Reducing the Impact of Crowd Geo-localization on Battery Life by Proposing Novel Information Distribution Algorithm

By Samuel Asferaw Demilew

Supervisors:

Prof. Jean-Marc Pierson, IRIT, Université Paul Sabatier, France

Dr. Georges Da-Costa, IRIT, Université Paul Sabatier, France

Dr. Dejene Ejigu, Addis Ababa University, Ethiopia

Presented in Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Information Technology (IP Networking and Wireless Mobile Internet)

Addis Ababa University
Addis Ababa, Ethiopia
March 2017

Addis Ababa University

School of Graduate Studies

This is to certify that the thesis prepared by Samuel Asferaw Demilew, entitled: *Reducing the Impact of Crowd Geo-localization on Battery Life by Proposing Novel Information Distribution Algorithm* and submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy in Information Technology (IP Networking and Wireless Mobile Internet) complies with the regulation of the university and meets the accepted standards with respect to originality and quality.

Signed by the Examination Committee:

Examiner: Prof. Devarajan Gopal Signature_______________ Date_______

Examiner: Dr. Yalemzewd Negash Signature_______________ Date_______

Advisor: Prof. Jean-Marc Pierson Signature_______________ Date_______

Advisor: Assoc. Prof. Georges Da-Costa Signature_______________ Date_______

Advisor: Dr. Dejene Ejigu Signature_______________ Date_______

Dr. Yihenew Wondie________________________________

Chair of Department or Graduate Programme Co-ordinator

Abstract

Reducing the Impact of Crowd Geo-localization on Battery Life by Proposing Novel Information Distribution Algorithm

Samuel Asferaw Demilew

Addis Ababa University, March 2017

Geo-localization is an inevitable challenge and problem which has been studied for many years when dealing with wireless nodes as they are equipped with limited life batteries. Emergence of 'crowd of wireless node users' across the world (especially in big cities and emerging big cities in developing countries) and increasing sensor network applications are exacerbating the problem. Existing wireless node localization systems depend on power hungry systems like GPS. Advances in battery design have been slow and difficult; therefore, it is essential for alternative strategies to be employed to realize the goals of reducing the impact of crowd geo-localization on battery life. Based on the type of information required for localization, localization methods can be divided into two categories: range-based and range-free protocols. Since range-based methods demand battery hungry dedicated ranging components, range-free methods are explored. This thesis focuses on the investigation of novel smart algorithms for crowd geo-localization.

We have proposed the following three but related smart novel rang-free localization algorithms for wireless networks:

- Centre of the Smallest Communication Overlap Polygon (CSCOP) localization algorithm
- Selective Anchor Nodes CSCOP localization algorithm
- Immune to Radio Range Difference (IRRD) localization algorithm

To evaluate the accuracy and reliability of the proposed algorithms, we have conducted extensive MATLAB simulations and comparisons with other state-of-the-art related work on:

average, minimum, and maximum location error in a scenario of both sparse and dense crowd. Moreover, to see the probability of error distribution of the proposed algorithms, probability distribution of location error is developed using MATLAB normal fit function. We have also investigated computational complexity analysis of the algorithm using "*Big O*" notation along with detailed elementary operations used in the algorithms.

Results obtained from our experiments show our algorithms significantly out-perform other related work in the domain in case of both sparse and dense crowd geo-localization. When we compare localization accuracy performance of our algorithms in case of sparse and dense crowd scenarios, they perform better in dense crowd scenario. For example, the average location error of our CSCOP algorithm in case of dense crowd with 30 anchor nodes involved is 3.7176 (in % radio range) which means 0.7435m error while its error in sparse crowd with 8 anchor nodes is 8.3552 (in % radio range) i.e., 1.6710m in our simulation scenario, where communication range is 20m. The algorithms, in addition to estimating reliable location, also define the smallest communication overlap polygon (SCOP) which can be applied in search and rescue operations as margins where the node situated. Moreover, unlike other related works which require at least three anchor nodes, the proposed algorithms work starting with two anchor nodes.

Furthermore, unlike other related algorithms, our Immune to Radio Range Difference algorithm breaks the traditional assumption that says "anchor nodes have the same radio range", by working in both homogeneous and heterogeneous radio ranges which is a break through finding in the domain area. Although, this work focuses on the crowd networks, applications of the proposed geo-localization algorithms could range to all wireless networks from cellular to sensor.

Keywords: Anchor Nodes, Range-free Localization, Smallest Communication Overlap Polygon, True Intersection Points, Unknown Nodes, Wireless Networks, Wireless Nodes

Acknowledgments

It is my deep hearted pleasure to express my gratitude to all individuals and institutions that have helped and encouraged me during my doctoral study.

First of all, my sincere gratitude goes to my supervisor Prof. Jean-Marc Pierson, whose enthusiasm and integral view on research with a mission to come with "novel and high-quality work and not less," have made a lasting impression on me. My sincere gratitude also extends to my co-supervisors Dr. Georges Da-Costa and Dr. Dejene Ejigu, whose encouragement, guidance, and critique were valuable at each step of my work.

I am deeply indebted to Mrs Judy Price who is UK citizen for editing and reviewing the language of the dissertation and encouraging me to work hard to come with "high quality work".

I cordially thank all colleagues in the IRIT laboratory, Toulouse, France specifically those in the SEPIA research group, who encouraged and gave feedback during progress presentations of the work. Similar thanks go to my colleagues in IT Doctoral Programme, AAU, particularly to Zelalem Fikire and Netsanet Getnet. Parallel thanks go to Prof. Lionel Brunie in INSA de Lyon, France for coordinating the visit to our supervisors in France. I also wish to extend my sincere gratitude to all my friends and relatives whose encouragement contributed to the success of this work.

My special thanks go to my family (brothers: Assefa, Sileshi, and Fantahun; sisters: Alganesh, Gedamnesh, Mahilet, and Lidiya) who have always encouraged and supported me during my study. My special thanks extends to my fiancée Eleni Dagnaw who encouraged me to stay strong and fresh on the work.

This work is dedicated to my father, Asferaw Demilew, whom I missed when I started my doctoral study. He encouraged me to go ahead with full strength and passion in all endeavours of my life in general and in my doctoral study in particular.

Thanks God, may your name be honoured and glorified!

Samuel Asferaw Demilew, March 17, 2017, Addis Ababa, Ethiopia

Table of Contents

List of Tables

List of Figures

Acronyms

AOA	Angle-of-Arrival
A-GPS	Assisted Global Positioning System
CPE	Convex Position Estimation
CSCOP	Centre of the Smallest Communication Overlap Polygon
FIPs	False Intersection Points
GPS	Global Positioning Systems
GSM	Global Mobile Systems
IPs	Intersection Points
ISM bands	Industrial, Scientific and Medical radio bands
IRRD	Immune to Radio Range Difference
LAN	Local Area Network
LBAs	Location Based Applications
LBSs	Location Based Systems
MDS-MAP	Multidimensional Scaling MAP
non-LOS	non Line-of-Sight
PAN	Personal Area Network
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator

SCOP	Smallest Communication Overlap Polygon
SOM	Self-Organizing Map
TDOA	Time-Difference-of-Arrival
TIPs	True Intersection Points
TOA	Time-of-Arrival
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Networks
WSN	Wireless Sensor Networks

CHAPTER 1 INTRODUCTION

1.1 Research Background

Locating wireless devices has always been a critical problem; more critical today, as the number of context-aware applications continually grows. Large number of wireless device users and sensor network applications are emerging everywhere. The increasing pervasiveness of 'smartphones' is stimulating a number of Location-Based Applications (LBAs) to become increasingly popular and be adopted by mobile users for different Location Based Services (LBS) i.e. social-networking, business needs, and entertainment (as shown in Figure 1-1) [1]. Some currently popular LBAs include mobile social networking, healthcare, travel and public transport.

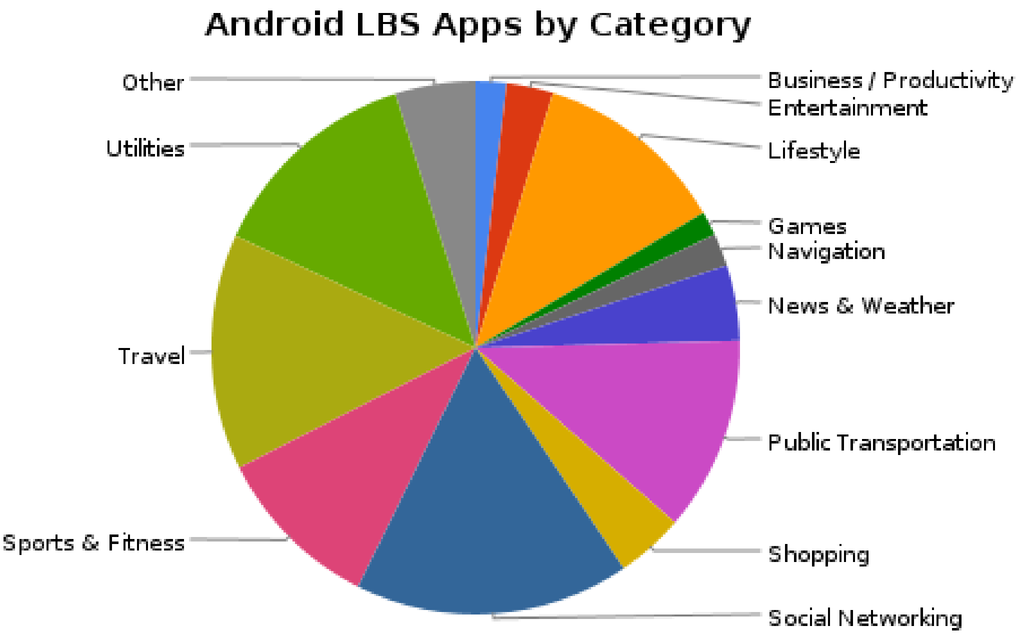


Figure 1-1: Android and iPhone LBS Application by % Category (reported by Skyhookwireless 2016)

Of all localization technologies, Global Positioning System (GPS) has been widely used for different location based services; however, the active use of GPS applications can cause device battery drain owing to their power-intensive location-sensing operations.

Despite an increase in processing power, memory and sensing capabilities, small wireless nodes continue to suffer battery life limitation hindering the active utilization of LBAs. For example, the typical battery capacity of 'smartphones' is barely above 1000 mAh (e.g., the lithium-ion battery of HTC Dream smartphones has the capacity of 1150 mAh) [2]. Unfortunately, GPS, the core enabler of LBAs (Location Based Applications) in many current LBAs, is power-intensive; its aggressive usage can cause complete battery drain within a few hours [3]. While the aggressive usage of GPS is specific to different applications, several LBAs - particularly local traffic and social networking - benefit from continuous location updates. Real Time Traffic, for instance, requires continuous GPS location updates showing that unless a mechanism is designed not to use GPS intensively and continuously or other alternatives, power scarce mobile devices will continue to suffer from a shortage of battery life which hinders LBSs (Location Based Services).

Importantly, in the last few decades, a crowd of wireless node users has become a common phenomenon and this is set to increase. Figure 1-2 illustrates the case of a globally moving mobile phone user crowd in cities or crowd of deployed sensor networks. In the crowd there are two types of wireless nodes: location aware (anchor nodes) and location unaware (unknown) nodes. If each wireless node in a crowd depends on GPS for localization, it will result in huge consumption of scarce battery. To cope with this growing challenge, a mechanism to reduce the impact of crowd geo-localization on battery life to benefit LBSs for sustainable LBAs is a necessity.

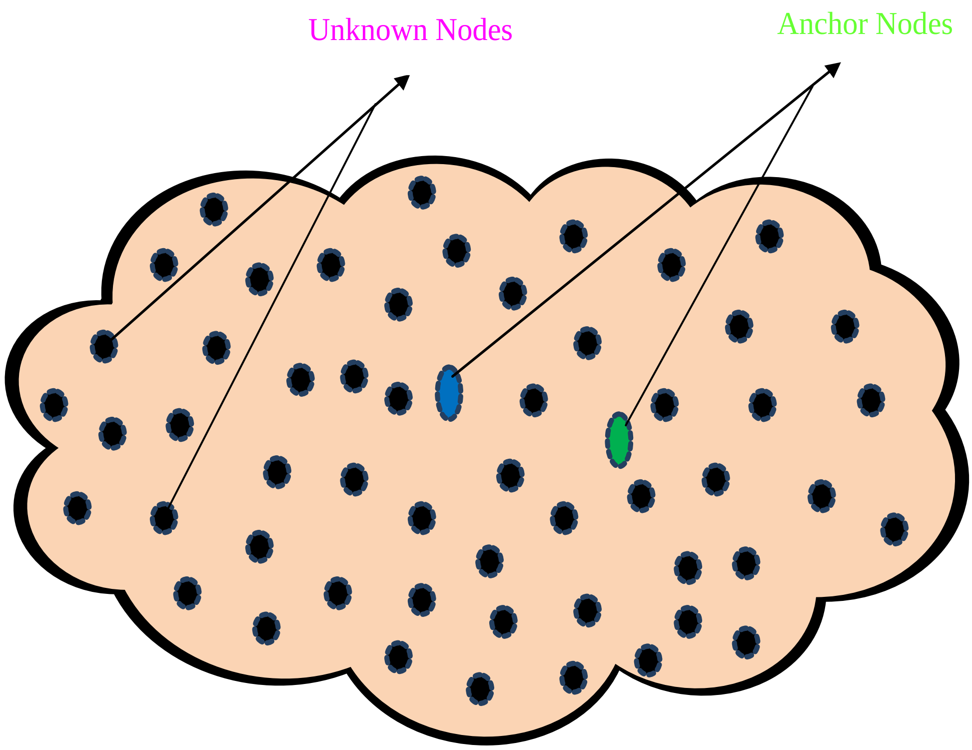


Figure 1-2: Crowd of Sensor Nodes

Many solutions have been proposed to improve the battery life of wireless devices, but little rigor and attention has been devoted to reducing the impact of crowd geo-localization on battery life.

