79.52    51.325   RSSI value: -16.0561
Node ID: 45  x,y coordintate: 18.6873    40.1808   RSSI value: -143.705
Node ID: 46  x,y coordintate: 48.9764    7.5967   RSSI value: -58.8388
```

Figure 5.2: Sample message broadcasting by sensor nodes screen shoot

A sensor node exchanges the table with all neighboring sensor nodes and builds a supper table using the information from all neighbors. Figure 5.2 shows sample message broadcast generated and figure 5.3 shows the neighbor relationship of unknown sensor nodes under the condition that the communication radius is 150m, the anchor percentage is 25%, and the total number of sensor nodes is 100 in which red * indicates anchor nodes and red O indicates unknown nodes.




```
~~~~~~~~~~~~~~~~~~~~~~~Neighbor Relationship~~~~~~~~~~~~~~~~~~~~~
100 Nodes,among them 25 Anchor nodes
Red * indicates anchor node, red O indicates unknown node
Communication radius:125m
Anchor node communication radius:125m
Communication model:Regular model
The average connectivity of the network is:24.98
The average number of neighbor anchor nodes in the network is:6.08
```

Figure 5.3: Sensor nodes neighbor relationship screen shoot

In order to implement APIT positioning algorithm, judging an unknown node whether locating within the triangle surrounded by three anchor nodes or not is the core phase of the algorithm. This is where we make improvement on original APIT localization algorithm. In our IM_APIT localization algorithm, the new method of PIT test was developed in the form of theorem, which is as follows:

"If the sum of distance from target node M to vertices of the triangle ABC (anchor nodes A, B and C) is greater than perimeter of the triangle ABC, then the target node M is outside triangle ABC. Otherwise M can be inside triangle ABC."

The last step of both original APIT and improved APIT localization algorithms is calculating localization error which is the core metrics to evaluate the performance of the algorithms. Figure 5.4 and Figure 5.5 show the localization error graph of node of the original APIT and improved APIT localization algorithms respectively. In these graphs red *indicates the anchor node, blue O indicates the estimated position of the unknown nodes, blue line segment (-) indicates the localization error of the unknown node and black O indicates an unknown node that cannot be located.

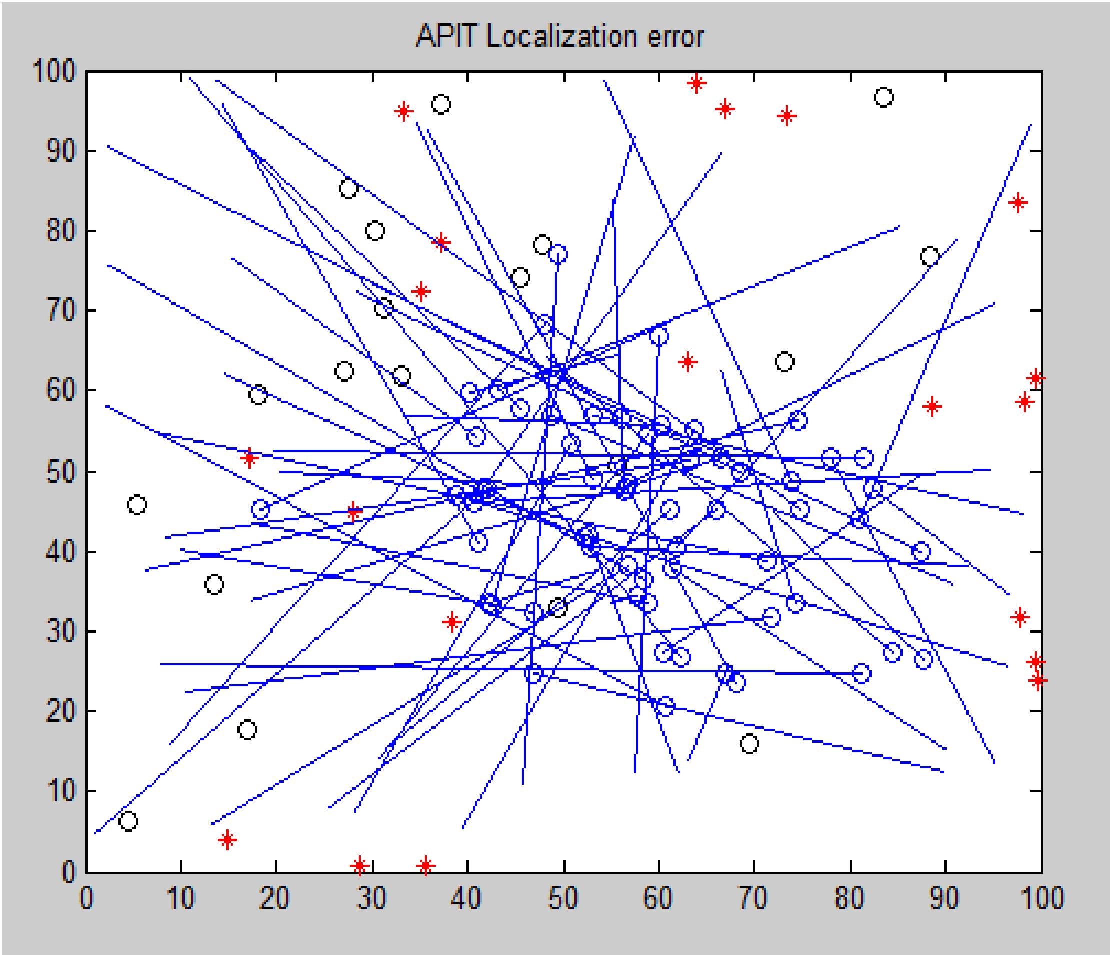


Figure 5.4: APIT localization Algorithm error graph of node

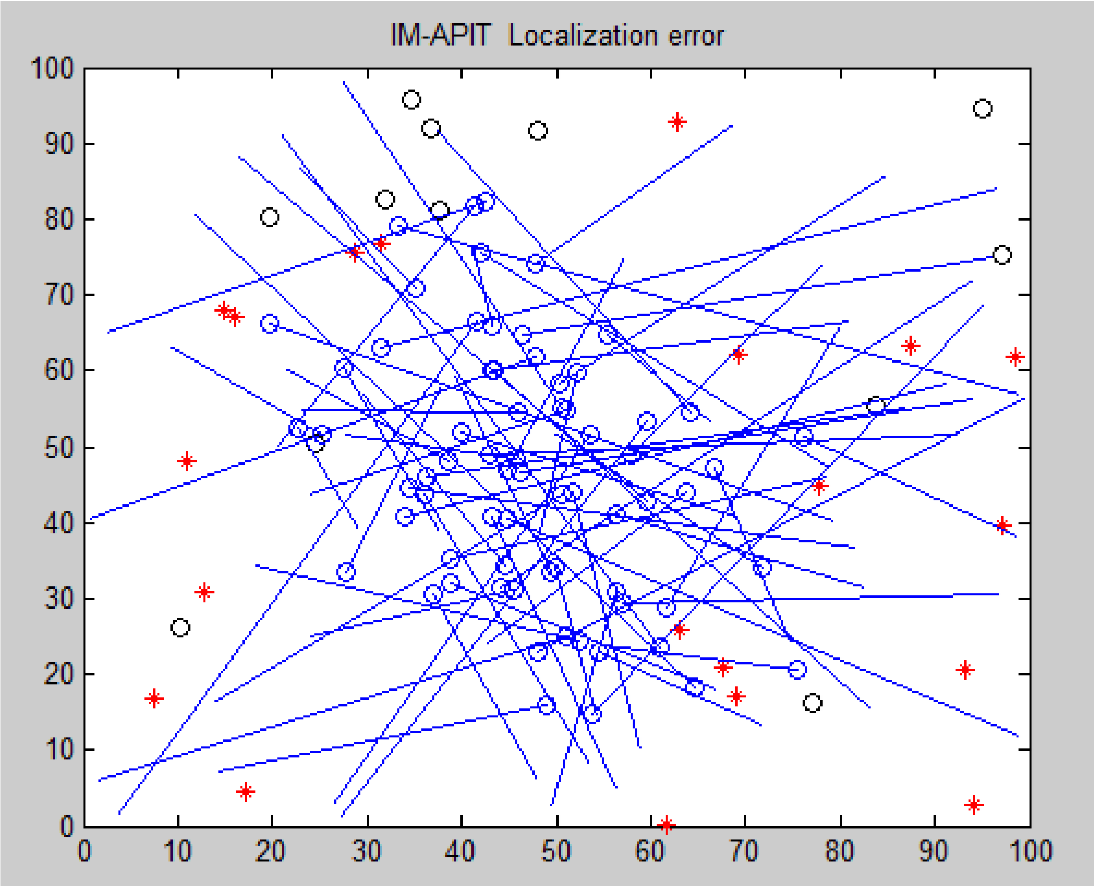


Figure 5.5: IM_APIT Localization Algorithm error graph of node

5.4 Simulation Experiments and Results

To test our improved APIT location algorithm and to investigate its performance, we conducted simulation experiment and evaluation using different parameters. To accomplish this, we followed the following procedure: first we set up the simulation environment in which node distribution scenario, topology of the WSN, channel model and other simulation parameters are defined in section 5.4.1. Lastly, we evaluated performance of improved APIT (IM_APIT) and original APIT localization algorithms based on localization accuracy and localization coverage in section 5.4.2.

5.4.1 Simulation Set up

In order to demonstrate the performance of the improved APIT localization algorithm we used the simulation software MATLAB R2013a to have comparison to original APIT algorithm and we used the following simulation parameters:

- Sensor nodes are randomly distributed in two dimensional 100m by 100m network square area.
- Total number of sensor nodes is 100
- Percentage of anchor nodes taken as 10%, 15%, 20%, 25%, 30%, 35% and 40% to study its influence on localization error rate.
- Communication radius of sensor node taken as 75m, 100m, 125m, 150m to study how it influence localization error rate.
- A general radio model is used to estimate meaningful distance from RSSI value, which is called regular model (free space radio model) in which its degree of irregularity (DOI) is 0. The parameter DOI is used to denote the irregularity of the radio pattern. It is defined as the maximum radio range variation per unit degree change in the direction of radio propagation. When the DOI is set to zero, there is no range variation, resulting in a perfectly circular radio model [36],[42] and this model is shown in formula (EQ. 5.1):

$$RSS = pt - pl(d0) - 10 \times \mathrm{n} \times \log_{10} \frac{d}{d0} \qquad \text{(EQ. 5.1)}$$

In the equation EQ.5.1, $Pl_{(d0)}$ (in dBm) represents the path loss at a reference distance d_0 (in meter), n is the path loss exponent. We define the following channel parameters value based on the suggestion in [21], [32], [42] as:

n = 2, d_0 = 1, $Pl(d_0)$ = 55;

Table 5.2: Summary of Simulation Parameters used for this Thesis

Parameters	**Value**
Network area	100m×100m squared area
Nodes distribution scenario	Randomly distributed
Communication radius (R)	75m, 100m ,125m and 150m
Total number of sensor nodes	100
Anchor percentage (AP)	10%,15%, 20%,25%, 30%, 35% and 40%
Radio model	Free space radio model(regular model)
Simulation Platform	MATLAB R2013a
Number of simulations	560

In our experiment, parameters that we feel they affect the localization error rate directly in the improved APIT (IM_APIT) algorithm are studied. These parameters include, Anchor percentage (AP) and communication radius of sensor nodes (R).