LBA developers are suggesting a reduction in the use of GPS by increasing location-update intervals (say, to more than a minute), thus allowing GPS hardware to sleep between successive location-updates. Such a simple solution can improve battery life by forcing applications to request location information less frequently, but is still battery hungry because it still depends on battery hungry GPS. EnLoc [3] modelled the trade-off between accuracy and energy consumption of wireless device geo-localization as an optimization problem.

To address this optimization problem, this thesis research attempts to present a novel range-free geo-localization method which compromises the trade-off between energy and accuracy by introducing sharing of location information in the crowd using wireless nodes inherent radio frequency (RF) node connectivity metric i.e., it looks at the challenge of reducing the impact of the growing crowd of small wireless nodes geo-localization on battery life. To cope with this challenge, it attempts to develop geo-localization algorithms which are both energy and precision efficient: novel range-free smart co-operative geo-localization algorithms. A crowd is the ideal environment in which to share location information among wireless nodes where a location aware (anchor) node can easily share its location to location unaware (unknown) wireless nodes. Hence, we developed smart geo-localization algorithms which enable unknown wireless nodes to locate themselves based on shared location information of anchor nodes using only existing infrastructure.

1.2 Research Motivation

Much research has been done aimed at obtaining high localization accuracy by using specific hardware infrastructures. However, these high accuracy research results are in trade-off with high battery power usage: firstly, power intensive, expensive, specialized hardware is used in contrast to power scarce small wireless devices; secondly, crowds of small wireless device users and crowds of sensor networks are emerging in cities and the trend will continue due to the number of emerging big cities especially in developing countries and increasing new applications of sensor networks. If this growing crowd depends on battery hungry localization systems, GPS for example, it will diminish wireless devices' battery life as their battery energy is a scarce resource. This research; therefore, using growing crowds of small wireless nodes as an opportunity, desires to focus on reducing the impact of crowd geo-localization on battery life by proposing novel range free algorithm.

1.3 Research Problem Statement

Wireless node geo-localization is a difficult problem to solve because of the low cost demand due to their inherent scarce battery problem. This made small wireless node localization a tough problem to be addressed well. The commonly used geo-localization system for wireless nodes is GPS, but it is battery intensive application. Since emergence of wireless nodes, their number and kind is increasing from time to time. Today, it is common to see crowds of cell phone users in cities and crowds of sensor nodes deployed for various applications. In such a scenario, the use of battery hungry location services like GPS by all members of the crowd means a high energy cost. To address this problem, location information of anchor nodes in a given crowd can be shared using devices like Bluetooth and WiFi which results in significant energy saving from each wireless node perspective, and a huge saving when we look from the crowd point of view.

Generally, wireless node geo-localization can be broadly classified into two: range-based and range-free schemes. Since range-based method, like GPS, demands ranging component, it is energy intensive method but usually gives better accuracy when compared to range-free methods. On the other side, rang-free methods do not demand ranging component; thus, they are not energy intensive but they usually give low accuracy when compared with range-based ones. Since wireless nodes have scarce battery, the option at hand is optimization of energy and accuracy on either of the above two methods that is by decreasing energy usage on the range-based method or by increasing accuracy on the range-free methods. This dissertation attempts to investigate this problem focussing on the latter solution by proposing novel range-free smart geo-localization algorithms.

Thus, this study attempts to answer the following research questions:

1) How to develop smart, energy-accuracy efficient, range-free crowd geo-localization algorithm?
2) How to increase the localization accuracy of range-free geo-localization methods?
3) How to make range-free geo-localization algorithm immune to anchor nodes' radio range difference (work with both heterogeneous and homogeneous radio ranges of anchor nodes)?

1.4 Research Objective

The general objective of this research is to develop a smart and novel range-free geo-localization algorithm to reduce the impact of crowd geo-localization on battery life. Whereas the specific objectives of the research look the following:

Specific Objectives:

1) To survey related works on energy-accuracy efficient for small wireless device geo-localization. This overview is used as the basis for a comprehensive understanding of geo-localization technologies, methods and algorithms in order to propose novel algorithm that addresses the research problem.

2) To develop a geometrical and a mathematical model for the problem of co-operative range-free geo-localization of wireless devices.

3) To develop smart novel cost effective range-free geo-localization algorithm that locates location unknown nodes in the crowd using shared location information of neighbour anchor nodes.

4) To develop a novel range-free geo-localization algorithm which is resilient to radio range difference of anchor nodes without the need to install any additional dedicated infrastructure.

1.5 Operational Definitions

Geo-localization: Finding location/position of a wireless device in wireless network. In this thesis, word geo-localization is used interchangeable with word localization as synonyms (with no meaning difference).

Radio Range: Refers to communication range or visibility of nodes. A wireless node can be visible when it is in the communication range (or radio range) of that node.

Anchor node: A wireless node in the crowd which is aware of its location. This node shares its location information to the location unaware node to help estimate its location.

Unknown node: A wireless node in the crowd unaware of its location. This node uses shared location information of anchor nodes to estimate its location.

Crowd: A sparsely or densely populated wireless node network.

1.6 Scope of the Study

This dissertation attempts to develop a novel co-operative smart geo-localization algorithm for wireless devices focusing on range-free methods. Literature on the domain classifies range–free methods into two: Local and Hop Counting (DV Hop) methods. This thesis focuses on the first method assuming an unknown node has at least two neighbour anchor nodes in the crowd. Hence, it is delimited to the development of novel range-free geo-localization algorithms focusing on range-free local methods. To evaluate the proposed geo-localization algorithms MATLAB simulation is used.

1.7 Contribution of the Study

This study contributes to range-free wireless device localization techniques. The main target is to design small wireless device co-operative geo-localization algorithms which extend the battery life of wireless nodes with reliable location accuracy. Hence, the main contributions of this research can be summarized as follows:

- Three but related smart and novel range-free geo-localization algorithms for green energy-accuracy efficient localization of small wireless devices, based on co-operative localization in today's growing crowd of wireless devices to work in infrastructure-less environments is developed in a way that has neither been reported elsewhere in current literature nor implemented in commercial solutions.
- The proposed algorithms outperform other state-of-the-art algorithms in location accuracy in case of both sparse and dense crowd.

- Geometrical and mathematical modelling of the smart co-operative range-free geo-localization problem in wireless networks is presented.
- Unlike other related algorithms which require at least three anchor nodes to estimate location of the unknown node, the proposed algorithms work with a minimum of two.
- Unlike other related algorithms which assume “anchor nodes have the same radio communication range”, breaking this tradition, a novel Immune to Radio Range Difference (IRRD) algorithm is proposed. This algorithm works in both heterogeneous and homogeneous radio ranges.
- The proposed algorithms, in addition to estimating the location of the unknown node, also define the Smallest Communication Overlap Polygon (SCOP) of the anchor nodes which can be used as the smallest likely area for search and rescue operation by defining TIPs as the search boundaries.

1.8 Organization of the Thesis

This thesis presents our proposed approach in solving the impact of crowd geo-localization on battery life in wireless networks in the smart range-free geo-localization paradigm. The organization of the presentation of the thesis is as follows:

Chapter 2 surveys related literature in the domain. This chapter surveys general literatures on geo-localization in wireless networks.

Chapter 3 reviews the related work in the domain. This chapter reviews related work (range-free methods and algorithms) on geo-localization in wireless networks.

Chapter 4 presents and discusses the proposed algorithms. In this chapter first each proposed geo-localization algorithm is presented, then computational complexity of each of these algorithms is discussed.

Chapter 5 first describes the research methodology used in designing and modelling the research problem, and method used in simulation and performance evaluation of the proposed algorithms. Then, it presents implementations and evaluations of the proposed algorithms. In this chapter the performance of the algorithms is evaluated and compared with state-of-the-art algorithms.

Chapter 6 gives conclusions and highlights for future work. Finally, following chapter seven, references used in the dissertation, an appendix of publications and MatLab source code of implementation software are presented.

Deliberately left blank

CHAPTER 2 SURVEY OF RELATED LITERATURE

Geo-localization is the process of estimating the physical location of a device. Estimating a location of a wireless device requires a reference location. To explain the process of localization, we classified wireless devices (nodes) into two: anchor nodes and unknown nodes. Anchor nodes are a network entity with known location. They are used as reference location. Unknown nodes are those nodes whose location is estimated based on the anchor nodes.

2.1 Underlying Physical Layer of Geo-localization Technologies

The performance of the localization process is directly affected by the underlying wireless technology used. This underlying technology defines the physical layer of the localization scheme. It defines the different physical layer specifications like: operating frequency, bandwidth, modulation technique, transmission power level, and antenna diversity. The frequency spectrum, where the localization scheme is deployed, directly influences localization performance. Physical spectrums for localization techniques are: infrared (IR) [4], ultrasound (US) [5] [6] and radio frequency (RF) [7] [8].

Technologies using the same frequency spectrum may have different physical layer specifications resulting in different localization performances. For instance, Bluetooth and 802.11 occupy the same ISM band; however, they have different physical layer related parameters like transmission power and modulation techniques. The performance of Bluetooth is slightly better than 802.11 [9] due to Bluetooth having lower transmission power which limits its transmission range.

Among all the parameters of physical layer, operating frequency plays the most significant role in defining localization characteristics. Hence, based on the underlying operating frequency, we classified the wireless devices into three: Infrared (IR), Ultrasound (US) and Radio Frequency (RF). Frequency (*f*) = 1/Time Period (*T*). Hence, at higher *f*, period T is small and a finer time resolution is offered. From Friis free space equation (2.1),

$$\text{Received Power} \propto \left(\frac{\lambda}{d}\right) \tag{2.1}$$

where, wavelength $\lambda \propto 1/f$ and d is the distance between the transmitter and receiver; attenuation of the signals increases with frequency [10]. In general, higher frequency leads to higher localization accuracy but shorter range.

2.1.1 Infrared (IR)

Infrared was used in Active Badge [4] which is one of the earliest localization schemes. Here, an IR badge is an anchor node periodically transmitting unique ID. To transmit the received IDs to a central server each room has an IR receiver because IR is limited to line of sight transmission. Then, the server maps unknown nodes' locations to the corresponding room. IR based localization schemes usually result in lower accuracy and are affected by sunlight. Moreover, it is not scalable for large deployment areas. Due to this, IR is not a preferred underlined physical layer used for localization.

2.1.2 Ultrasound (US)

Unlike infrared signals, ultrasound signals can propagate through the walls. For localization purpose, ToA can be precisely measured using existing low-cost devices. In US based localization, ToA provides more robust location estimate when compared with signal strength (SS) measurements. Due to this it is a preferred method for US based localization. In Cricket [6], unknown node uses ToA of US signals from various anchor nodes to estimate its location. Since RF travels much faster than US, a RF signal is used by anchor nodes to indicate the start of US transmission in Cricket. An unknown node measures the time difference between the reception of RF and US signals from anchor nodes. This difference is the ToA for US signals from anchor nodes to unknown node. Finally, using the broadcast anchor nodes location information in the initial RF transmission, the unknown node estimates its location.

2.1.3 Radio Frequency (RF)

RF is the commonly used underlying wireless technology by most communication networks because RF location estimation systems are less expensive than their dedicated counterparts. Localization based on RF underlying technology of communication network, lowers deployment cost. The RF localization can use received SS measurements (example, RADAR [8]) or ToA (example, GPS). Time of arrival for RF signals requires precision time resolution (in order of nanoseconds) increasing hardware cost. Since higher frequencies are used, most surfaces act as reflectors and a multipath effect predominates; thus, RF based ToA techniques are preferred for large capital, open area deployments (example, [7]). On the other hand, RF based (3-10 GHz) ultra wide band

(UWB) has a nanosecond precision clock and is inherently inexpensive estimating ToA because of its tight transmitter receiver synchronization [11].

2.1.4 Summary of Underlying Physical Layer of Geo-localization Technologies

The location of a device is usually estimated by monitoring a distance dependent parameter such as wireless signal strength from a base station with known location. In practical deployments, signal strength varies with time and its relationship to distance is not well defined. Due to this challenge, RF is also exploited for range-free methods which use node connectivity (visibility) metric to estimate the location of unknown nodes. Unlike range-based methods (discussed in Chapter 3 sub-section 3.1) which measure ToA, SS, TDoA or AoA, range-free (discussed in Chapter 3 sub-section 3.2) methods use node connectivity to estimate location of the unknown node so avoiding the cost of measuring ToA, SS, TDoA or AoA by ranging hardware components of wireless node. Anchor nodes, when they are in the communication range of a given unknown node, are visible to the unknown node which can estimate its location based on these visible neighbour anchor nodes [12]. Hence, we based our work on this method.

Some of the key parameter values for different frequency spectrums are presented in Table 2-1.

Table 2-1: Parameters of Underlying Physical Layer of Geo-localization Technologies

	IR	**US**	**RF**
Frequency (Hz)	3000 G	20 K – 40 K	3 K – 300 G
Speed in air (m/s)	3×10^8	343 (20^{o}C)	3×10^8
Geo-localization Technologies	IrSimple (www.irda.org)	SONAR Medical Imaging	GPS, Bluetooth, 802.11(WiFi), UWB

2.2 Geo-localization Technologies in Wireless Networks

Today wireless networks are becoming more popular than the well established and extensively deployed wired communication networks of the past. The popularity of the wireless networks is mainly due to its nature of ubiquitous access and new services for mobile users allowing access from many locations and free roaming. Location Based Applications (LBAs), such as network planning, location based services, law enforcement and improving network performance demand the knowledge of the physical location of mobile user devices (like phones, laptops, PDAs and sensor nodes).

There are different technologies which are used to estimate the physical location of nodes in wireless networks. The following subsections present the prominent ones:

2.2.1 Global Positioning System (GPS) based Geo-localization

Satellite-based navigation is the widely used technology for outdoor navigation. GPS consist of 24 satellites (21 active) which revolve round the earth at an altitude of approximately 12,000 miles once every 12 hours [13]. For the framework presented in this dissertation, the satellites act as anchor nodes while the GPS receiver is an unknown node as shown in Figure 2-1.

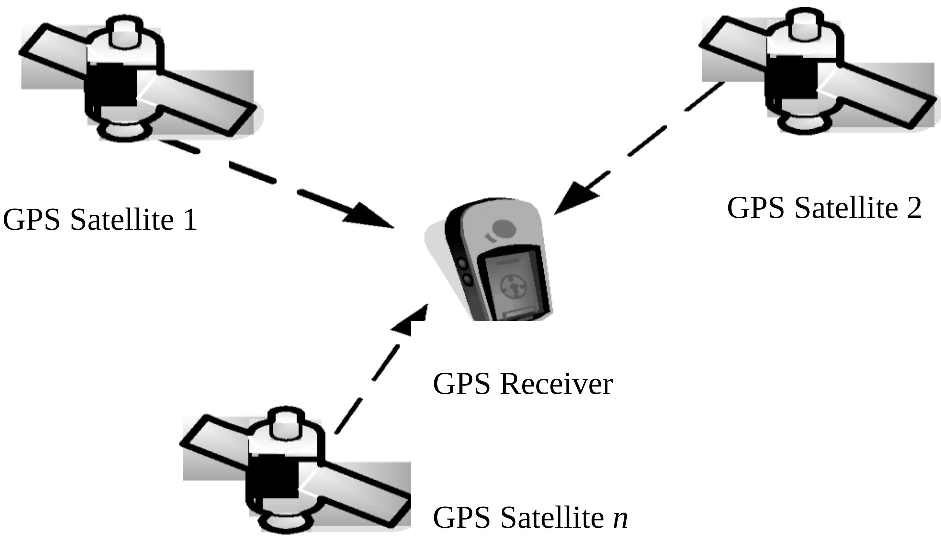


Figure 2-1: GPS Localization Framework

The satellites broadcast their exact location to the GPS receivers by transmitting "almanac" (approximate orbit information for all satellites) and "ephemeris" (precise satellite orbit information for short sections) data. An unknown node (GPS receiver) measures ToA from different anchor nodes (GPS satellites) to estimate its distance. A GPS receiver identifies each satellite by the pseudorandom (PN) code which is unique to each satellite. The PN code between a transmitter and receiver is synchronized prior to any data exchange. The receiver and satellites have to be time synchronized. The receiver may choose any four appropriate satellites or all 10 of the satellites in view ('all-in-view' scheme) to estimate its location. Using the ToA information from at least four satellites, the receiver can estimate its location in 3-D space (longitude, latitude and altitude). The location accuracy of GPS systems is about 6-12 metres [13]. GPS can be used in conjunction with any wireless network (example, cellular network) or as an independent scheme to determine location. GPS is a RF based localization system operating in L1 band (1575.42 MHz) which is a high operating frequency with many multi-paths in a cluttered environment and prohibits GPS signal propagation from building interiors. GPS gives inaccurate results in cluttered or indoor environments where the direct path to the satellite is blocked so its usage is limited to open, outdoor environments.

The estimation method used in GPS is multilateration. Since this technique is affected by outliers, many satellites ('all-inview' mode) can be considered for location estimation to reduce the dependence of estimation on outliers. Since location is estimated at an unknown, GPS is a client-based localization scheme with user privacy. The satellites convert PN codes (P-code) to an encrypted code (P(Y)-code) before transmission to avoid spoofing of satellite signals by malicious users. Thus, GPS may be considered a secure localization scheme for military applications. However, civilian GPS signals use only P-code and can be easily compromised since these codes are not very strong. The receiver can be deceived by an attacker transmitting fake GPS reference signals [14]. The GPS receiver itself can be tampered with to display false location, but this can be avoided by using tamper proof hardware for the GPS receiver. This cannot guarantee security [15]. Hence, civilian GPS is insecure and considered as an open system.

GPS services can only be extended and improved in indoor environments. Consequently, GPS cannot be used as the only positioning technology to meet all requirements in all types of terrain. Though possible in some situations [16], GPS signals cannot penetrate sufficiently to most indoor environments to be used by a receiver. In urban environments and other Radio Frequency (RF) shadowed environments, satellite navigation usually performs poorly. Moreover, the Time to First Fix (TTFF) of a cold start of a GPS device can take a few minutes [17], far too long for many applications. In spite of the fact that GPS localization systems are known for their battery hungriness, GPS receivers are increasingly incorporated in ‘smartphones’. Weyn M. [18] says GPS is a natural choice for integration into ‘smartphones’ for an opportunistic seamless localization system. GPS currently provides the most accurate technique for localization of mobile devices; normally locating a device with errors of about 10 metres [19] [20]. However, it has the following major drawbacks:

Firstly, as we can see the Table 2.2 on page 28, the most energy hungry localization technology of all is GPS; for example, Vtrack [20] and MicroBlog [19] report that a phone continually using GPS can last for 10 hours before its battery is drained, while Enloc [3] reports 9 hours.

Secondly, if there is no line-of-sight between a GPS device and the satellites then location information cannot be determined. The immediate consequence of this is GPS often does not work in indoor environments and in city centres with many high-rise buildings [21].

2.2.2 Assisted GPS (A-GPS) Based Geo-localization

A-GPS geo-localization systems are set up to resolve i) the long delay occurred in locating a mobile wireless node when using GPS ii) obstructions block the view from a handset to a GPS satellite [17]. A-GPS can be accurate up to 10 meters but trade-off of high energy cost. Wireless A-GPS operates on wireless networks like WiFi and GSM. Figure 2.2 shows principle of A-GPS.

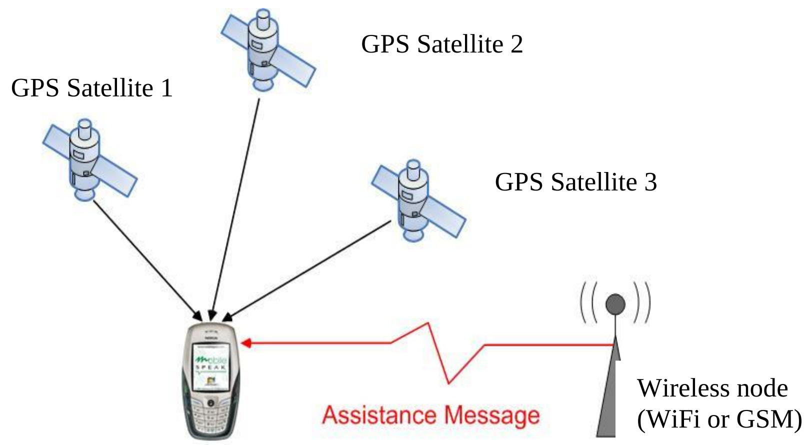


Figure 2-2: Principle of A-GPS

A-GPS implementation requires A-GPS circuitry inside the phone. A-GPS also requires message exchanges with an A-GPS location system in the infrastructure [17].

2.2.3 Global System of Mobile (GSM) based Geo-localization

Global System of Mobile/Code Division Multiple Access (GSM/CDMA) mobile cellular network has been used to estimate the location of outdoor mobile clients. However, this method usually gives poor location accuracy which does not meet the accuracy requirement of most LBSs. For example, the accuracy of the method using cell-ID or enhanced observed time difference (E-OTD) gives low accuracy, in the range of 50–200 m, of course, depending on the cell size. The accuracy is higher in densely covered areas (e.g, urban places) and much lower in sparsely covered areas (e.g, in rural environments) [7]. The GMS/CMDA frequencies in different regions are different. Generally it falls in 850MHz, 900MHz, 1800MHz, and 1900MHz bands. The GSM/CDMA network is already covered in most buildings; hence, there is no or less need for extra infrastructure.

Most GSM based positioning methods use the location of the different cell towers sensed by the mobile device and can be used in network-based or terminal-based positioning. For network-based positioning the co-operation of the mobile operator [18] is required. Consequently, using standard hardware alone or information like cell tower locations from the mobile operator, TOA, TDOA, AOA and pure RSS based methods are not feasible either. According to authors in [18], the only viable method is to use a fingerprint based method similar to WiFi along with RSSI. The fingerprint can be used for Cell-ID localization or an RSS fingerprinting localization. Cell-ID localization will only identify the area cell which is used by the mobile devices. Nevertheless, as we can see from Table 2.2, this technology is the least energy-hungry technique with a large coverage.

2.2.4 Wireless Fidelity (WiFi) based Geo-localization

WiFi is the most used and popular indoor wireless localization technology because it is a widely available network infrastructure [22]. It has a series of standards in IEEE 802.11 using two licence-exempt bands: 2.4 GHz, and 5 GHz. Currently, most commercial products, such as phones, laptops and tablets, support WiFi. Hence, the infrastructure and user device cost can be very low. Wi-Fi localization systems use a radio propagation model to find the distance to the access points. Then, techniques like TOA, TDOA can be used to estimate the location of a mobile device. However, among the methods of localization using WiFi technology, multilateration using timing is difficult to achieve because it is not feasible to get precise timing information from standard of-the-shelf devices in a generic way. This makes time dependent localization techniques: Time Difference of Arrival (TDOA), Time of Arrival (TOA) and Round Trip Time (RTT) not feasible for WiFi generic hardware. Multipath distortion and variability of Wi-Fi signal strength in time limit the accuracy of such techniques.

Currently, popular Wi-Fi localization solutions are based on the scene analysis technique, called the fingerprinting technique. This method uses the Received Signal Strength Indication (RSSI) to measure the strength of signals received from the surrounding access points at discrete locations in space. Such a radio map has to be built in an offline phase (before the system is operational). Then, the estimation of the position is made by matching the measured RSSI from several access points (AP) with the RSSI values of the previously (offline) calibrated location points stored in a radio map database. The accuracy of the fingerprinting technique is expected to be in the range of 1 to 10 m depending on the offline-built radio map and the density of APs, [23] [22].

The major drawback of this method include: i) the offline construction of the RSSI radio map generated in advance ii) updating is required whenever there is removal/relocation of the existing access points and installation of new ones [24]; iii) the other fundamental problem in Wi-Fi fingerprinting is that heterogeneous mobile devices measure radio signal strength differently necessitating either a calibration for each new wireless device or specialized methods, like hyperbolic location fingerprinting [25].

RADAR [8] was one of the first Wi-Fi fingerprinting localization systems. Since then many Wi-Fi based localization systems have emerged. For example, Hensen et al. [26] propose SmartCampusAAU as a platform which enables indoor localization and navigation supporting both device- and infrastructure-based positioning. The work of Laoudias et al. [27] describes the localization approach based on signal strength differences which is robust to device variations and maintains the localization accuracy regardless of the number and type of contributing devices. In addition to research works, various commercial WiFi fingerprinting based localization systems are available on the market, like Ekahau Real-Time Location System [28], Skyhook Wireless [1] and Navizon indoor location solutions [29].

In general, to complement GPS, WiFi based localization is used in indoor environments. As we can see from Table 2.2 WLAN (WiFi) stands as the second most energy hungry localization technology next to GPS.

2.2.5 Bluetooth based Geo-localization

Bluetooth is a wireless technology that can be used for localization and tracking. It is a personal area network standard. It is widely used for short distance communication like earphones, cell phones. It uses a very low transmission power; as a result, the coverage of Bluetooth is shorter than WiFi and other WLAN technologies. Thus, Bluetooth is not suitable for large area localization. Like WiFi, it uses 2.4 GHz and 5 GHz bands. Bluetooth localization systems have similar working principles as the self-localization schemes of sensor networks, both types of systems work based on obtaining the range information to anchor devices or access points and exploring unknown device locations using various algorithms. Most of the available research and commercial systems use the RSSI for calculating distances between Bluetooth devices and finally use trilateration to estimate location of the unknown node. Moreover, many other reported systems have explored proximity and cell-based approach. Bluetooth geo-localization methods are mainly based on RSSI and Link Quality (LQ) measurements [30]. Cell-based methods depend on the proximity (visibility) of Bluetooth beacons for estimating the position of a device and are gaining popularity with the realization of Low Energy Bluetooth standard. The localization accuracy of the system is 1 - 5 m and depends on the positioning technique used and the characteristics (density, layout, etc.) of the deployed infrastructure of Bluetooth devices and anchors (beacons) [23] [31]. Examples of Bluetooth positioning systems include ZONITH Indoor Positioning System [32], TOPAZ [33] and Apple iBeacon [34].

2.2.6 Others

ZigBee: An emerging wireless communication technology based on IEEE 802.15.4 standard for PAN/LAN intended for applications which do not need significant data throughput, but require low-power consumption is the ZigBee technology. It uses 915 MHz band in the USA and Australia, 868 MHz band in Europe, and 2.4 GHz in other regions [35]. It is used for long distance communication between devices in wireless mesh network and when compared with WiFi standards, it has low cost, low data transfer rate and short latency time. LQI (Link Quality Indication) in IEEE 802.15.4 standard is defined to indicate the quality of the link and can be used to derive RSS (Received Signal Strength) integrating manufactured chips (CC2430/CC2431) [35] to get the RSSI making the implementation of the system easier.

ZigBee based localization uses proximity and TOA techniques based on distances computed from RSSI of the surrounding ZigBee nodes usually giving the accuracy of 1-10 m [36]. A flexible indoor Wireless Sensor Networks localization system using ZigBee RSSI level measurements is described in Larranaga et al. [37]. Their localization system consists of two steps: i) calibration phase, which is performed whenever an unknown ZigBee node needs to be located and ii) actual positioning phase with 3m average accuracy.

Ultra-Wide Band (UWB): unlike other technology, UWB uses a sub-nanosecond radio pulse to transmit data in a wide range of bandwidth. Its transmission can be regarded as background noise to other wireless technologies, hence in theory, it can use any spectrum without interference with other users. It uses small transmission power- 41.4dBm/MHz (meaning the power consumption is low) and it is immune to multi-path problems in theory [38].

This technology has emerged providing better positioning accuracy than Wi-Fi. It is appropriate for high-accuracy real-time localization using TOA, TDOA, AOA and fingerprinting. Its signals are less sensitive to multipath distortion and environment than conventional RF-based localization systems; as a result it gives higher accuracy. With bandwidths of at least 500MHz and high time resolution in the order of nanoseconds, localization accuracy at a cm-level can be obtained [39]. A MUSIC-based method [40] can achieve high localization accuracy, in the order of 1 cm, by using spatially distributed antennas which transmit the same UWB impulse sequence for self-localization of IR UWB nodes [41] but its usage for localization is limited because of the high cost of UWB equipment and infrastructure deployment [42]. Currently implemented and deployed UWB positioning systems include the Ubisense system [43] and PulsON [44] system developed by Time Domain.