According to [5], [36], [14], minimizing the number of anchors is desirable from the equipment cost or deployment point of view. For example, using too many anchor nodes in the network that estimate their positions by global positioning system must be equipped with a GPS device, which is both power hungry and expensive; thus limiting the overall network lifetime. The anchor percentage (AP) is defined as the total number of anchor nodes divided by the total number of nodes in the network. This parameter is useful to calculate the trade-off between localization accuracy, the percentage of the nodes that can be localized against the deployment cost. For example, increasing the number of anchor nodes will lead to high accuracy as well as the percentage of the nodes that can be localized. On the other hand, the deployment cost will increase. A good localization algorithm must investigate the minimum number of anchor nodes that is needed for desired accuracy of the application

The Anchor percentage (AP) can be calculated by formula (EQ. 5.2) shown:

$$AP = \frac{Number\ of\ anchors}{Number\ of\ anchors + Number\ of\ unknown\ nodes} \times 100\% \qquad \text{(EQ. 5.2)}$$

5.4.2 Performance Evaluation Metrics and Results

Metrics is the standard of measurement, it varies with the measured environment [20]. In this section, we evaluated improved APIT (IM_APIT) localization algorithm in terms of localization accuracy and localization coverage as evaluation metrics comparing with the original APIT algorithm while varying Anchor percentage (AP) and communication radius of sensor nodes. According to [5], [14], [37] these evaluation metrics are discussed as follows:

Localization accuracy is defined as how well the position estimated by the localization algorithm approaches the actual position. A good localization algorithm should provide the approach to the actual position as closely as possible. However, positional accuracy is not the only ultimate goal of a good localization algorithm. This is largely application dependent. Different applications will have different requirements on the resolution of the positional accuracy. The impact of the required positional accuracy depends on the inter-node spacing. If the inter-node spacing is of the order of 100m, then the positional error of 1m can tolerable. However, if the inter-node spacing is of the order of 0.5m, then 1m error is highly unacceptable. The simplest way to calculate localization accuracy is to determine the residual error between estimated positions and actual positions for every sensor nodes in the network, sum them and average the results as shown in formula EQ. 5.3.

$$Localization\ accuracy = \frac{\sum_{i=1}^{n}\sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{n} \qquad \text{(EQ. 5.3)}$$

Where, $(x_{i,} y_i)$ are actual coordinates, (x_i', y_i') are estimated coordinates of unknown sensor nodes and n is the total number of sensor nodes in the network.

Average localization error is calculated in formula (EQ. 5.4), assuming the actual coordinates is $(x_{i,} y_i)$, the estimated coordinates of unknown sensor nodes is (x_i', y_i'), and n is the total number of sensor nodes in the network and R is the communication radius of the unknown sensor nodes.

$$Average\ Localization\ error = \frac{\sum_{i=1}^{n}\sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{nR} \qquad \text{(EQ. 5.4)}$$

Based on the evaluation metrics, the performance of our localization algorithm and the original APIT algorithm was simulated and compared by varying anchor percentage (AP) and communication radius (R) of sensor nodes.

Experiment One: **Localization error versus Anchor percentage(R)**. In this simulation experiment, different values of Anchor percentage (AP) in the interval of 10%, 15% , 20%, 25%, 30%, 35% and 40%, and fixed value of communication radius (R) was taken with other simulation parameters shown in table 5.2. For each seven of anchor percentage (AP) values ten simulations were performed, the corresponding localization error results were recorded, the average value was taken and tabulated as in table 5.3 to table 5.6 and then presented in a graph as shown in the figure 5.6 to figure 5.9 for later analysis.

Table 5.3: Localization error varying with Anchor Percentage (AP) for R = 75m

Anchor Percentage (AP)	APIT	IM_APIT
	Localization Error	**Localization Error**
10%	0.69	0.60
15%	0.61	0.56
20%	0.55	0.52
25%	0.52	0.50
30%	0.50	0.49
35%	0.48	0.46
40%	0.44	0.41

Table 5.4: Localization error varying with Anchor Percentage (AP) for R = 100m

Anchor Percentage (AP)	APIT	IM_APIT
	Localization Error	**Localization Error**
10%	0.46	0.44
15%	0.44	0.42
20%	0.43	0.41
25%	0.42	0.40
30%	0.41	0.39
35%	0.40	0.38
40%	0.39	0.36

Table 5.5: Localization error varying with Anchor Percentage (AP) for R = 125m

Anchor Percentage (AP)	APIT	IM_APIT
	Localization Error	**Localization Error**
10%	0.37	0..35
15%	0.34	0.32
20%	0.33	0.31
25%	0.31	0.29
30%	0.30	0.28
35%	0.28	0.27
40%	0.27	0.26

Table 5.6: Localization error varying with Anchor Percentage (AP) for R =150m

Anchor Percentage (AP)	APIT	IM_APIT
	Localization Error	**Localization Error**
10%	0.33	0.31
15%	0.32	0.30
20%	0.30	0.28
25%	0.29	0.27
30%	0.28	0.26
35%	0.27	0.25
40%	0.26	0.23

In Figure 5.6 as anchor percentage (AP) increases for communication radius (R) is 75m, it is observed that there is decrease of localization error on both APIT and IM-APIT localization algorithm. When AP increases from 10% to 40%, localization error decreases from 0.69 to 0.44 in original APIT and it decreases from 0.60 to 0.41 in IM_APIT localization algorithm.

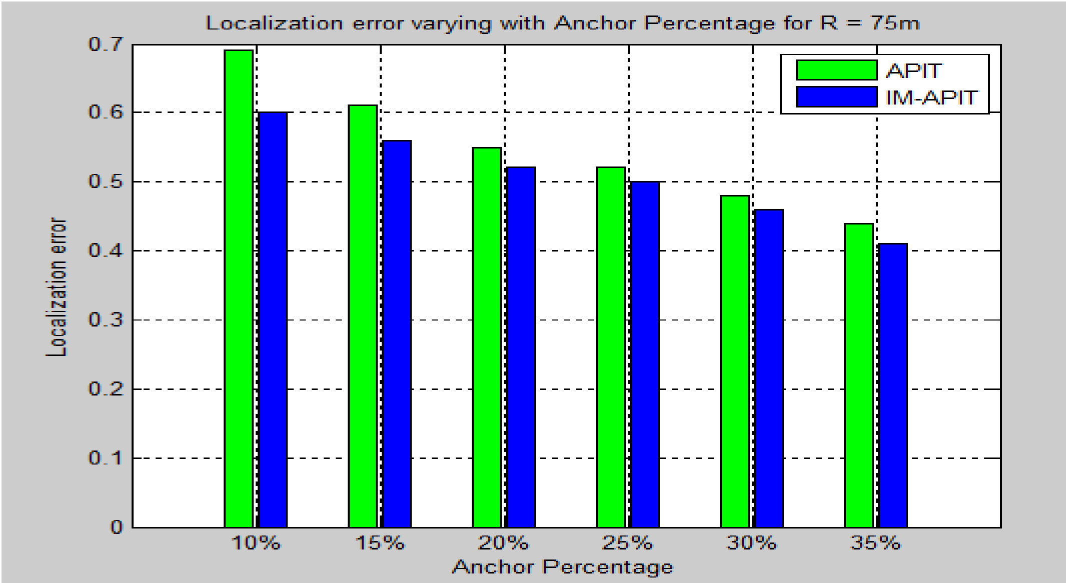


Figure 5.6: Localization error varying with Anchor percentage for R = 75m

In figure 5.7 for R = 100m, as AP increase, there is better change in performance on both APIT and IM_APIT localization algorithms in terms of localization accuracy relative to the result shown in table 5.3. Localization error of APIT decreases from 0.46 to 0.39, while localization error of IM_APIT decreases from 0.44 to 0.36. This shows IM_APIT localization algorithm performs better than APIT.

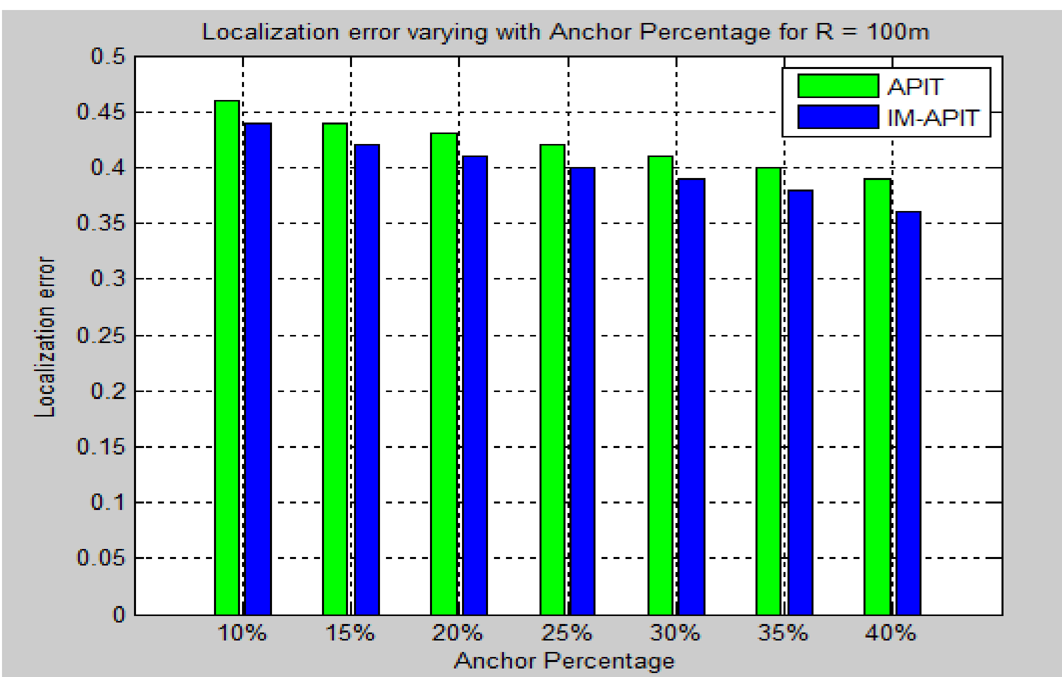


Figure 5.7: Localization error varying with Anchor percentage for R =100m

Figure 5.8 show that for communication radius of sensor nodes value is 125m, the overall trend of localization error in both IM_APIT and APIT localization algorithms decreases as the anchor percentage increases and it is clearly shown that IM_APIT localization algorithm performs better than APIT. The localization error of IM_APIT localization algorithm decreases from 0.37 to 0.27, while the localization error of APIT localization algorithm decreases from 0.35 to 0.26.