Radio Frequency Identification (RFID): RFID systems are designed for the reader to detect a tag in its vicinity and retrieve the data stored in that reader. Therefore, the absolute location of the tag is not known but the RFID system is aware that a tag is positioned in the vicinity of a reader. Its accuracy depends on the type of the system used, either active or passive RFID [45]. In addition to the proximity technique which estimates the location of the tag according to a reader location, there are many other methods for performing accurate positioning using active RFID technology; for instance: AOA, TDOA and RSSI achieving accuracy in the range of 1-5 m depending on the density of tag deployed and RFID reading ranges [46]. Examples of such works include SpotON [47] and LANDMARC [48].

Infrared (IR): Dedicated localization techniques, usually based on infrared and ultrasound technologies, give a high degree of accuracy but require expensive equipment; are limited to a small scale and usually have high installation and maintenance costs [36]. In infrared (IR)-based systems, each tracked person wears a small infrared device which emits a unique pulse signal representing its unique identifier. This emitted signal is detected by at least one particular IR sensor in the vicinity and sends it to a location server. Then, a location server estimates IR device location using aggregated data obtained from fixed IR sensors deployed within the indoor environment.

The first IR positioning system, the Active Badge System [4], for example, gives location information at the room or smaller level depending on deployed IR sensor infrastructure. On the other hand, using an ultrasound time-of-flight lateration, ultrasound-based techniques give more accurate location than infrared signals. It gives accuracy in the range of 1 cm – 1m [49]. The well-known examples include Active Bat System and the

Cricket indoor location system [36]. However, the existence of Non Line of Sight (NLoS) circumstances and multipath propagation in indoor environments are the main problems in the development of reliable ultrasound-based indoor localization systems.

Inertial sensors: The dead reckoning-based localization techniques, also called Inertial Navigation Systems (INS), can be used for both indoor and outdoor positioning. For example, in indoor environments, such systems use accelerometers to obtain human velocity rate information through step detection and step length estimation. On the other hand, digital compass measurements are used for direction and angular rate information. Such systems have been implemented in diverse indoor tracking domains; for instance, pedestrian navigation [50] and mobile robot localization [51] [52]. An Android smartphone application performing location tracking in home and office environments using the integrated motion sensors of the smartphone and an optional foot-mounted inertial measurement unit for personal localization and tracking is presented by Hardegger et al. [53] in their work Smart ActionSLAM. However, in such methods, even very small errors in the rate information provided by inertial sensors cause an unbounded accumulative growth in the error of the integrated measurements, usually referred as the “drift error”.

2.2.7 Summary of Geo-localization Technologies in Wireless Networking

Various localization technologies, techniques of localization, coverage of signal, and battery consumption in hours and error in metres when used for localization from the above discussed different literatures are summarized in Table 2-2. The ‘√’ mark for attributes means ‘yes’. For example, is GPS available in Smart Phone? Yes it is available. On the other hand ‘-’ mark in the table refers to ‘not yet known’.

Table 2-2: A Review of Geo-localization Technologies and Systems in Wireless Networks

Geo-localization Technologies	Methods	Techniques	Infra-Structure Commonly Available	Radio Emitting Power (mW)	Coverage (m)	Battery Lifetime (h)	Location Error
GPS	TDOA	Trilateration	√	Vary	Outdoor	9 h (low)	10 m
A-GPS	TDOA	Trilateration	√	Vary	Indoor/ Outdoor	low	10 m
WiFi	Cell-ID TOA TDOA AOA RSSI	Proximity Trilateration Angulation Scene anal.	√	32–200	50	40 h (high)	1-100 m
Cellular Network	Cell-ID TOA AOA RSSI	Proximity Trilateration Angulation	√	100–2000	In kilometres	60 h (high)	400 m
Bluetooth	Cell-ID TOA RSSI	Proximity Trilateration Scene anal.	√	10	10	60 h (high)	1-5 m
802.15.4 (Wireless HART*, ZigBee)	Cell-ID RSSI	Proximity Trilateration	-	1	35-75	medium	1-10 m
Passive RFID	Cell-ID RSSI	Scene anal.	-	-	-	high	1-5 m
Active RFID	Cell-ID RSSI	Proximity Trilateration Scene anal.	-	-	-	high	1-5 m
Inertial Sensors	Dead Reckoning	Dead Reckoning	-	-	-	high	1-10 m
Camera Tracking	Computer vision	Image Analysis	-	-	-	low	1cm-1m
UWB	RSSI ToA, AoA TDOA	Trilateration Angulation	-	-	-	low	1cm-1m
Ultrasound	TOA TDOA	Trilateration	-	-	-	low	1cm-1m
Infrared	Cell-ID TOA	Proximity Trilateration	-	-	-	medium	1cm-5m

In recent years, Location Based Services (LBSs) are gaining a lot more influence in daily life. To obtain the needed current location of devices various localization techniques are deployed such as Global Positioning Systems (GPS), Assisted GPS (A-GPS), Wireless Fidelity (WiFi)-based localization, cellular network based localization technology, Bluetooth based localization and many more. Most modern devices have an inbuilt GPS receiver with high accuracy making GPS the predominantly used localization technology. The use of GPS to acquire location information also comes at a price of very high energy consumption, as shown in Figure 2-3 [3].

This is serious problem because wireless devices have weak batteries. The extra energy load caused by localization could use up all available energy from the battery after less than nine hours in smart cell phones [3].

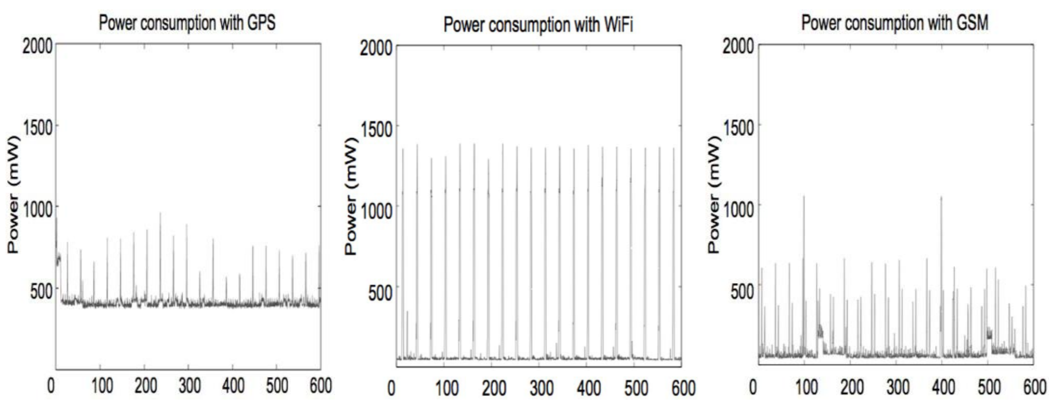


Figure 2-3: Energy consumption of GPS, WiFi and GSM for position fixes every 30 seconds

With the introduction of more advanced services like geo-tagging and proactive location based services, positioning updates are required much more frequently, sometimes even continuously over certain periods of time. This puts extra strain on battery life. To respond to this problem, several ways to increase the energy-efficiency of wireless device localization technologies have been proposed. In this review of related literature, such improvement approaches towards energy efficiency of wireless device localization are classified into four different categories, which are examined closely.

2.3 Methods of Improving the Energy-Efficiency of Wireless Device Geo-localization

Although GPS localization technology offers a high degree of accuracy in positioning, it is enormously energy hungry meaning wireless devices run out of power after about 9 hours if used continuously [3]. However, the trend in battery capacity, technology development shows improvement not in line with need. An alternative to increasing the energy pool, decreasing the energy use of localization technologies could be promising options. For instance, the use of assisted-GPS instead of plain GPS already improves energy consumption.

Recently, many approaches to decrease the energy consumption of wireless devices geo-localization have emerged. In general, most of these methods can be classified into the following categories:

1) Substitution of GPS with other technologies

2) Introducing middle-ware

3) Smart switching of localization technologies

4) Smart reduction of GPS operations

5) Co-operative localization

Except for the last one, all the above methods are employed on the optimization of battery life of individual wireless devices with no cooperation with other wireless devices in the network. Each of these five approaches of reducing energy consumption during wireless device localization is discussed in subsequent subsections:

2.3.1 Substitution of GPS with other Geo-localization Technologies

This approach tries to overcome the energy issues of GPS by substituting it completely with other technologies like GSM, WiFi, Blutoth and the like. These days these technologies are under exploitation for energy efficient localization technologies. Depending on what kind of wireless device is used, WiFi sensing or GSM connections are usually active most of the time while the phone is active, which lowers the energy needed for positioning. Usually, while WiFi based approaches are often used for indoor tracking, GSM based approaches are mainly useful outdoors [19].

LaMarca et. al. proposed a system called Place Lab [54] which can be used both outdoors and indoors. As shown in Figure 2-4 [54], their main goal was to prove alternate technologies to offer ubiquitous localization with low cost. Their system uses already existing infrastructure of radio frequency anchors (beacons): WiFi access points, GSM cell towers and Bluetooth devices. In this method the wireless device only listens to certain information the beacon transmits, e.g. the unique identifier of a GSM tower without requiring active connection. For wireless device positioning, Place Lab uses a large database compiled and maintained by large communities which contains more than 2.2 million beacons of which more than 40,000 is GSM tower locations.

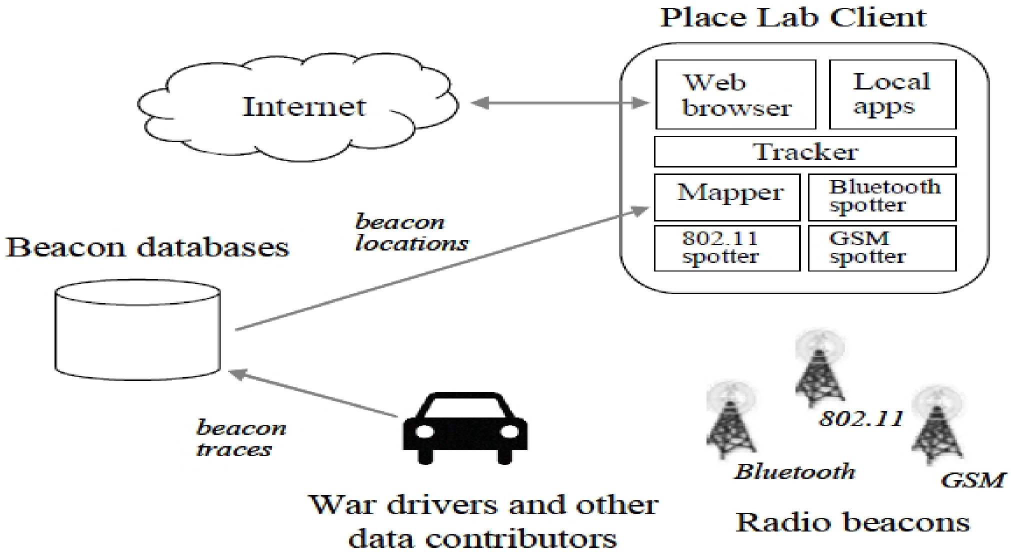


Figure 2-4: The Place Lab system architecture

2.3.2 Introducing Middle-ware

This method uses middle-ware as an interface between the actual battery and other imbedded sensors in the wireless device. Energy efficient continuous sensing on mobile phones can be achieved by modifying the current architecture of mobile phones by designing to support continuous human centric sensing at a low power overhead. This can be achieved by design choices and integrating Little Rock into an actual phone [55]. Little Rock is middle-ware or interface between the actual battery and other imbedded sensors, hence, not actively needed sensors in the wireless device can stay turned off saving battery.

The middle-ware can be a context aware middle-ware. For example, ACE (Acquisition Context Engine) is a context aware middle-ware which infers users current context; like, driving or at home [56]

2.3.3 Smart Switching of Geo-localization Technologies

In smart switching of geo-localization technologies, the modelling of localization technology's energy consumption and accuracy is modelled in two different ways: i) static model- assigns static (never change) values ii) dynamic model- updates and changes depending on the results of performed positioning processes [57] [58]. Independently of which model is used, the resulting system intelligently chooses which localization technology should be used. The following sub-sections discuss the two techniques:

2.3.3.1 Hierarchical Positioning Algorithm

Figure 2-5 represents example architecture of the hierarchical positioning algorithm system [59]. In this figure, the wireless device moves along the dashed line. On its way it comes across GSM cell towers which are used to check whether their coverage area overlaps the target zone. As long as that is not the case, no higher accuracy and less energy-efficient sensor has to be activated. In this case, the next higher accuracy providing technology has to be turned on when the device enters the coverage area of the second cell tower. At this moment, WiFi is turned on until the device enters the next higher coverage area that overlaps with the target zone, in this case, the WLAN of access point AA: 03. This invokes again a shift to the higher accuracy technology GPS, which is used until the device actually enters the target zone. Since Location Based Services (LBSs) require different minimal location accuracy values, this approach allows the power hungry technologies to stay turned off for as long as possible.