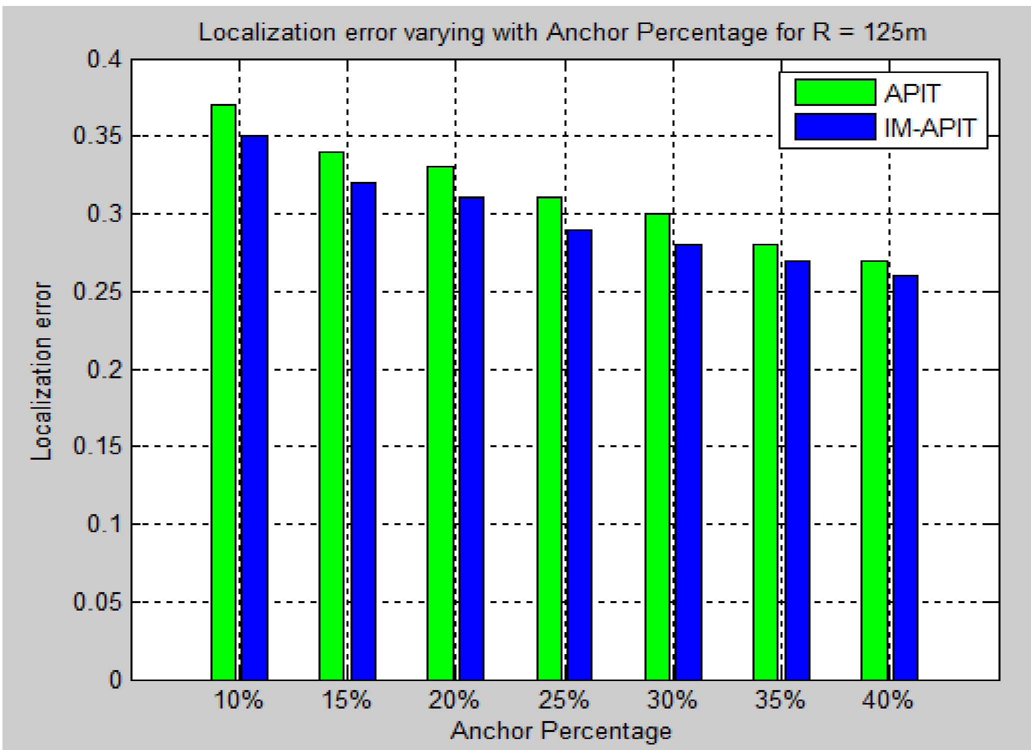


Figure 5.8: Localization error varying with Anchor percentage for R = 125m

In figure 5.9 for communication radius of sensor nodes value is 150m, the overall trend of localization error in both IM_APIT and APIT localization algorithms decreases as the anchor percentage increases and it show that the IM_APIT localization algorithm performs better than APIT localization algorithm in which the localization error of IM_APIT decreases from 0.31 to 0.23, while the localization error of APIT decreases from 0.33 to 0.26.

From this experiment, it can be concluded that as anchor percentage increases on both IM_APIT and APIT localization algorithms, localization error decreases. However, increasing anchor percentage has its own tradeoffs in terms of energy consumption and additional hardware requirement. Therefore taking into consideration the tradeoffs, we conclude from this experiment that taking anchor percentage value of 25%, we can have tolerable localization error for values of communication radius is 150m.

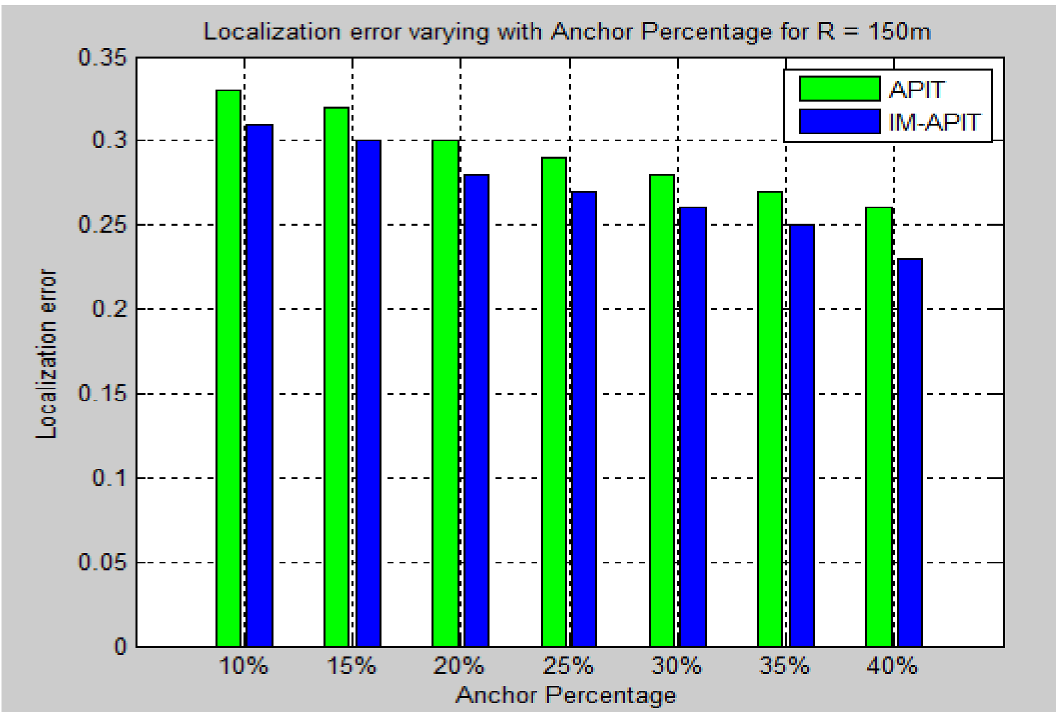


Figure 5.9: Localization error varying with Anchor percentage for R = 150m

Experiment Two: Localization error versus Communication radius of sensor nodes (R).

In this simulation experiment, different values of communication radius of sensor nodes (R) in the interval of 75m,100m,125m and 150m were taken with fixed value of anchor percentage (AP) in addition to the remaining simulation parameters given in the table 5.2 and simulated ten times for each value of communication radius (R) , recorded and then average value of ten trials was taken and tabulated as shown in the table 5.7 to table 5.13 and then illustrated in a graph as shown in the figure 5.10 to figure 5.13 for later analysis.

Table 5.7: Localization error varying with Communication radius (R) for AP = 10%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.69	0.60
100m	0.46	0.44
125m	0.37	0.35
150m	0.30	0.28

Table 5.8: Localization error varying with Communication radius (R) for AP = 15%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.61	0.56
100m	0.44	0.42
125m	0.34	0.32
150m	0.28	0.26

Table 5.9: Localization error varying with Communication radius (R) for AP = 20%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.55	0.52
100m	0.43	0.41
125m	0.33	0.31
150m	0.29	0.27

Table 5.10: Localization error varying with Communication radius (R) for AP = 25%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.52	0.50
100m	0.42	0.40
125m	0.33	0.29
150m	0.31	0.26

Table 5.11: Localization error varying with Communication radius (R) for AP = 30%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.50	0.49
100m	0.41	0.39
125m	0.30	0.28
150m	0.28	0.26

Table 5.12: Localization error varying with Communication radius (R) for AP = 35%

Communication radius (R)	Localization Error of APIT	Localization Error of IM_ APIT
75m	0.48	0.46
100m	0.40	0.38
125m	0.28	0.27
150m	0.27	0.26

Table 5.13: Localization error varying with Communication radius (R) for AP = 40%

Communication radius (R)	**Localization Error of APIT**	**Localization Error of IM_ APIT**
75m	0.44	0.41
100m	0.39	0.31
125m	0.27	0.23
150m	0.26	0.21

In this experiment two, as we can observe in Figures 5.10 to 5.13 the overall trend of localization error in APIT localization algorithm and IM_APIT localization algorithm decreases as communication radius of sensor nodes increases. Especially as we have discussed under analysis of experiment one, better performance of both APIT and IM_APIT localization algorithms was observed for AP from 25% to 40% at the value of communication radius 150m and it is clearly shown in figures 5.10 to 5.13 that IM_APIT localization algorithm performs better than APIT localization algorithm.

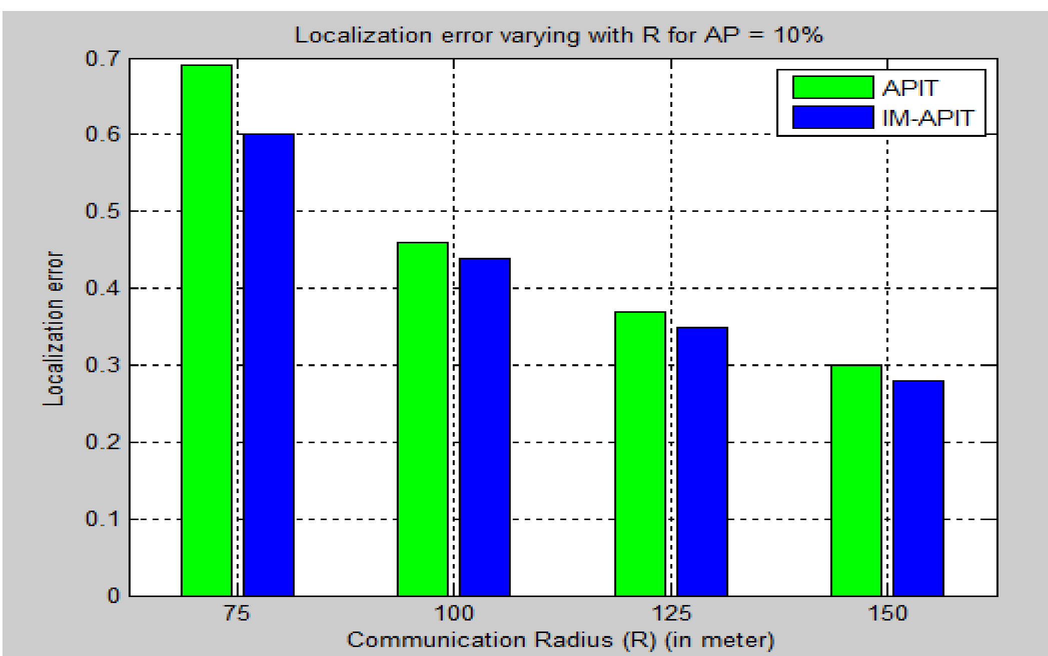


Figure 5.10: Localization error varying with communication radius (R) for AP = 10%

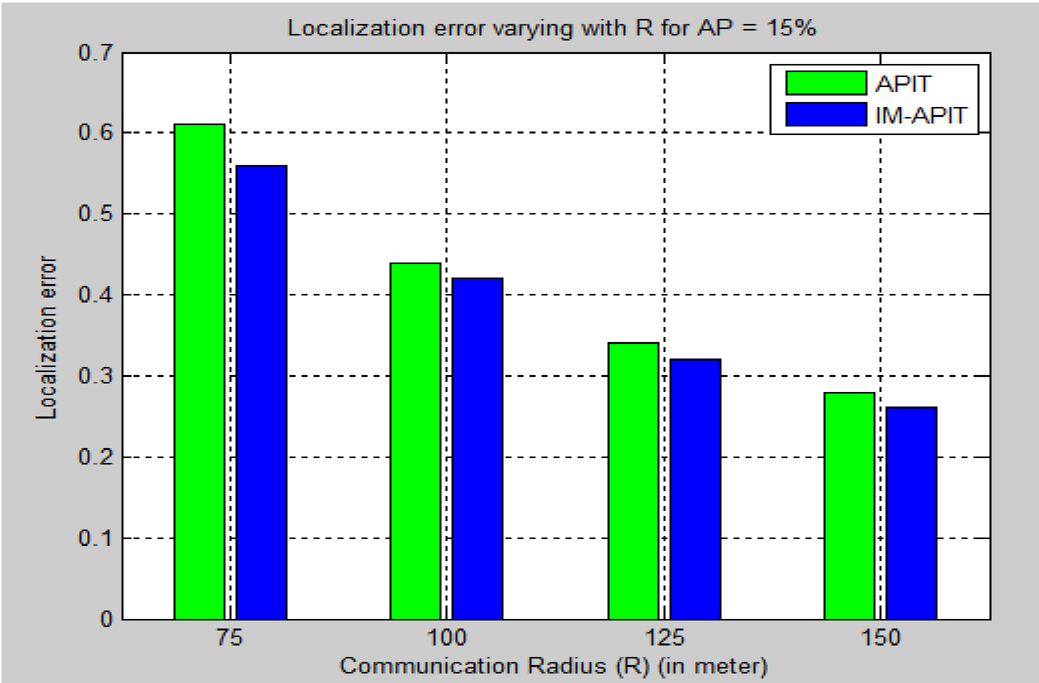


Figure 5.11: Localization error varying with communication radius (R) for AP = 15%

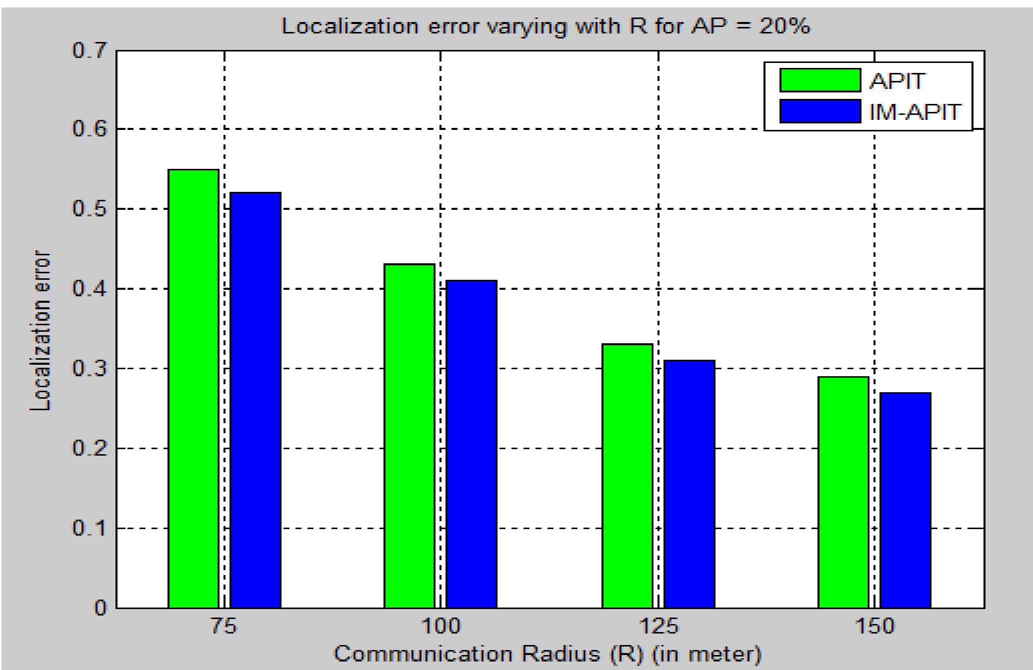


Figure 5.12: Localization error varying with communication radius (R) for AP = 20%

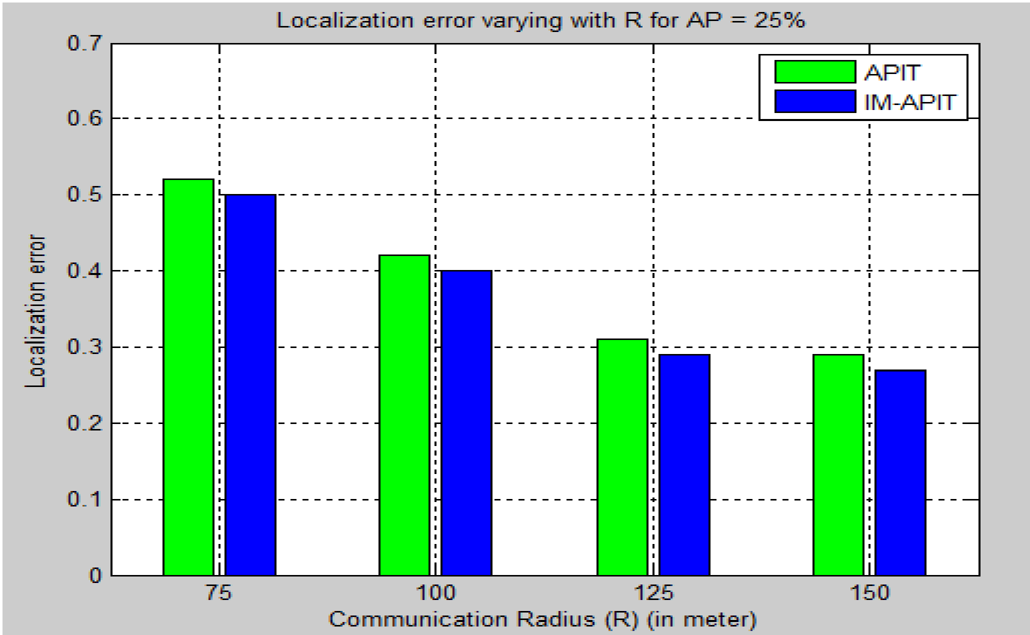


Figure 5.13: Localization error varying with communication radius (R) for AP = 25%

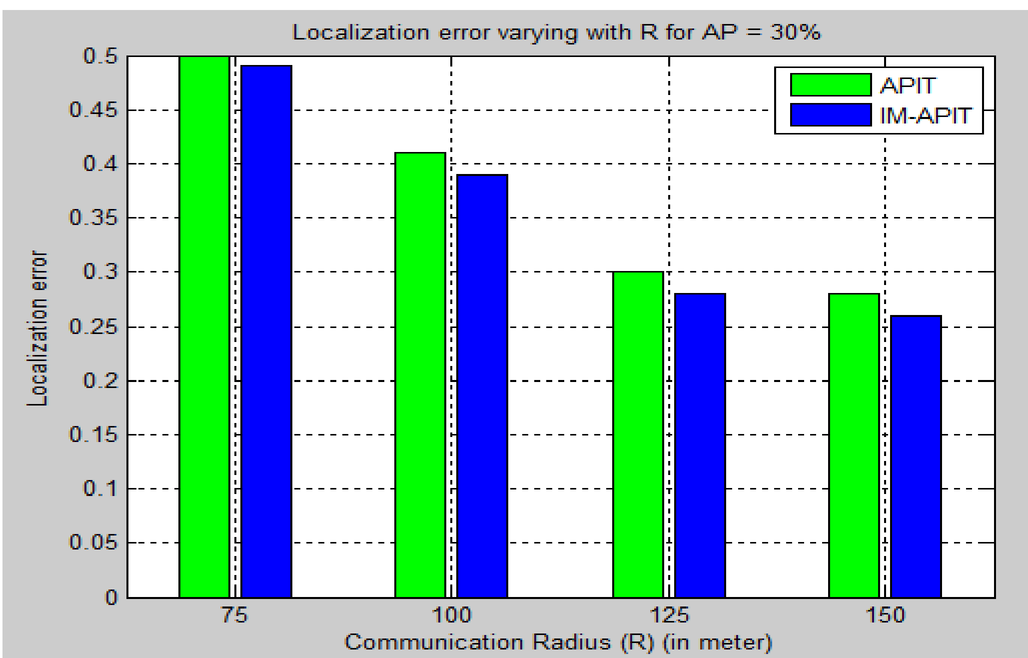


Figure 5.14: Localization error varying with communication radius (R) for AP = 30%

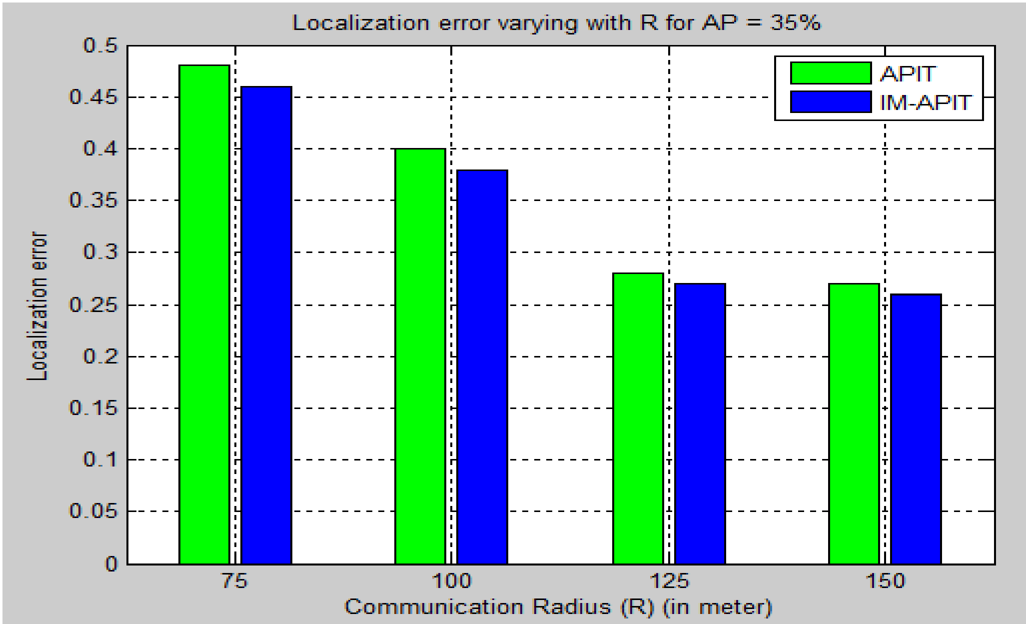


Figure 5.15: Localization error varying with communication radius (R) for AP = 35%