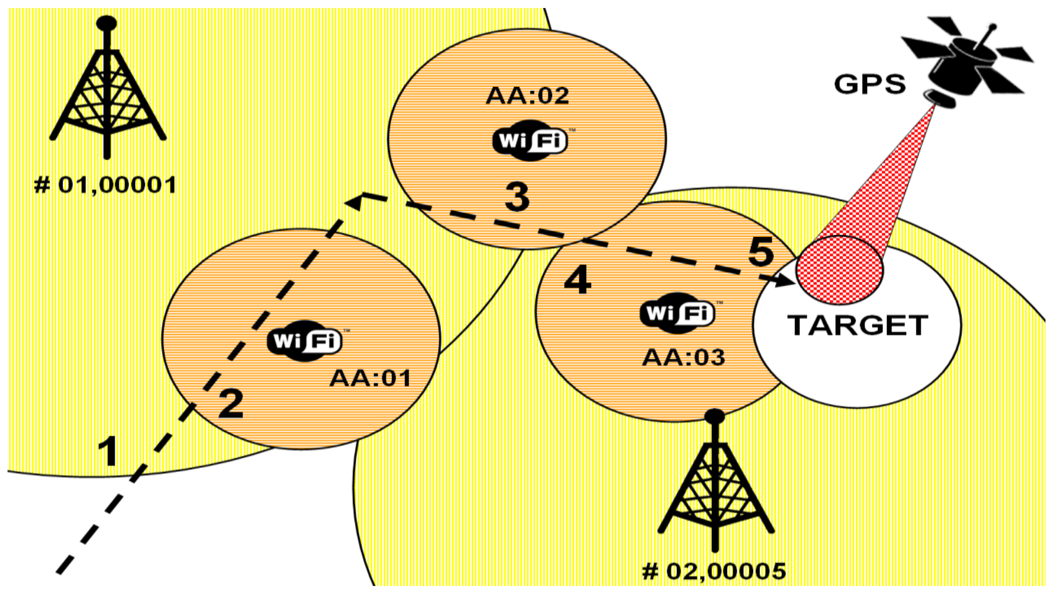


Figure 2-5: Overview of the hierarchical positioning algorithm functionality

2.3.3.2 Dynamic Location-Based Sensor Selection

Lin et. al. follow a basically similar but slightly different approach [57]. They argue the precision required by location based services varies and correlates with each specific location of the wireless device. Consequently, they propose a system which adaptively estimates the best localization technology (sensor) for the current location and accuracy requirement. Unlike a hierarchical algorithm which uses static model, they propose a complex and dynamic system, called a-Loc which adjusts to precision requirements of LBAs by using dynamic sensor error and static sensor energy consumption models, as shown in Figure 2-6 [57].

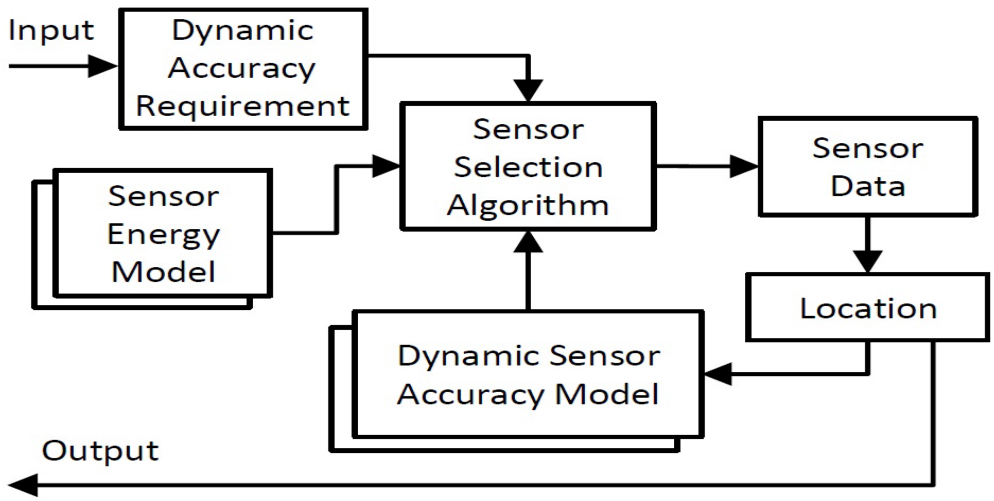


Figure 2-6: System design of a-Loc

2.3.4 Smart Reduction of Necessary GPS Operations

This method uses GPS in most situations because it gives the maximum accuracy of all commonly used localization technologies. However, to be energy-efficient, the number of performed GPS operations is reduced and optimized, so the minimum amount of energy is used while achieving an accuracy required by LBAs. This approach can be further sub-grouped into two: Prediction Based Reduction and Movement Based Reduction.

2.3.4.1 Prediction Based Reduction

Constandache et. al. argue that the energy budget available to localization is only a fraction of the battery because other parts of the wireless device also need energy [3]. Hence, for continuous positioning, the available energy is not enough. As a result, the position of the device should only be estimated intervals because no extra localization operation can be performed. Having this central idea, they propose a device centric localization system, EnLoc [3] which tries to minimize the accuracy deviation or average localization error for a given energy budget. Consequently, to exploit typical movement and habitation practices of users; the system uses prediction based heuristics to predict the location and movement of wireless devices, so reducing the number of location readings needed. Using this prediction based approach; it is possible to lower error rates throughout the time localization is active because it uses GPS. This method exploits movement patterns of the individual user and habitual whereabouts, many studies have showed such data is reliable [60]. For the implementation of their approach, Constandache et. al. collected data on the University of Illinois campus. This data is used to describe the habitual movements of a user and compiled on tree based information representations as shown in Figure 2-7.

To decrease the needed localization operations, position estimations are performed if the user has started to move to a new habitual activity area, depending on the gathered information. A linear prediction in combination with probability maps of cities is used to check whether the user's path deviates. If it does deviate, GPS readings have to be performed to monitor his position. In such cases there is no energy consumption reduction.

EnLoc works mainly with GPS; however, it also supports WiFi and GSM based localization. Among the supported localization sensors, a sensor type that minimizes the error for the whole time period is chosen based on predictions. This helps a sensor with medium accuracy, but low energy consumption to conduct location readings more frequently than others.

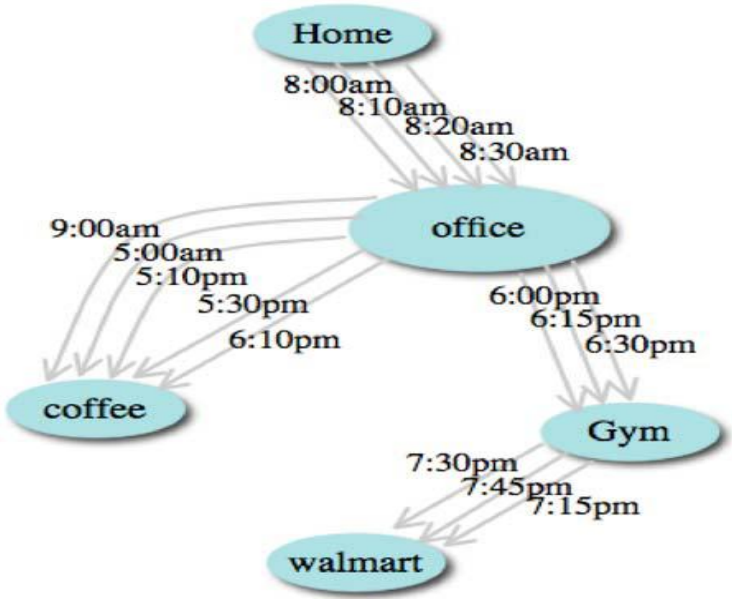


Figure 2-7: User specific habitual locations throughout a day

2.3.4.2 Movement Monitoring Based Reduction

Kjaergaard et. Al. developed a system, EnTracked [61], which estimates wireless device location using low energy consumption. The estimation of position depends on whether the wireless device is moving or not. If it moves, its speed is learned and used to estimate the point in time when the next location reading should be executed because the user moves for a certain distance or time, until the last known position will become too inaccurate for location based services. Thus, this method reduces necessary GPS localization operations by monitoring and estimating the movement of wireless devices.

EnTracked contains a detailed model for energy consumption of various devices, like the GPS receiver, accelerometer and GSM radio. It also takes into consideration the time taken to power on and off those devices and a delay model is used to provide the time it takes for GPS to obtain a new position fix. Figure 2-8 illustrates their works basic workflow and system logic. After the first estimation, the system waits for the device to move to make the next position estimation, up to that it uses the first estimation. The movement checking is accomplished using an accelerometer. If movement is detected, the velocity of the device is measured for the next position estimation. To achieve this, the inbuilt speed determination function of the GPS receiver can be used.

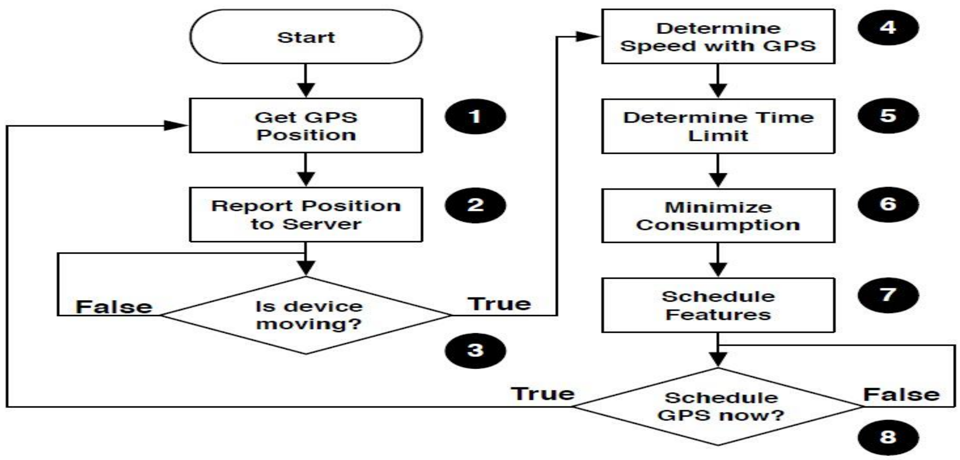


Figure 2-8: System logic of EnTracked

EnTracked was tested by emulation and real world tests. While emulation resulted in 43-56% saving, the real world tests which were conducted during the phase of one week in residential and urban districts achieved energy-efficiency 62.3-69.7% energy savings surpassing the emulated values.

2.3.5 Co-operative Localization

The fourth type of approach in decreasing energy during wireless device localization is co-operative localization. The concept of co-operation in networks is fairly new: it relies on direct communication between agents rather than through a fixed infrastructure [62] [63] 64]. Co-operation has been successfully applied to wireless peer-to-peer communication, leading to standards such as Bluetooth [65] and Zigbee [66] and is expected to be applied to cellular systems over the next few years [67] [68].

As this method is the focus of this dissertation, the detail of range-free co-operative localization methods and algorithms are discussed in Chapter 3 sub-section 3.2, where anchor nodes share their position to help unknown nodes to locate themselves.

2.3.6 Summary on Methods of Improving the Energy-Efficiency of Geo-localization in Wireless Network

Wireless device localization energy improvements are necessary developments in order to ensure the long-term use of location based services. Research is tackling this issue in many different ways which can be classified into five different categories. The first tries to substitute the commonly used and energy hungry GPS localization with more energy efficient technologies, like WiFi and GSM; however, those localization technologies have much lesser precision than GPS resulting in a serious drawback to some location based services that require good precision. As presented in work [8] and [54], the location precision can be enhanced by using additional positioning techniques such as received signal strength based estimations. However, to be able to use them constantly, extensive preparatory work in the form of wardriving and signal strength measurements have to be conducted which is again followed by tedious maintenance of the collected data to make it up to date.

The second method uses middleware to reduce power usage by introducing interface between the actual physical battery and localization sensors. The middleware turns on the sensors only when there is a need for them otherwise they remain turned off – so saving power. The third uses multiple localization technologies and smartly switches between them based on the precision currently demanded. At any time, only one kind of the available sensors is active by default, this is the one consuming the least energy but still giving the demanded accuracy. A more precise sensor is selected only if more precision than the current sensor can provide is required. Apart from performing energy measurements for each localization strategy and consequentially constructing a model, no extensive preparatory work is needed.

Approaches of the fourth classification apply smart prediction and estimation algorithms based on user movement monitoring. Even though only GPS is used, very encouraging energy savings can be achieved. However, these approaches deeply penetrate user privacy, particularly when compiling user based habitual maps.

The final classification which is co-operative localization look promising in saving large amounts of energy wasted for wireless node localization on the cost of poor accuracy. Nevertheless, the improvement of energy savings should not be to the cost of accuracy, rather this method has to optimize both accuracy and cost (battery). Hence, this research investigates this approach to reduce the impact of crowd of wireless device geo-localization on battery life while optimizing accuracy.

Deliberately left blank

CHAPTER 3 RELATED WORK: GEO-LOCALIZATION METHODS AND ALGORITHMS

In recent years, rapidly growing location based applications have attracted a lot of research efforts on geo-localization of nodes in wireless networks [69] [70] [71] [72] [73]. GPS, which is widely used localization method with high accuracy, is not a sound solution for wireless network applications because its high energy consumption is against the energy scarce nature of wireless nodes' battery [69] [72] [73] [74] [75] [76].

Consequently, researchers are investigating innovative ideas to realize energy efficient, practical, flexible and robust localization in wireless networks. Generally, proposed localization methods can be categorized into two classes: i) range-based and ii) range-free. The main difference in the heart of this classification lies whether ranging information is required or not in the localization process.

Both range-based and range-free geo-localization methods and algorithms are discussed in detail in the following sub-sections, respectively. Figure 3-1 shows the general classification view of geo-localization methods in wireless networks.

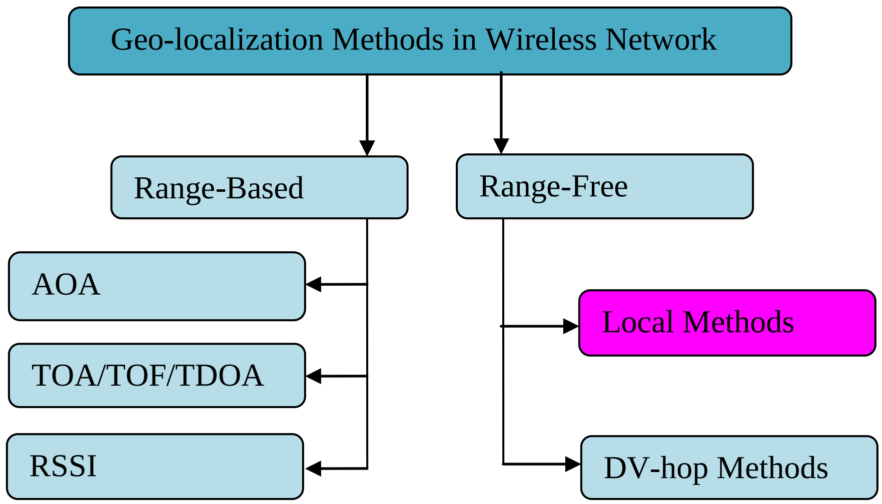


Figure 3-1: Classification of Geo-localization Methods in Wireless Networks

3.1 Range-Based Geo-localization Methods

As its name indicates, range-based localization methods depend on accurate ranging results among wireless nodes. To do these ranging results, the wireless node needs dedicated ranging infrastructure to measure point-to-point distance, angle, time or (relative) velocity. Once obtaining ranging results, the location of wireless nodes can be estimated through geographical computations like trilateration (also can be multilateration) [69] [77] [78] [79] or triangulation (as presented in the following subsections) [8] [80] [81] [82].

In Trilateration, as shown in Figure 3-2, the wireless node N_x is the node which we want to localize. In the figure, the three nodes (A_1, A_2, and A_3) are anchor (or beacon) nodes and know their locations. Here, it is assumed the wireless node N_x has communicated and knows the location (x_i, y_i) of each anchor A_i and the distance d_i between N_x and A_i.

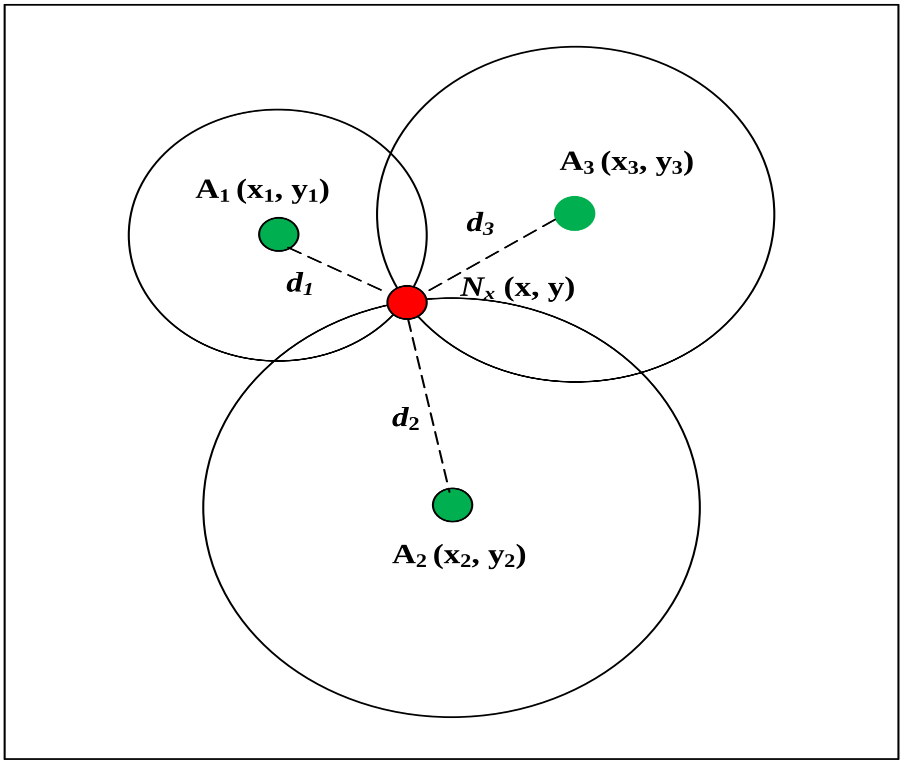


Figure 3-2: Trilateration / Maltilateration

Then, N_x can estimate its location using Equation (3.1) which is usually referred to as trilateration. In the equation the location of N_x is unknown and labelled as (x, y).

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = {d_1}^2 \\ (x - x_2)^2 + (y - y_2)^2 = {d_2}^2 \\ (x - x_3)^2 + (y - y_3)^2 = {d_3}^2 \end{cases} \tag{3.1}$$

The following subsections explain range-based localization methods from the viewpoint of the three types of fundamental ranging information: (i) angle of arrival, (ii) time of arrival/flight, and (iii): received signal strength indicator

3.1.1 Angle of Arrival

AOA-based methods first measure the angle at which a signal arrives at an anchor node. Finally using triangulation an unknown node estimates its position [83] [84]. The computation is quite simple; however, AOA techniques require special antenna and may not perform well due to omni-directional multipath reflections. Moreover, measuring signals accurately can be difficult if an unknown node is surrounded by scattering objects.

Localization techniques (RSSI, TOA and TDOA) use distance estimates for localization; however, Angle of Arrival (AOA) localization method uses the angle information for localization. Localization with AOA requires two angle measurements as shown in Figure 3-3; for example, as the signal sent from the unknown node N_x arrives at anchor nodes A_1 and A_2, the antenna array of A_1 and A_2 can detect the arriving signal's AOA as α and β, respectively. Finally, the two anchor nodes send the measured angle information α and β as well as their positions (x_1, y_1) and (x_2, y_2) to an unknown node N_x. Now, N_x can compute the distance between anchors, denoted as d in the figure using the positions of anchor nodes.

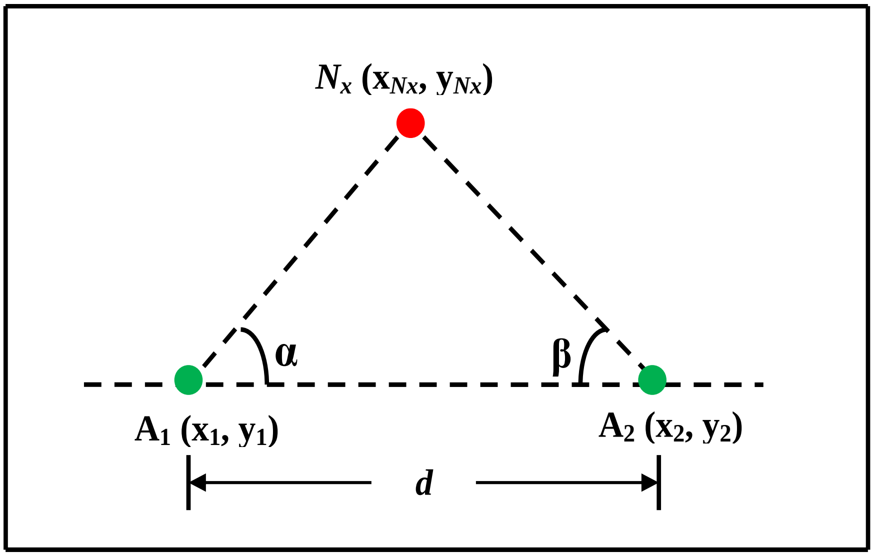


Figure 3-3: Angle of Arrival (AOA) Localization

At last, N_x computes its location (x_{Nx}, y_{Nx}) using the triangulation method given in Equation (3.2) [83] [84].

$$\begin{cases} x_{N_x} = x_1 + \dfrac{d \times \sin\alpha \times \sin\beta}{\sin(\alpha+\beta)} \\ y_{N_x} = \dfrac{d \times \cos\alpha \times \sin\beta}{\sin(\alpha+\beta)} \end{cases} \tag{3.2}$$

Like other range-based methods AOA also has shortcomings: i) it uses antenna array for detecting the angles which means it needs additional dedicated hardware for angle measurement, an energy constraint; ii) the AOA measurement precision is affected by background noise and/or multi-path reflections which could result in significant errors in angle measurement which in turn affect position estimation.

3.1.2 Time of Arrival

TOA calculates distances based on transmission times and speeds of signal [85]. The most popular TOA-based localization system is GPS. GPS receiver computes node position based on signal propagation time by precisely synchronizing with a satellite's clock.

The distance calculation using Time-of-Arrival (TOA) is usually more accurate than using RSSI, particularly in the environment with many obstructions, for example, in an indoor office space [85]. Time-of- Arrival (TOA) based direct distance measurement method is shown in Figure 3-4, where t_a is the signal transmission instant from the anchor node, while t_u is the signal reception instant at the unknown node. Assuming the nodes are synchronized (share a common clock), the TOA is computed as $|t_a - t_u|$. Given the speed of signal propagation is equal to *c (speed of light)*, we can compute the distance between the nodes as $c \times |t_a - t_u|$.

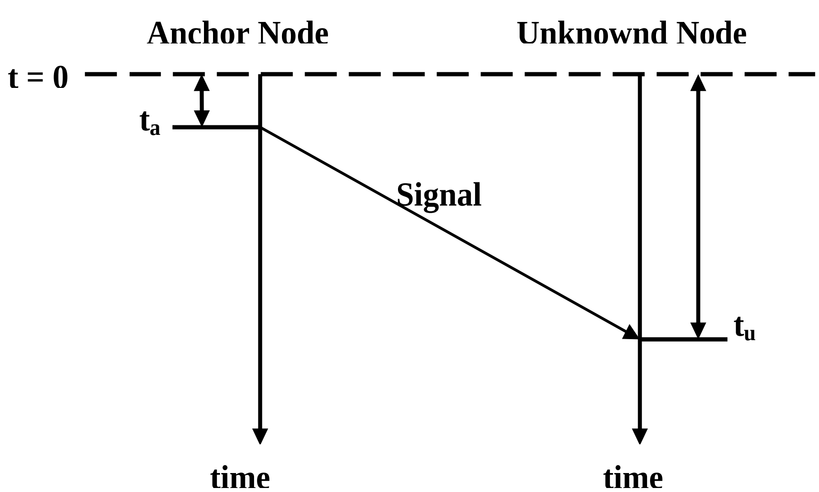


Figure 3-4: Time-of- Arrival (TOA) Localization in a synchronized

TOA measurement; however, demands perfect-synchronization (which is not practical in the real networks) between the two nodes. The IEEE standard 802.15.4a supports a two-way ranging method to minimize the problem of synchronization, as depicted in Figure 3-5.

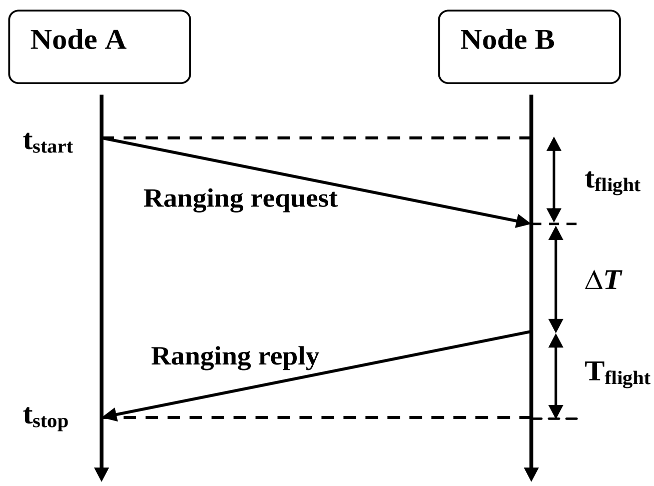


Figure 3-5: Two-Way Ranging

This two-way ranging technique necessitates the two nodes to exchange at least two packets. For example, in the figure, the node A starts a ranging measurement by sending a ranging packet to node B at time t_{start}. Then, node B receives the packet from A, and replies with a second ranging packet, transmitted after a delay ΔT. The packet is received by the device A at time t_{stop}. Now, using Equation (3.3), we can compute the propagation time from Node A to Node B, denoted as t_{flight}.

$$t_{flight} = \frac{t_{stop} - t_{start} - \Delta T}{2} \tag{3.3}$$

It is obvious the two-way ranging method reduces the influence of node synchronization; however, the localization techniques based on TOA still have the problem of node timing resolution. To achieve a precise measurement of TOA, wireless nodes need to have ultra-resolution timing ability necessitating to be equipped with specially designed radio chips [86] [87] or high speed clocks and processors [88] [89] [90] [91]. Ultra Wide Band (UWB) [92] and Direct Sequence Spread Spectrum (DSSS) are the two widely used TOA measurements in geo-localization technologies [93]. Both UWB and DSSS are wide-band or ultra-wideband signals with enhanced time domain resolution. Among TOA measurement technologies, the DSSS signal has been used in ranging systems for several years (e.g., the GPS). To make such systems robust to noise and interference, a signal coded by a pseudo-noise (PN) sequence is transmitted from the sender. At the receiver side the received signal needs to be decoded with a local PN sequence [61]. Finally, the distance between the transmitter and the receiver is computed using the arrival time of the received signal. In DSSS ranging systems, the resolution of TOA estimation is mainly determined by the signal bandwidth; for instance, if a bandwidth of 100 MHz is used, distance estimation errors are usually less than 3 metres under ideal LOS (line of sight) conditions [94], but the system requires high speed clocks, for example, this necessitates the PN generator to have a time resolution of at least 1/100MHz=10 ns.

One of the key factors that affect TOA estimation is signal bandwidth, the wider the bandwidth, the higher the range measurement precision. Among the existing range-based localization technologies Ultra Wide Band (UWB) systems (usually with bandwidths more than 1 GHz) have shown a promising future, particularly for complex environments like indoor localizations [89]. Moreover, multi-path fading has less effect on UWB signals. Incorporating the UWB systems on wireless nodes; however, is not easy because UWB systems require sophisticated additional hardware to provide swift sampling and accurate timing; as a result, integrating it in wireless nodes is quite challenging and costly. Another variant of a TOA system is Time-Difference-of-Arrival (TDoA). This method requires base stations to transmit both RF and ultrasound signals simultaneously. Here, the RF signal is used for synchronization reason. An unknown node first computes the difference of the arrival times between the two signals, and then determines the distance to the base station. At last, using multilateration on the collected range estimates location on the unknown node is estimated.

3.1.3 Received Signal Strength Indicator

RSSI calculates distance depending on transmitted and received power levels of radio propagation model. RSSI is usually used with RF signals. However, the measurement estimation can be inaccurate due to multipath fading effect in outdoor environments [95] [96] [97].