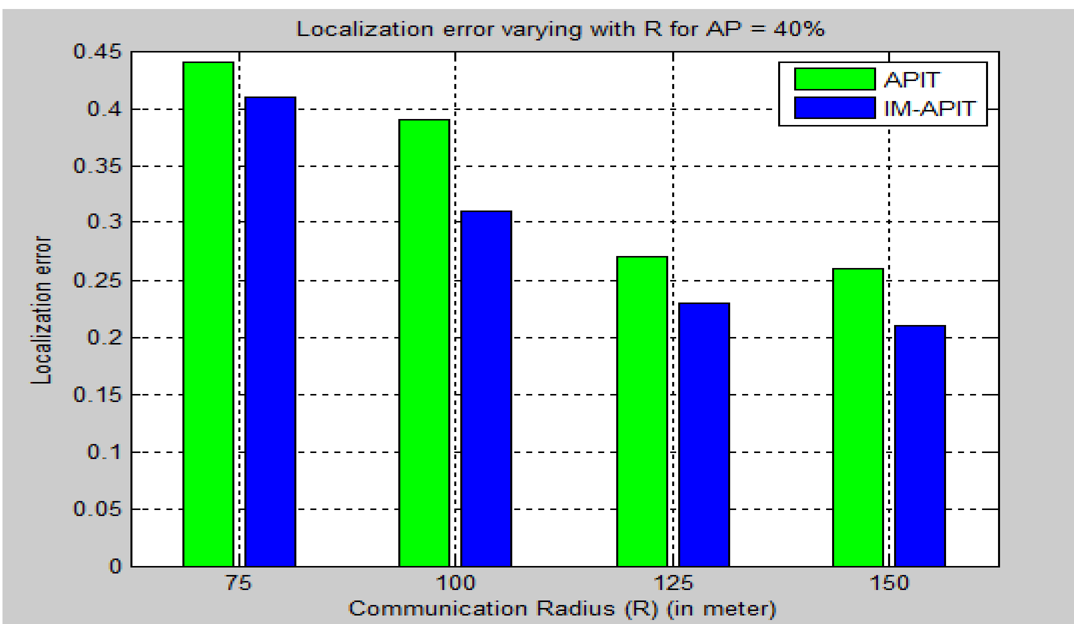


Figure 5.16: Localization error varying with communication radius (R) for AP = 40%

Experiment Three: Anchor Percentage (AP) versus Energy Consumption versus

In this simulation experiment to compare the energy consumption of both APIT and IM-APIT localization algorithms, the runtime of both algorithms is recorded by anchor percentage (AP) in the interval of 10%, 15%, 20%, 25%, 30%, 35% and 40% and different values of runtime for different anchor percentage is observed as shown in table 5.14

Table 5.14: APIT and IM_APIT run time varying anchor percentage

Anchor percentage (AP)	**APIT run time (in sec)**	**IM_APIT run time (in sec)**
10%	0.7	0.7
15%	0.8	0.9
20%	2	2.2
25%	4.9	5.2
30%	12	14
35%	30	35
40%	47	51

To associate the recorded different run time value of corresponding anchor percentage of both APIT and IM_APIT localization algorithms with the energy consumption of both algorithms, the following values are considered [46] as shown below.

CPU active mode current consumption	8mA (0.008A)
CPU LPM (low power mode) current consumption	0.0005mA (0.000005A)
current consumption for packet receiving (Rx)	18.8mA (0.0188A)
current consumption for packet transmitting (Tx)	17.4mA (0.0174A)
idle mode	0.426mA (0.000426A)
Voltage	3V

Since both APIT and IM_APIT localization algorithms send and receive the same amount of data we will not expect difference energy consumption in Rx, Tx and idle radio. The only difference expected energy consumption is on CPU active mode.

Power (P) in terms of voltage (V) and current (I) is given by formula:

$$P = V \times I$$

From table 5.15, V = 3V, I = 0.008A

Power (P) of CPU in active mode will be,

$$P = 3V \times 0.008A = 0.024\text{Watt (J/sec)}$$

Power in terms of energy and time also formulated as; Power = Energy / time.

From this again energy can be formulated as:

$$\text{Energy} = \text{Power} \times \text{time} \qquad \text{(EQ. 5.5)}$$

Since power is 0.024 J/sec, using formula given in EQ.5.5, the energy consumption of both APIT and IM_APIT localization algorithms is calculated for the corresponding anchor percentage taking the corresponding run time value from table 5.14 and tabulated in the table 5.15.

Table 5.15: Anchor Percentage versus Energy consumption

Anchor Percentage(AP)	**APIT Energy Consumption (Joule)**	**IM_APIT Energy consumption (Joule)**
10%	0.0168	0.0168
15%	0.0192	0.0216
20%	0.048	0.0552
25%	0.1176	0.1248
30%	0.288	0.336
35%	0.72	0.84
40%	1.128	1.224

In this simulation experiment as it can observed from figure 5.17 that as anchor percentage increases, the energy consumption of both APIT and IM_APIT localization algorithm increases. As it can be observed, the energy consumption of IM_APIT algorithm is slightly higher than energy consumption of APIT algorithm in insignificant value. For instance when anchor percentage of both APIT and IM_APIT algorithms is 25%, energy consumption of IM_APIT is 0.1248 Joule, while energy consumption of APIT localization algorithm is 0.1176 Joule.

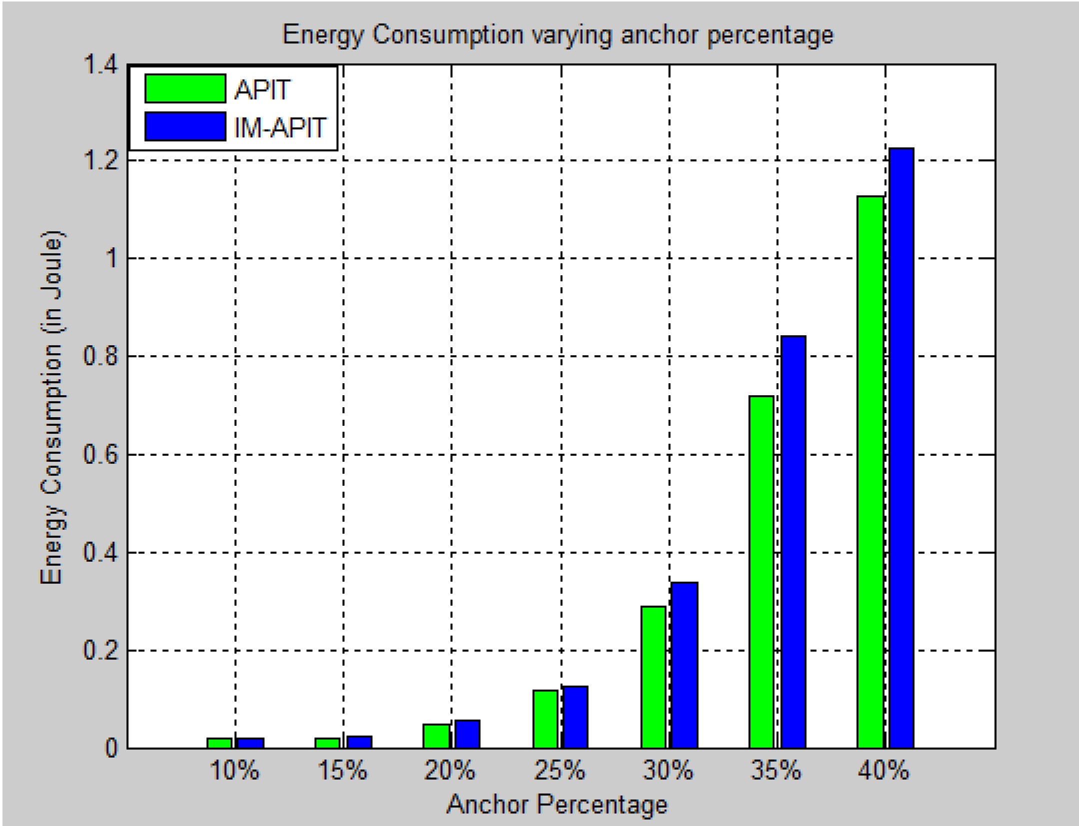


Figure 5.17: Energy consumption versus Anchor percentage

5.5 Summary

In this chapter, we described the implementation details, simulation experiment and evaluation of performance of our improved APIT localization algorithm. To evaluate the performance of our IM_APIT localization algorithm, we performed simulation experiment using MATLAB R2013a simulator. Localization error was used as a main performance evaluation metrics for both our improved APIT and APIT localization algorithms and their performance was compared using the same parameters. The simulation experiment result shows our IM_APIT localization algorithm performs better than APIT localization algorithm without additional network cost.

Chapter Six: Conclusion and Future work

6.1 Conclusion

Wireless sensor network is a new class of wireless network that is becoming very popular with a huge number of civilian and military applications. WSNs have gained worldwide research and industrial interest particularly with the rapid increase in wireless communication technologies and micro-electromechanical systems (MEMS) technology which has facilitated the development of smart sensors. In applications such as target tracking, geographic routing, environmental monitoring and forest fire monitoring, the location of information generating sensors is very important.

Location of sensor nodes can be obtained either by placing the sensor nodes at points with known coordinates manually or by deployment of global positioning systems (GPS) on every sensor node. Manual configuration is impractical in large-scale deployment and adding GPS to all nodes in the network is infeasible because of high energy consumption, cost and line-of-sight nature of signal of GPS satellites. Because of the problems related to GPS, it was necessitated research on localization algorithms.