Radio Signal Strength Indicator (RSSI) is considered as the most popular modality for range estimation in wireless networks because almost every node in the wireless network has the ability to analyze the strength of a received message, [95]. As a result, RSSI information can be obtained with no dedicated additional hardware for ranging when compared with other ranging methods [96] [97]. This makes it energy efficient and cost effective. Consequently, in order to effectively utilize RSSI for localization, two types of methods have been studied: (i) direct calculation of distance from RSSI; (ii) RSSI fingerprinting. The following subsections discusse these two types of methods:

3.1.3.1 Direct Calculation of Distance

As the distance from the emission source increases, the intensity of an emitted signal decreases. This decrease relative to the original intensity is the attenuation [98]. This signal attenuation with respect to distance occurs in a polynomial manner. In the most ideal circumstances (free space), signal power decay or attenuation is proportional to d^2, where d denotes the distance between the transmitter and the receiver; such an effect is also called free space loss [99]. By having a function which correlates attenuation and distance, distance between the transmitter and the receiver can be estimated from the received signal strength by the receiver. One of the commonly used radio propagation models is the log-distance path loss model (without multipath effects) in Equation (3.4)

$$RSSI(d)[dBm] = RSSI_{ref} - 10n\log_{10}\left(\frac{d}{d_{ref}}\right) \qquad (3.4)$$

where, $RSSI$, which is a logarithmic measurement of signal strength is measured in dBm and d is the distance between emitter and receiver; $RSSI_{\mathrm{ref}}$ is the signal strength value at reference distance d_{ref} and n is the attenuation constant (rate at which the signal decays). Generally, n is computed from empirical data which is usually around 2 in a free-space environment but increases if the environment is more complex (walls, large metallic objects, etc.). In complex environments with many obstructions, for example, in an indoor office space, the value of n ranges between 3 to 6 [100]. Derived from Equation (3.4), a known model for computing the distance d between emitter and receiver is using Equation (3.5), where $RSSI_{\mathrm{ref}}$ is measured at $d_{\mathrm{ref}} = 1$ m. Finally, having the distance, the locations of nodes is computed using trilateration or multilateration method.

$$d = 10^{\frac{RSSI_{ref} - RSSI}{10n}} \qquad (3.5)$$

The RSSI value in addition to the distance also depends on the power supply, antenna orientation, the movement of the emitter and the receiver, and complexity of the environment [101] making it unpredictable [102] [103] [104] [105] [106] because the reflecting and attenuating caused by objects in the environment can have much larger effect on RSSI than distance.

As a result, obtaining the accurate distance from RSSI without a detailed model of the physical environment is hard [95] [96] [107] [97]. On localization experiments conducted using RSSI in open space, results show the location error is mainly between 3.7m and 4.5m with the communication range about 25m [108] which would have been different if it is complex environment.

3.1.3.2 RSSI Fingerprinting

As direct distance estimation from RSSI is found to be inaccurate in the indoor scenario, due to that a different method called RSSI fingerprinting has been proposed [109] [110]. Unlike the direct distance measurement method, RSSI fingerprinting method involves two steps: in the first step (which is referred as learning step), anchor nodes record the power level of frames sent periodically by the wireless nodes. The current position and orientation of the wireless nodes is contained in these frames. This step is also known as the offline phase because it is usually performed before the activation of the localization service provided by the network. In the offline step, the anchor nodes map each frame with the measured RSSI and the time of reception. Here, since all the nodes are synchronized, the time values are valid throughout the network. Once we finish the offline phase, we build a database having each wireless node's location and orientation (north, south, east, and west) and the RSSI measurements taken by each anchor nodes.

The second step is referred as online step. In this step the physical location of wireless nodes is estimated by the beacon nodes by matching the empirical values contained in the database and the online received RSSI measurements.

However, compared to the first method, the fingerprinting method has the following disadvantages: its offline setup is expensive and its high data volume takes high memory and online mapping time. Moreover, this method is not so flexible because any change in the configuration, for instance, the adding of a new anchor node, will involve building a new database. This makes it complex and tedious work in the dynamic environment where anchor nodes can change in their number and original location.

3.1.4 Summary of Range-Based Geo-localization Methods

Among range-based methods, RSSI methods have the lowest cost because the RSSI information can be obtained without any additional dedicated hardware, but they usually give lower accuracy since the RSSI signal is easily affected by the environment. From the RSSI methods, the fingerprinting method gives better precision than the direct distance calculation method, but it has the following serious short comings: i) it demands large memory for the database to store offline RSSI measurements and ii) it is not flexible; as a result, it is not a sound solution in a dynamic world.

Even though TOA and AOA based methods can achieve better precision than RSSI, they all look for additional hardware (for example, high-speed clocks to support ultra-high resolution timing for TOA based methods and antenna array to effectively detect the angles for AOA based methods) putting additional pressure on the scarce battery of wireless node.

Table 3-1 summarizes the typical range-based localization methods discussed above.

Table 3-1: Comparison of Range- Based Localization Algorithms

Range-Based Methods		**Strength**	**Weakness**
RSSI	Direct Calculation of RSSI	-No additional dedicated ranging hardware -Scalable -Low overhead cost	-Affected by signal attenuation -Low accuracy
	Fingerprinting	-No additional dedicated hardware -Better accuracy	-Non-scalable -Non-flexible -Memory cost
TOA	Direct Calculation of TOA	-Better accuracy than RSSI -Low overhead cost	-Strict Synchronization -Ultra-high timing requirement -Expensive hardware
	Two-way Ranging	-Better accuracy than RSSI -No rigid synchronization	-Ultra-high timing requirement -Expensive additional dedicated ranging hardware
AOA		- Low timing/ synchronization requirement -Better accuracy than RSSI	-Expensive additional dedicated ranging hardware -Hardware constraints -Affected by multipath fading and noise

3.2 Range-Free Geo-localization Methods

Unlike range-based schemes, the range-free schemes do not use range measurement techniques. In order to estimate the position of unknown nodes, these schemes depend on node connectivity information, i.e., who is within the visibility range of whom. For instance, if one node is within the visibility range of another node, the distance between these two nodes can be estimated as one hop, and these two nodes are referred as neighbour nods.

Generally, there are two types of nodes in this method. The nodes which are aware of their positions are referred as anchor (or beacon) nodes, while other nodes which are unaware of their location are called unknown nodes. This method usually assumes unknown nodes can hear from multiple anchor nodes which are often considered fixed. Unknown nodes first gather the connectivity information as well as the positions of anchor nodes, and then calculate their own locations.

The main advantage of the range-free schemes is they are appropriate to be implemented on low-cost wireless/sensor networks since no battery hungry ranging information is needed. Moreover, they are robust- the connectivity information between nodes is not easily affected by the environment. Hence, the focus of this work is on these methods.

Generally, range-free geo-localization algorithms can be classified into two categories: i) local methods and ii) DV-hop (Distance Vector-hop) also called Hop Counting methods [12]. In the local methods, an unknown node collects the location information of its neighbour anchor nodes to estimate its own location. On the other hand, in hop-counting method, each unknown node looks for anchor nodes in the network to give their estimated hop sizes and then attempts to get the smallest hop count to these anchor nodes. Next, every unknown node estimates its distance to each of these anchor nodes using the hop count. At last, the unknown nodes estimate their location using trilateration basing on the estimated distances to three appropriate anchor nodes. The following sub-sections present them in detail.

3.2.1 Local Methods

Local method localization algorithms include: Centroid [76] [111], Original Convex Position Estimation (Original CPE) [112], Improved Convex Position Estimation (Improved CPE) [113] [114], Approximate Point-In-Triangulation (APIT) [115], and Mid-perpendicular [116] [117]. Each of these algorithms is presented in detail in the following sub-sections:

3.1.1.1 Centroid Algorithm

Bulusu et al. proposed a Centroid algorithm which is probably the earliest and simplest range-free scheme in which each unknown node estimates its position by computing the centroid of all the anchor nodes it hears [76]. After that, the unknown node can be an anchor node and broadcast packets to the vicinity. This method of localization is very simple, economic and simple to implement. However, localization accuracy is vulnerable to the density and deployment of anchor nodes and location accuracy it gives is poor when compared with other related works.

The authors selected a simple radio propagation model appropriate for the outdoor environment. In the model, perfect spherical radio propagation and identical transmission radio range for all anchor nodes is assumed. For example, in Figure 3-6, there are *n* anchor nodes at known locations: A (x_1, y_1), A (x_2, y_2) … A (x_n, y_n) with the same communication radio range represented as R. These anchor nodes are the neighbour anchor nodes of the unknown node N_x. The shaded part in the figure shows the overlap of these anchor nodes transmission area. This method attempts to locate the unknown node N_x inside this overlap.

The authors assume that i) each anchor nodes periodically (period=T) broadcasts its location beacon signal and ii) all anchor nodes are well synchronized to avoid collisions during the transmissions. We use their definition of terms listed below to ease the explanation on the algorithm:

R: Node transmission/communication range

T: Time interval between two beacon signals transmitted by anchor nodes

t: Unknown node N_x uses this amount of time to collect beacon signals, $t > T$

$N_{sent}(i, t)$: Number of beacons sent by anchor A_i in time t

$N_{recv}(i, t)$: Number of beacons received by unknown node in time t (beacons are sent By anchor A_i)

CM_i: Connectivity metric for anchor A_i

CM_{thresh}: Threshold for CM

(x_{cen}, y_{cen}): Estimated position of the unknown node by the Centroid algorithm

(x_a, y_a): Actual (or real) position of the unknown node

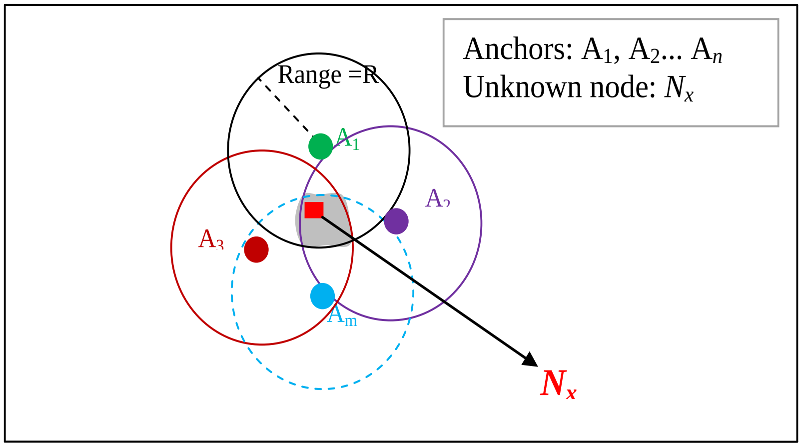


Figure 3-6: Centroid algorithm geo-localization

In this algorithm, during the fixed time period t, the unknown node N_x listens to the channel and gathers all the beacon signals from various neighbour anchor nodes. Even though each anchor A_i has sent *Nsent* (i, t) signals, due to radio propagation interference, the unknown node N_x may actually receive $N_{recv}(i, t)$ signals from A_i. Therefore, $N_{recv}(i, t)$ <= *Nsent* (i, t)). Having this metric, the authors then, define the connectivity metric for each anchor A_i, represented as CM_i so as to know whether an anchor node is really within the radio range of the unknown node N_x using Equation (3.6) below.

$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \tag{3.6}$$

They also set a threshold (referred as CM_{thresh}) for CM_i. The unknown node N_x considers the corresponding anchor A_i as its neighbour when CM_i is larger than CM_{thresh} otherwise it will not consider A_i as its neighbour and will not use it to estimate its position. At last, if

N_x has n anchor nodes (A_1, A_2 …, A_n) with connectivity metric larger than CM_{thresh}, then, N_x estimates its position at the centroid of these n anchor nodes using Equation (3.7):

$$\begin{cases} x_{cen} = (x_1 + x_2 + \ldots + x_n)/n \\ y_{cen} = (y_1 + y_2 + \ldots + y_n)/n \end{cases} \tag{3.7}$$

Finally, the Centroid algorithm programme procedure is given below:

Algorithm: Centroid

During a period t, normal node N obtains the positions of k anchors (A_1, A_2 …, A_k)
$x_{cen} \leftarrow 0$; $y_{cen} \leftarrow 0$
for $i \leftarrow 1$ to k
do $x_{cen} \leftarrow (x_{cen} + x_i)$; $y_{cen} \leftarrow (y_{cen} + y_i)$ where (x_i, y_i) is the position of A_i
xcen $\leftarrow$ x_{cen} / k; $y_{cen} \leftarrow y_{cen}$ / k
return x_{cen} and y_{cen}

Figure 3-7: Centroid Algorithm

The authors computes the location error of the algorithm using Equation (3.8) below.

$$location\ \ error = \sqrt{(x_{cen} - x_a)^2 + (y_{cen} - y_a)^2} \tag{3.8}$$

They conducted an experiment in a 10×10 m outdoor parking lot using 4 anchor nodes (at the different corners) with radio range about 8.94 m. In the experiment, the anchor nodes are set to transmit beacon signals of their positions every 2 seconds (T=2s) and CM_{thresh} is set to be 90%. An unknown node keeps static for 41.9s (t=41.9s) every time it moves to a new place inside the parking lot to receive the beacon signal for estimating its location. The result shows about 1.83m average location error.

Since then adding the RSSI information, some improvements have been proposed on Centroid algorithm. One of these is a Weighted Centroid Localization (WCL) algorithm [111]. Equation (2.10) shows how this algorithm computes the location of the unknown node, N_x, where (x_i, y_i) is the co-ordinate of anchor A_i, and w_i is the weight associated with the link between the unknown node and anchor node A_i. Here, the value of w_i is the RSSI value of anchor node A_i at the unknown node.

$$x_{wcl} = \frac{\sum_{i=1}^{m} (w_i \times x_i)}{\sum_{i=1}^{m} w_i} \ , \ y_{wcl} = \frac{\sum_{i=1}^{m} (w_i \times y_i)}{\sum_{i=1}^{m} w_i} \tag{3.9}$$

Their experiment shows the RSSI values sensed by the unknown node are always in the range of [–110 dB, –50 dB]. In order to make w_i to be positive, they calculated it using equation (3.10).

$$w_i = \frac{1}{-(RSSI_i + 49)} \tag{3.10}$$

Findings of this experiment have indicated that, the estimated position by WCL algorithm get closer to anchor nodes with higher weights when compared with the original Centroid algorithm which estimates at the centre of all anchor nodes.

3.1.1.2 Original CPE Algorithm

The Convex Position Estimation (CPE) algorithm was first proposed by Doherty et al to advance the accuracy of Centroid algorithm [112]. The authors first present their optimization concept then they estimate the positions of the unknown nodes using a joint optimization problem. For example, in Figure 3.8, where the three anchor nodes A_1, A_2, A_3 have the same communication range, the unknown node N_x attempts to estimate its location inside the communication (transmission) overlap of its neighbour anchor nodes.