Based on the location determination mechanism localization algorithms are categorized in to range-based and range-free localization algorithms. Range-based localization algorithms utilize the measurement such as angle of arrival (AOA), time of arrival (TOA), time difference of arrival (TDOA) and received signal strength indicator (RSS) and they have higher location accuracy than the range-free localization schemes, but are hardware intensive and high cost of sensor nodes. As far as the tradeoff between location precision and deployment cost is concerned, range-free localization schemes are more suitable for WSNs than expensive range-based ones. Among range-free localization algorithms, DV-HOP, Centroid and APIT are the well-known range free localization algorithms. However, these localization algorithms have their own shortcomings and they got interest of researchers in the area. We focused our research on the APIT localization algorithm in this thesis.

In this thesis, improved APIT (IM_APIT) localization algorithm is proposed. In APIT localization algorithm among its four basic phases, Point-In-Triangle (PIT) test is the core one for the implementation of the algorithm. Modification of PIT test is proposed to minimize the probability of wrong judgement of whether the unknown sensor node is inside triangle formed by three anchor nodes or outside by stating and proving one theorem from the concept of plane geometry as a PIT test so that to improve the performance of the algorithm.

IM_APIT localization algorithm and the original APIT are implemented using MATLAB R2013a software and the simulation results are presented, analyzed and evaluated based on the localization error rate, and the main contributions of this thesis are listed in the following:

- New and simple side based PIT test method is developed
- As the result of new PIT test, localization error of IM_APIT localization algorithm decreases in a better way than APIT.
- When AP = 20%, localization error of our IM_APIT localization algorithm is 0.28, while localization error of APIT localization algorithm is 0.32.
- When AP = 25%, localization error of our IM_APIT localization algorithm is 0.26, while localization error of APIT localization algorithm is 0.31.
- When AP = 30%, localization error of our IM_APIT localization algorithm is 0.23, while localization error of APIT localization algorithm is 0.29.

In general, based on localization accuracy IM_APIT localization algorithm shows better performance than the original APIT localization algorithm.

6.2 Future Works

Although we tried our best to realize localization accuracy of unknown sensor nodes using APIT localization algorithm in wireless sensor network by developing a side based new PIT test method; for the sum of distances from a target sensor node to the three vertices of a triangle formed by the three anchor nodes is greater than semi-perimeter and less than perimeter of the triangle, the target sensor node has possibility to be inside and outside of the triangle. Additional investigation to clearly identify the position of the node and minimizing energy consumption are left as future work.

REFERENCES:

[1] F. Mekelleche, H. Haffaf (2017), Classification and Comparison of Range-based Localization Techniques in Wireless Sensor Networks, Journal of Communications Vol.12, No. 4

[2] P. Naik1, N. Telkar and, K. Kotin (2016), Survey on Wireless Sensor Network with their remaining challenges, IJSRST, Volume 2 , Issue 6 , India

[3] J.Bangash, A. H.Abdullah, A.W. Khan (2015), Issues and Challenges in Localization of Wireless Sensor Networks, Faculty of Computing, Universiti Teknologi Malaysia

[4] M. S. Obaidat, S. Misra (2014), Principles of Wireless Sensor Networks, Cambridge University pp. 14-25

[5] A.K. Paul, T. Sato (2017), Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges, Journal of Sensor and Actuator Networks, Department of Electronics and Communications Engineering, East West University, Dhaka 1212, Bangladesh

[6] M.R.Ahmed , X. Huang , D.Sharma ,et al (2012) Wireless sensor network characteristics and architectures, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:6, No:12

[7] E. Zanaj, E. Elbasani, M. Alinci, et al (2014), Application of the localization algorithms in WSN, Polytechnic University of Tirana, Albania

[8] Tashnim J.S. Chowdhury, C. Elkin, et al. (2016), Advances on Localization Techniques for Wireless Sensor Networks, Computer Networks, IEEE, USA

[9] A. Kumar, N. Chand, V. Kumar, (2011), Range Free Localization Schemes for Wireless Sensor Networks, International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.6

[10] G.Mao, B.Fian, (2009) Localization Algorithms and Strategies for Wireless Sensor Networks, Information Science Reference, New York.

[11] Dr. Yan Zhang, (2010) RFID and Sensor networks Architectures, Protocols, Security and Integrations, CRC Press, (276-292)

[12] S. Li, X. Ding, T. Yang, (2015), Analysis of Five Typical Localization Algorithms for Wireless Sensor Networks, Wireless Sensor Network, China (27-33**)**

[13] S. P. Singha, S. C. Sharma, (2015), Range Free Localization Techniques in Wireless Sensor, 3rd International Conference on Recent Trends in Computing India

[14] D. Zhang, Z. Fang, H. Sun et al. (2017), A Range-Free Location Algorithm Based on Homothetic Triangle Cyclic Refinement in Wireless Sensor Networks, China

[15] Habib M. Ammari (2014), The Art of Wireless Sensor Networks, Volume 1: Fundamentals, Springer, 353-384

[16] F. Zeng, M. Yu, Ch. Zou, et al. (2012), An Improved Point-In-Triangulation Localization Algorithm Based on Cosine Theorem, China

[17] J. Wang , H. Jin (2009), Improvement on APIT Localization Algorithms for Wireless Sensor Networks, 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, IEEE, China

[18] Ch.Liu, S. Liu1, W. Zhang et al.(2016), The Performance Evaluation of Hybrid Localization Algorithm in Wireless Sensor Networks, Mobile Netw Appl, springer

[19] S. M. Hosseinirad , M. Niazi2, J. Pourdeilami, et al. (2014), On improving APIT algorithm for better localization in WSN, Journal of AI and Data Mining Vol. 2, No. 2, 2014, 97-104.

[20] H. Mahmoud A. Fahmy (2016), Wireless Sensor Networks Concepts, Applications, Experimentation and Analysis, Springer, Egypt, (55-120)

[21] T. VAL and A.WEI (2013), Improvement of Range-free Localization Systems in Wireless Sensor Networks, Computers and Embedded Systems, National Institute of Applied Sciences of Toulouse, France

[22] Y. K. Fulara (2015), Some aspects of wireless sensor networks, International Journal on AdHoc Networking Systems (IJANS) Vol. 5, No. 1, India

[23] M. Kocakulak, I. Butun (2017), An Overview of Wireless Sensor Networks Towards Internet of Things, IEEE, USA

[24] P.Tiwari1, V.Prakash, R.Gaurav , et al.(2015), Wireless Sensor Networks: Introduction, Advantages, Applications and Research Challenges, HCTL Open International Journal of Technology Innovations and Research (IJTIR), Volume 14, India

[25] M.Bhende, S.J. Wagh, A .Utpat (2014), A Quick Survey On Wireless Sensor Networks, Fourth International Conference on Communication Systems and Network Technologies, India

[26] D. K.Gupta (2013), A review on wireless sensor networks, International Conference on Recent Trends in Applied Sciences with Engineering Applications, Vol.3, No.1, 2013

[27] D. Prashar, K.Jyoti, D.Kumar (2016), A Comparison of Distributed Range Free Localization Algorithms in Wireless Sensor Networks, Indian Journal of Science and Technology, Vol 9(28), India

[28] X. Zhang (2016) Localization in Wireless Sensor Networks, Arizona State University, Ph.D. dissertation

[29] A. Pal (2010), Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges, Network Protocols and Algorithms, 2010, Vol. 2, No. 1

[30] Ch. J. Zinsmeyer ,T. Korkmaz (2012), A Comparative Review of Connectivity-Based Wireless Sensor Localization Techniques, Journal of Internet Services and Information Security (JISIS), volume: 2, number: 1/2, pp. 59-72, USA

[31] S. Kumar and R. M. Hegde (2017), A Review of Localization and Tracking Algorithms in Wireless Sensor Networks, Indian Institute of Technology Patna

[32] W. Tie-zhou, Z. Yi-shi, Z. Hui-jun, et al. (2013), Wireless Sensor Network Node Location Based on Improved APIT, Journal of Surveying and Mapping Engineering Jun. 2013, Vol. 1 Issue 1, PP. 15-19

[33] M. Farrag, M. Abo-Zahhad, M.M. Doss et al. (2016), Different Aspects of Localization Problem for Wireless Sensor Networks-Review, International Journal of Computer Networks and Communications Security, VOL. 4, NO. 5, MAY 2016, 130-140

[34] E. Qasem, Sh.Tarek, R. Sheltami*a*, et al. (2017), A Comparative Study of Range-Free and Range-Based Localization Protocols for Wireless Sensor Network, International Journal of Distributed Systems and Technologies, KSA

[35] R. Sharma S. Malhotra, (2015), Approximate Point in Triangulation (APIT) based Localization Algorithm in Wireless Sensor Network, International Journal for Innovative Research in Science & Technology, Volume 2, Issue 03

[36] T.He , C.Huang , B. Blum et al , (2003), Range-Free Localization Schemes for Large Scale Wireless sensor networks, MobiCom , San Diego, California, USA

[37] FeiLiu, G. Feng (2014), Research on RSSI and Triangles Deformation Based APIT Localization Algorithm for Wireless Sensor Networks, Advanced Materials Research Vols. 998-999, pp 1305-1310, China

[38] K. Chen, J.Xu, Y.Fu et al. (2017), Performance Evaluation of an Improved APIT Localization Algorithm for Underwater Acoustic Sensor Networks, Journal of Computers Vol. 29 No. 1, 2018, pp. 132-142, China

[39] Noorfarha binti Mohd Ngabas, Jiwa Bin Abdullah (2014) A Review of Simulation Framework for Wireless Sensor Networks Localization, IEEE, Malaysia

[40] A. Nayyar, R.Singh (2015),A Comprehensive Review of Simulation Tools for Wireless Sensor Networks (WSNs), Journal of Wireless Networking and Communications 2015, 5(1): 19-47,India

[41] Matlab. http://www.mathworks.com/.

[42] F.Xiu-fang, Q.Hui-bo (2009),Improvement and Simulation for a Localization Based on APIT, 2009 International Conference on Computer Technology and Development, China

[43] Q. Miao, B. Huang and B. Jia (2018), Estimating Distances via Received Signal Strength and Connectivity in Wireless Sensor Networks, Inner Mongolia University, Hohhot, China

[44] G. Han H. Xu · J. Jiang (2011),Localization algorithms of Wireless Sensor Networks, Department of Information & Communication Systems, Hohai University, Changzhou, China

[45] Ademuwagun, A. and Fabio, V. (2017) Reach Centroid Localization Algorithm. Wireless Sensor Network, 9, 87-101

[46] http://www. http://zolertia.sourceforge.net/wiki/index.php/Z1, viewed on April 15, 2019

APPENDIX A: MATLAB Source Code for IM_APIT Localization Algorithm

```
   function im_apit(all_nodes)
   % parameter setting
clear;
clc;
close all;
all_nodes.square_L = 100;
all_nodes.nodes_n = 100;
all_nodes.anchors_n = 25;
all_nodes.true = [100*rand(1,100)' 100*rand(1,100)'];
all_nodes.estimated =[100*rand(1,25)' 100*rand(1,25)'];
all_nodes.anc_flag = [ones(25,1);zeros(75,1)];
comm_r = 150;
anchor_comm_r =1;
model = 'Regular model';
neighbor_matrix = +(rand(100,100)<0.25);
neighbor_rss = -161.3221+(161.5)*rand(100,100);

% ~~~~~~~~~~~~Nodes deployment~~~~~~~~~~~~~~

figure;
hold on;
box on;
plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');
plot(all_nodes.true(all_nodes.anchors_n+1:all_nodes.nodes_n,1),all_nodes.true(all_nodes.anchor
s_n+1:all_nodes.nodes_n,2),'bo');
axis([0,all_nodes.square_L,0,all_nodes.square_L]);
title('Node Distribution ');%(IM-APIT )');
try %Draw a regular distribution of grid lines

   x=0:all_nodes.grid_L:all_nodes.square_L;
   set(gca,'XTick',x);
   set(gca,'XTickLabel',num2cell(x));
   set(gca,'YTick',x);
   set(gca,'YTickLabel',num2cell(x));
   grid on;
catch
```

```
    %none
end
try %Draw the boundaries of the C area
    square_L=all_nodes.square_L;
    area=all_nodes.area;
    plot([area(2) area(2)],[area(3) area(4)],'-k',[area(2) square_L],[area(3) area(3)],'-k',[area(2)
square_L],[area(4) area(4)],'-k');
catch
   %none
end
disp('~~~~~~~~~~~~~~~~~~~~~~~~~Node distribution~~~~~~~~~~~~~~~~~~~~~~~~~~~');
disp([num2str(all_nodes.nodes_n),' Nodes',' among them ',num2str(all_nodes.anchors_n),'
Anchor nodes']);
disp('Red * indicates anchor node, blue O indicates unknown node');


 %`~~~~~~~~~~ Topology~~~~~~~~~~~~~~~~~~

figure;
hold on;
 box on;
for i=1:all_nodes.nodes_n
    for j=i+1:all_nodes.nodes_n
        if neighbor_matrix(i,j)==1
            plot(all_nodes.true([i,j],1),all_nodes.true([i,j],2),'-b');
        end
    end
end
plot(all_nodes.true(all_nodes.anchors_n+1:all_nodes.nodes_n,1),all_nodes.true(all_nodes.anchor
s_n+1:all_nodes.nodes_n,2),'ro');
plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');
axis([0,all_nodes.square_L,0,all_nodes.square_L]);
title('Neighbor Relationship');% (IM-APIT)');
disp('~~~~~~~~~~~~~~~~~~~~~~~~~Neighbor
Relationship~~~~~~~~~~~~~~~~~~~~~~~~~~~');
 disp([num2str(all_nodes.nodes_n),' Nodes,','among them ',num2str(all_nodes.anchors_n),'
Anchor nodes']);
 disp('Red * indicates anchor node, red O indicates unknown node');
disp(['Communication radius:',num2str(comm_r),'m']);
disp(['Anchor node communication radius:',num2str(comm_r*anchor_comm_r),'m']);
disp(['Communication model:',model]);
```

```
try
   disp(['DOI=',num2str(DOI)]);
catch
   %none
end
if anchor_comm_r==1
   disp(['The average connectivity of the network
is:',num2str(sum(sum(neighbor_matrix))/all_nodes.nodes_n)]);
   disp(['The average number of neighbor anchor nodes in the network
is:',num2str(sum(sum(neighbor_matrix(1:all_nodes.nodes_n,1:all_nodes.anchors_n)))/all_nodes.
nodes_n)]);
else
   disp(['The average number of anchor nodes that an unknown node can listen to
is:',num2str(sum(sum(neighbor_matrix(all_nodes.anchors_n+1:all_nodes.nodes_n,1:all_nodes.an
chors_n)))/(all_nodes.nodes_n-all_nodes.anchors_n))]);
   disp(['The average number of unknown nodes in the unknown nodes communication area
is:',num2str(sum(sum(neighbor_matrix(all_nodes.anchors_n+1:all_nodes.nodes_n,all_nodes.anc
hors_n+1:all_nodes.nodes_n)))/(all_nodes.nodes_n-all_nodes.anchors_n))]);
end

   % ~~~~~IM_APIT ~~~~~~~~~~~~~

%
   unknown_node_index=all_nodes.anchors_n+1:all_nodes.nodes_n;
   length(unknown_node_index);
   row_n=ceil(all_nodes.square_L);    col_n=row_n;
   centroid_x=repmat(((1:col_n)-0.5),row_n,1);
   centroid_y=repmat(transpose(((1:row_n)-0.5)),1,col_n);
   for i=unknown_node_index
     disp(['Node ID: ',num2str(i),'  x,y coordintate: ',num2str(all_nodes.true(i,1)),'    '
num2str(all_nodes.true(i,2)),'   RSSI value: ',num2str(neighbor_rss(i))]); % display node ID ,x-y
coordinates and neighbor rss value
       neighboring_anchor_index=find(neighbor_matrix(i,1:all_nodes.anchors_n)==1);
       neighboring_anchor_n=length(neighboring_anchor_index);
       if neighboring_anchor_n>=3
           gridmap=zeros(row_n,col_n);
           grid_covered_flag=zeros(row_n,col_n);
%         else
%              disp('since neighboring_anchor_n<3 ,APIT does not work.');
%         end
           for a=1:neighboring_anchor_n-2
```

```
            for b=a+1:neighboring_anchor_n-1
                for c=b+1:neighboring_anchor_n
%  Our new PIT test method is: "If the sum of distance from target node M to vertices of the
triangle ABC (anchor nodes  A, B and C) is greater than perimeter of the triangle ABC, then the
target node M is outside triangle ABC."
neighboring_node_index=setdiff(find(neighbor_matrix(i,:)==1),neighboring_anchor_index([a b
c]));
                    length(neighboring_node_index);

neighboring_node_rss_of_abc=neighbor_rss(neighboring_node_index,neighboring_anchor_inde
x([a b c]));
      perimeter = (sqrt(((all_nodes.estimated(1,1)-
all_nodes.estimated(2,1)).^2+(all_nodes.estimated(1,2)-
all_nodes.estimated(2,2)).^2))+(sqrt(((all_nodes.estimated(1,1)-
all_nodes.estimated(3,1)).^2+(all_nodes.estimated(1,2)-
all_nodes.estimated(3,2)).^2))+(sqrt(((all_nodes.estimated(2,1)-
all_nodes.estimated(3,1)).^2+(all_nodes.estimated(2,2)-all_nodes.estimated(3,2)).^2)))));
     RSS = (neighboring_node_rss_of_abc)/(length(neighboring_node_index));
         load 'Parameters_Of_Models.mat';
         dist= d0*10.^((Pt-Pl_d0-RSS)./(10*eta));

    if perimeter > dist

Grid_in_triangle_abc=inpolygon(centroid_x,centroid_y,all_nodes.estimated(neighboring_anchor
_index([a b c]),1),all_nodes.estimated(neighboring_anchor_index([a b c]),2));%a grid covered
by a triangle abc
                 gridmap=gridmap+Grid_in_triangle_abc;
%       disp ('unknown node i reside outside triangle abc');

    else

Grid_in_triangle_abc=inpolygon(centroid_x,centroid_y,all_nodes.estimated(neighboring_anchor
_index([a b c]),1),all_nodes.estimated(neighboring_anchor_index([a b c]),2));%a grid covered
by a triangle abc
      gridmap=gridmap-Grid_in_triangle_abc;
%        disp ('unknown node i reside inside  triangle abc');
                  end
                        grid_covered_flag=grid_covered_flag|Grid_in_triangle_abc;
                     end
                end
             end
```

```
            if any(any(grid_covered_flag))
               weight_max=max(max(gridmap(grid_covered_flag)));

weight_max_index=intersect(find(gridmap==weight_max),find(grid_covered_flag==1));

[weight_max_ind_row,weight_max_ind_col]=ind2sub(size(gridmap),weight_max_index);
               all_nodes.estimated(i,:)=mean([weight_max_ind_col
weight_max_ind_row;weight_max_ind_col weight_max_ind_row]); %*grid_length-
0.5*grid_length);
               all_nodes.anc_flag(i)=2;
            end
        end
%         else
          %  disp('since neighboring_anchor_n<3 ,APIT does not work.');
    end
%     end

% ~~~~~~~Localization error result~~~~~~~~~~~~~~~
    figure;
    hold on;
    box on;

plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');%the
anchors
    Unresolved_unknown_nodes_index=find(all_nodes.anc_flag==0);%the unresolved unknown
nodes
    Unresolved_num=length(Unresolved_unknown_nodes_index);

plot(all_nodes.true(Unresolved_unknown_nodes_index,1),all_nodes.true(Unresolved_unknown_
nodes_index,2),'kO');
    resolved_unknown_nodes_index=find(all_nodes.anc_flag==2);%estimated locations of the
resolved unkonwn nodes

plot(all_nodes.estimated(resolved_unknown_nodes_index,1),all_nodes.estimated(resolved_unkn
own_nodes_index,2),'bO');

plot(transpose([all_nodes.estimated(resolved_unknown_nodes_index,1),all_nodes.true(resolved_
unknown_nodes_index,1)]),...

transpose([all_nodes.estimated(resolved_unknown_nodes_index,2),all_nodes.true(resolved_unk
nown_nodes_index,2)]),'b-');
```

```
    axis ([0,all_nodes.square_L,0,all_nodes.square_L]); %auto;
    title(' IM-APIT  Localization error');
    try % Draw a regularly distributed grid line
        x=0:all_nodes.grid_L:all_nodes.square_L;
        set(gca,'XTick',x);
        set(gca,'XTickLabel',num2cell(x));
        set(gca,'YTick',x);
        set(gca,'YTickLabel',num2cell(x));
        grid on;
    catch
        %none
    end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~Draw a regularly distributed grid
line~~~~~~~~~~~~~~~~~~~~~~~~~~');

    disp('Red * indicates the anchor node');
    disp('Blue O indicates the estimated position of the unknown node');
    disp('Black O indicates an unknown node that cannot be located');
    disp('Blue - indicates the positioning error of the unknown node (estimated position and true
position of the connected unknown node)');
     disp(['Total ',num2str(all_nodes.nodes_n),' Nodes: ' ,num2str(all_nodes.anchors_n),' Anchor
nodes,',...
          num2str(all_nodes.nodes_n-all_nodes.anchors_n),' Unknown nodes,
',num2str(Unresolved_num) , ' cannot be located ']);

Localization_error=sum(sqrt(sum(transpose((all_nodes.estimated(resolved_unknown_nodes_ind
ex,:)-all_nodes.true(resolved_unknown_nodes_index,:)).^2))))/...
      (length(resolved_unknown_nodes_index)*comm_r);
        disp(['Localization error is ', num2str(Localization_error)]);

      end
```

APPENDIX B: MATLAB Source Code for APIT Localization Algorithm

```
function apit(all_nodes)
 % parameter setting
clear;
clc;
close all;
all_nodes.square_L = 100;
all_nodes.nodes_n = 100;
all_nodes.anchors_n = 25;
all_nodes.true = [100*rand(1,100)' 100*rand(1,100)'];
all_nodes.estimated =[100*rand(1,25)' 100*rand(1,25)'];
all_nodes.anc_flag = [ones(25,1);zeros(75,1)];
comm_r = 150;
anchor_comm_r = 1;
model = 'Regular model';
neighbor_matrix = +(rand(100,100)<0.25);
neighbor_rss = -161.3221+(161.5)*rand(100,100);

% ~~~~~~~~~~~~~Nodes deployment~~~~~~~~~~~~~~~
figure;
hold on;
box on;
plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');
plot(all_nodes.true(all_nodes.anchors_n+1:all_nodes.nodes_n,1),all_nodes.true(all_nodes.anchor
s_n+1:all_nodes.nodes_n,2),'bo');
axis([0,all_nodes.square_L,0,all_nodes.square_L]);
title('Node Distribution Graph');
try %Draw a regular distribution of grid lines

   x=0:all_nodes.grid_L:all_nodes.square_L;
   set(gca,'XTick',x);
   set(gca,'XTickLabel',num2cell(x));
   set(gca,'YTick',x);
   set(gca,'YTickLabel',num2cell(x));
   grid on;
catch
   %none
end
try %Draw the boundaries of the C area
```

```
    square_L=all_nodes.square_L;
    area=all_nodes.area;
    plot([area(2) area(2)],[area(3) area(4)],'-k',[area(2) square_L],[area(3) area(3)],'-k',[area(2)
square_L],[area(4) area(4)],'-k');
catch
   %none
end
disp('~~~~~~~~~~~~~~~~~~~~~~~~Node distribution~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
disp([num2str(all_nodes.nodes_n),' Nodes',' among them ',num2str(all_nodes.anchors_n),'
Anchor nodes']);
disp('Red * indicates anchor node, blue O indicates unknown node');

 %`~~~~~~~~~~ Topology~~~~~~~~~~~~~~~~~~
figure;
hold on;
 box on;
for i=1:all_nodes.nodes_n
    for j=i+1:all_nodes.nodes_n
        if neighbor_matrix(i,j)==1
            plot(all_nodes.true([i,j],1),all_nodes.true([i,j],2),'-b');
        end
    end
end
plot(all_nodes.true(all_nodes.anchors_n+1:all_nodes.nodes_n,1),all_nodes.true(all_nodes.anchor
s_n+1:all_nodes.nodes_n,2),'ro');
plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');
axis([0,all_nodes.square_L,0,all_nodes.square_L]);
title('Neighborhood graph');
disp('~~~~~~~~~~~~~~~~~~~~~~~~Neighborhood graph~~~~~~~~~~~~~~~~~~~~~~~~~~');
disp([num2str(all_nodes.nodes_n),' Nodes,','among them ',num2str(all_nodes.anchors_n),'
Anchor nodes']);
disp('Red * indicates anchor node, red O indicates unknown node');
disp(['Communication radius:',num2str(comm_r),'m']);
disp(['Anchor node communication radius:',num2str(comm_r*anchor_comm_r),'m']);
disp(['Communication model:',model]);
try
    disp(['DOI=',num2str(DOI)]);
catch
    %none
end
if anchor_comm_r==1
```

```
    disp(['The average connectivity of the network
is:',num2str(sum(sum(neighbor_matrix))/all_nodes.nodes_n)]);
    disp(['The average number of neighbor anchor nodes in the network
is:',num2str(sum(sum(neighbor_matrix(1:all_nodes.nodes_n,1:all_nodes.anchors_n)))/all_nodes.
nodes_n)]);
else
    disp(['The average number of anchor nodes that an unknown node can listen to
is:',num2str(sum(sum(neighbor_matrix(all_nodes.anchors_n+1:all_nodes.nodes_n,1:all_nodes.an
chors_n)))/(all_nodes.nodes_n-all_nodes.anchors_n))]);
    disp(['The average number of unknown nodes in the unknown nodes communication area
is:',num2str(sum(sum(neighbor_matrix(all_nodes.anchors_n+1:all_nodes.nodes_n,all_nodes.anc
hors_n+1:all_nodes.nodes_n)))/(all_nodes.nodes_n-all_nodes.anchors_n))]);
end

    % ~~~~~APIT ~~~~~~~~~~~~~


    unknown_node_index=all_nodes.anchors_n+1:all_nodes.nodes_n;
    row_n=ceil(all_nodes.square_L);    col_n=row_n;
    centroid_x=repmat(((1:col_n)-0.5),row_n,1);
    centroid_y=repmat(transpose(((1:row_n)-0.5)),1,col_n);
    for i=unknown_node_index
disp(['Node ID: ',num2str(i),'  x,y coordintate: ',num2str(all_nodes.true(i,1)),'    '
num2str(all_nodes.true(i,2)),'   RSSI value: ',num2str(neighbor_rss(i))]); % display node ID ,x-y
coordinates and neighbor rss value
        neighboring_anchor_index=find(neighbor_matrix(i,1:all_nodes.anchors_n)==1);
        neighboring_anchor_n=length(neighboring_anchor_index);
        if neighboring_anchor_n>=3
            gridmap=zeros(row_n,col_n);
            grid_covered_flag=zeros(row_n,col_n);
            for a=1:neighboring_anchor_n-2
                for b=a+1:neighboring_anchor_n-1
                    for c=b+1:neighboring_anchor_n
                        %~~Determine if the unknown node i is inside the triangle abc
                        % Approximate P.I.T Test: "If no neighbor of M is further from/close to all three
anchors A, B and C simultaneously,
                        % M assumes that it is inside triangle abc. Otherwise,M assumes it resides outside
the triangle."

neighboring_node_index=setdiff(find(neighbor_matrix(i,:)==1),neighboring_anchor_index([a b
c]));
```

```
neighboring_node_rss_of_abc=neighbor_rss(neighboring_node_index,neighboring_anchor_inde
x([a b c]));

in_out_judge=neighboring_node_rss_of_abc>repmat(neighbor_rss(i,neighboring_anchor_index(
[a b c])),length(neighboring_node_index),1);
                        if
any(sum(transpose(in_out_judge))==0|sum(transpose(in_out_judge))==3)%outside

Grid_in_triangle_abc=inpolygon(centroid_x,centroid_y,all_nodes.estimated(neighboring_anchor
_index([a b c]),1),all_nodes.estimated(neighboring_anchor_index([a b c]),2));%a grid covered
by a triangle abc
                            gridmap=gridmap-Grid_in_triangle_abc;
                        else%inside

Grid_in_triangle_abc=inpolygon(centroid_x,centroid_y,all_nodes.estimated(neighboring_anchor
_index([a b c]),1),all_nodes.estimated(neighboring_anchor_index([a b c]),2));%a grid covered
by a triangle abc
                            gridmap=gridmap+Grid_in_triangle_abc;
                        end
                        grid_covered_flag=grid_covered_flag|Grid_in_triangle_abc;
                    end
                end
            end
            if any(any(grid_covered_flag))
                weight_max=max(max(gridmap(grid_covered_flag)));

weight_max_index=intersect(find(gridmap==weight_max),find(grid_covered_flag==1));

[weight_max_ind_row,weight_max_ind_col]=ind2sub(size(gridmap),weight_max_index);
                all_nodes.estimated(i,:)=mean([weight_max_ind_col
weight_max_ind_row;weight_max_ind_col weight_max_ind_row]); %*grid_length-
0.5*grid_length);
                all_nodes.anc_flag(i)=2;
            end
        end
    end

% ~~~~~~~Localization error result~~~~~~~~~~~~~~
figure;
    hold on;
```

```
    box on;

plot(all_nodes.true(1:all_nodes.anchors_n,1),all_nodes.true(1:all_nodes.anchors_n,2),'r*');%the
anchors
    Unresolved_unknown_nodes_index=find(all_nodes.anc_flag==0);%the unresolved unknown
nodes
    Unresolved_num=length(Unresolved_unknown_nodes_index);

plot(all_nodes.true(Unresolved_unknown_nodes_index,1),all_nodes.true(Unresolved_unknown_
nodes_index,2),'kO');
    resolved_unknown_nodes_index=find(all_nodes.anc_flag==2);%estimated locations of the
resolved unkonwn nodes

plot(all_nodes.estimated(resolved_unknown_nodes_index,1),all_nodes.estimated(resolved_unkn
own_nodes_index,2),'bo');

plot(transpose([all_nodes.estimated(resolved_unknown_nodes_index,1),all_nodes.true(resolved_
unknown_nodes_index,1)]),...

transpose([all_nodes.estimated(resolved_unknown_nodes_index,2),all_nodes.true(resolved_unk
nown_nodes_index,2)]),'b-');
    axis([0,all_nodes.square_L,0,all_nodes.square_L]);
    title('APIT Localization error');
    try % Draw a regularly distributed grid line
        x=0:all_nodes.grid_L:all_nodes.square_L;
        set(gca,'XTick',x);
        set(gca,'XTickLabel',num2cell(x));
        set(gca,'YTick',x);
        set(gca,'YTickLabel',num2cell(x));
        grid on;
    catch
        %none
    end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    disp('~~~~~~~~~~~~~~~~Draw a regularly distributed
gridline~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');

    disp('Red * indicates the anchor node');
    disp('Blue O indicates the estimated position of the unknown node');
    disp('Black O indicates an unknown node that cannot be located');
```

```
    disp(' Blue - indicates the positioning error of the unknown node (estimated position and true
position of the connected unknown node)');
    disp(['Total ',num2str(all_nodes.nodes_n),' Nodes: ' ,num2str(all_nodes.anchors_n),' Anchor
nodes,',...
        num2str(all_nodes.nodes_n-all_nodes.anchors_n),' Unknown nodes,
',num2str(Unresolved_num) , ' cannot be located ']);

Localization_error=sum(sqrt(sum(transpose((all_nodes.estimated(resolved_unknown_nodes_ind
ex,:)-all_nodes.true(resolved_unknown_nodes_index,:)).^2))))/...
       (length(resolved_unknown_nodes_index)*comm_r);
    disp(['Localization error is ', num2str(Localization_error)]);

end
```