The heart of the CPE algorithm is computing the smallest rectangle which bounds the communication overlap (shaded area in Figure 3-8), and then estimating the location of the unknown node N_x at the centre of this rectangle. To explain how they find this smallest rectangle, the authors propose an abstract optimization model.

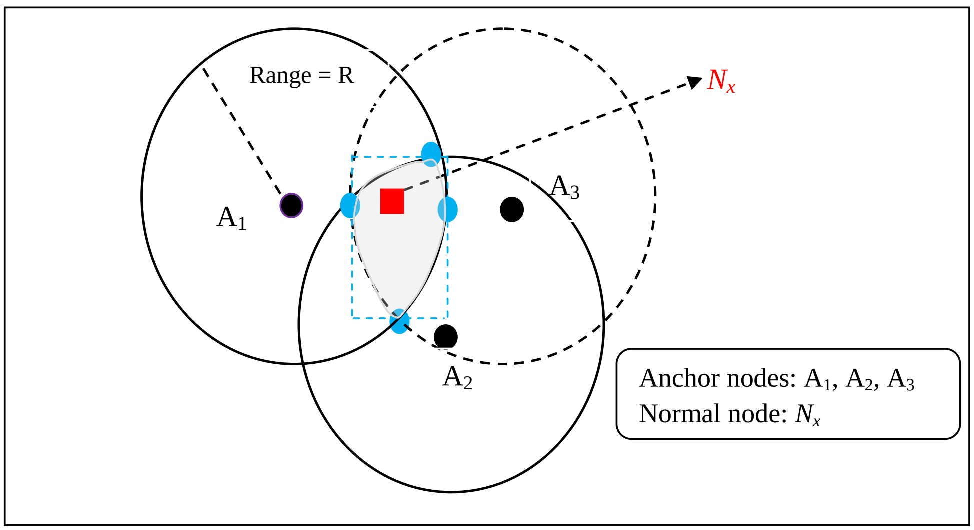


Figure 3-8: Example of CPE algorithm

For an overview of this abstract optimization process, let us look at Figure 3-9. In Figure 3-9(a), the unknown node N_x starts with a big rectangle. Next, N_x starts to optimize one side, in the figure, the right side of the rectangle. Going through a large amount of tests and computations, the exact right side is found as in Figure 3-9(b). In the same way, N_x computes for other sides. Eventually, the smallest rectangle shown in Figure 3-9(c) is found. The centre of this rectangle is considered as the location of the unknown node, N_x.

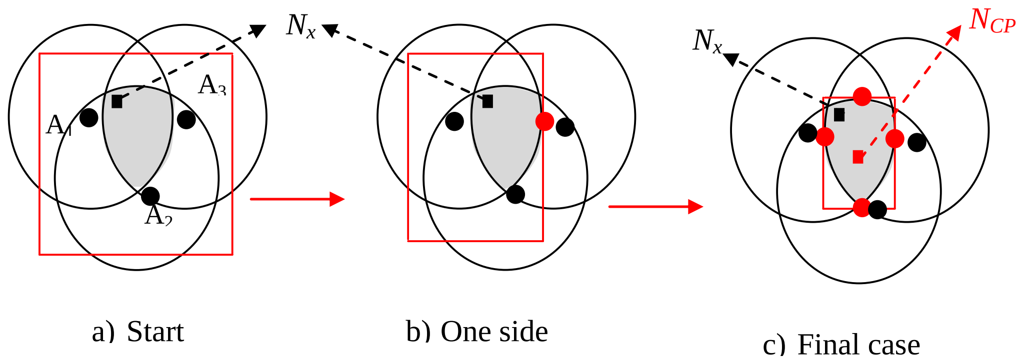


Figure 3-9: Process of Original CPE algorithm in finding smallest rectangle

Since the resource-limited unknown node is unable to do complex computations required by the optimization process, the original CPE algorithm is a centralized localization system. As a result, all unknown nodes are required to send the collected connectivity information to a centralized controller first. Then, the centralized controller estimates the location of every unknown node, and sends them to the corresponding unknown nodes. This causes high traffic; in other words, when the network is large, the CPE algorithm scales poorly.

3.1.1.3 Improved CPE Algorithm

To overcome the shortcomings of Original CPE, an improved, simplified and distributed version of Original CPE algorithm (which we call here after Improved CPE) is proposed [113] [114]. Although both algorithms look for the smallest rectangle which bounds the communication overlap of neighbour anchor nodes, they differ on how to find this rectangle. As shown in Figure 3-10, in finding the smallest rectangle, the Improved CPE algorithm finds an Estimated Rectangle (*ER*) which is usually bigger than the original smallest rectangle without using a complex optimization process.

In estimating the location of the unknown node, the Improved CPE algorithm performs the following three steps: i) the unknown node N_x transmits a location request signals to its neighbour anchor nodes, ii) then, receiving the request, the neighbour anchor nodes (A_1, A_2, A_3 in the figure) transmit their coordinates (x_1, y_1), (x_2, y_2), (x_3, y_3) with the agreed response to N_x. iii) Finally, N_x estimates its position N_{ER} (x_{ER}, y_{ER}) as the centre of ER using Equation (3.11) below.

$$x_{ER} = \frac{\min x_i + \max x_i}{2}, \; y_{ER} = \frac{\min y_i + \max y_i}{2} \tag{3.11}$$

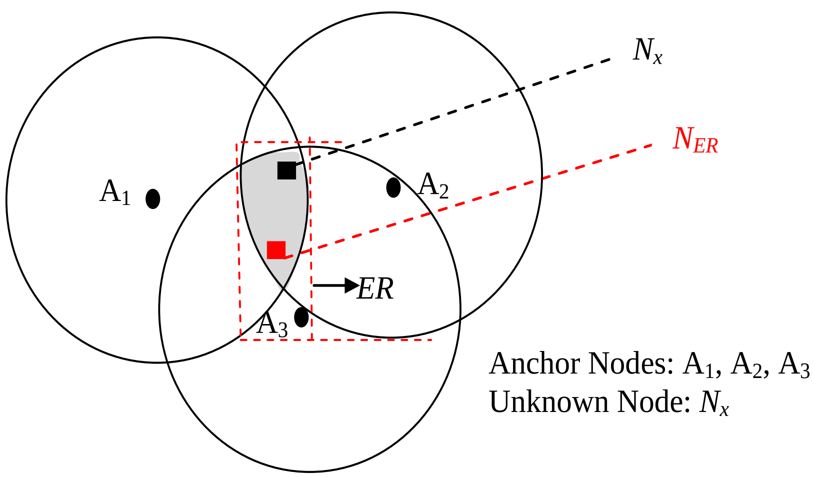


Figure 3-10: Improved CPE Algorithm

Finally, the programme procedure of the Improved CPE algorithm is summarized as follows:

Algorithm: “Improved CPE”

Suppose the normal node N_x has n neighbour anchor nodes $A_1, A_2 \ldots, A_n$. The position of A_i is (x_i, y_i).

$x_{max} \leftarrow x_1$; $x_{min} \leftarrow x_1$; $y_{max} \leftarrow y_1$; $y_{min} \leftarrow y_1$;

for $i \leftarrow 2$ **to** n

  if $x_i < x_{min}$ **then** $x_{min} \leftarrow x_i$; **elseif** $x_i > x_{max}$ **then** $x_{max} \leftarrow x_i$

  if $y_i < y_{min}$ **then** $y_{min} \leftarrow y_i$; **elseif** $y_i > y_{max}$ **then** $y_{max} \leftarrow y_i$

$x_{ER} \leftarrow (x_{min} + x_{max})/2$; $y_{ER} \leftarrow (y_{min} + y_{max})/2$

return x_{ER} and y_{ER}

Figure 3-11: Procedure of Improved CPE algorithm

3.1.1.4 APIT Algorithm

He et al. proposed an Approximate Point-In-Triangulation (APIT) test based on the principle of dividing the whole network into triangular regions between anchor nodes [115]. Theses triangular regions are made up of vertices formed by all the possible sets of connecting three neighbouring anchor nodes. The unknown node applies a test to determine whether it is inside or outside the triangle formed. The test is repeated for all connecting three anchor nodes heard by unknown node, and the location is estimated as the centre of gravity of the intersection of all the triangles that the node may exist in.

For example, let us assume that in a network there are 5 anchor nodes (A_1, A_2, A_3, A_4, and A_5) as indicated in Figure 3-12. In the figure, the solid circles are anchor nodes, while the hollow circles are unknown nodes. The concerned unknown node is marked as N_x.

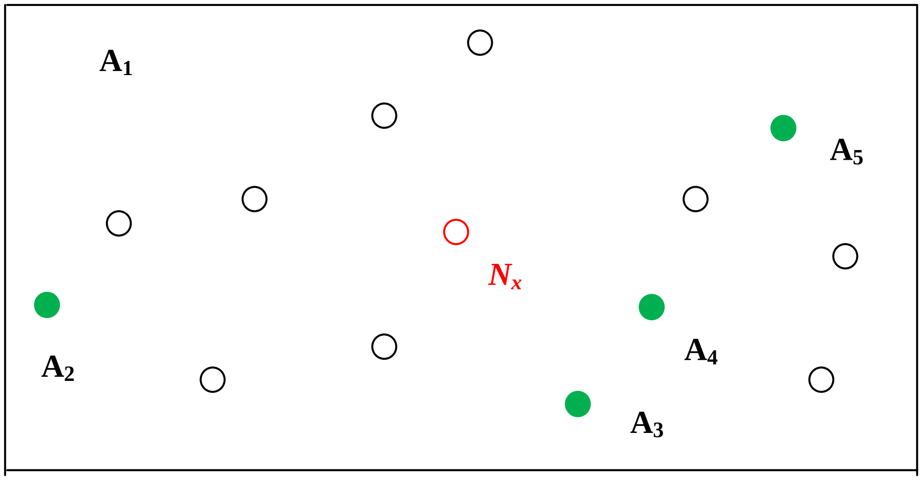


Figure 3-12: A APIT algorithm network scenario

Like other range-free algorithms, the APIT algorithm also assumes the unknown node N_x is aware of the positions of anchor nodes. Hence, as shown in Figure 3-13, N_x can form triangles using any three anchor nodes. The overlap of the triangles (N_x inside i.e. shaded area in Figure 3-13) is where N_x resides. To do this, however, N_x has to determine whether it is inside or outside of these triangles as described in step two of the algorithm.

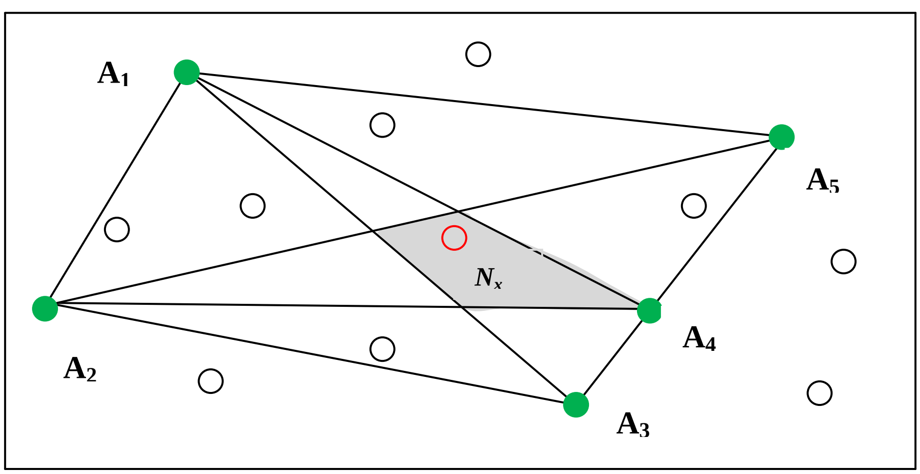
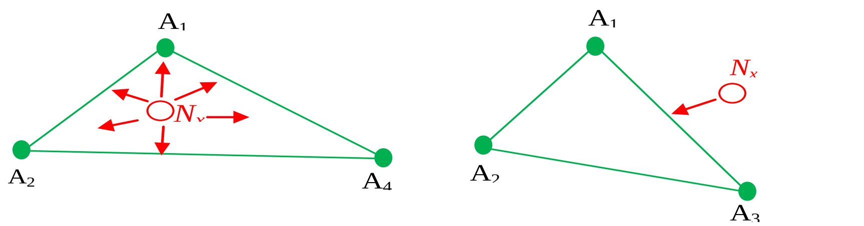


Figure 3-13: Formation of triangles by any three anchor nodes in APIT algorithm

The APIT algorithm has 4 steps: 1) beacon exchange, 2) Point-in-Triangulation (PIT) testing, 3) APIT aggregation, and 4) Centre of Gravity (COG) position calculation. While describing these steps, emphasis is given on the second step as it is the heart of the algorithm: First step, *beacon exchange*: It is assumed that the anchor nodes periodically broadcast beacon signals (containing their positions) to its neighbour nodes. In this algorithm, it is mandatory for each anchor node to be equipped with a powerful transceiver in order to make its signal received by all unknown nodes in the network. Receiving the signal from an anchor A_i, each unknown node detects and records the received signal's RSSI value plus the position of anchor node A_i. Here, the RSSI information is used in the PIT testing step to estimate whether a node is inside a triangle formed by three anchor nodes.

Second step, *the Point-in-Triangulation (PIT) testing:* This step checks if an unknown node N_x resides inside a triangle formed by three anchors or not. As indicated in Figure 3-14, the Perfect PIT test can be achieved by moving N_x along in any direction. For example, in Figure 3-14 (a), the unknown node N_x moves in every possible direction comparing its distance to anchor nodes (RSSI is used to measure the distance). If N_x finds its distance to the three anchor nodes never increases or decreases simultaneously, it assumes it is inside the three anchor nodes otherwise it is not.

Let us take an example, in Figure 3-14(a) when N_x moves a little to A_1, its distance to A_1 decreases, but its distances to A_2 and A_4 both increase. In this case, N_x is considered to be inside the triangle $\Delta A_1A_2A_4$ (triangle taken from Figure 3-13 above). In contrast, N_x will be considered outside a triangle when N_x moves a little, if its distances to the three vertexes of the triangle increase or decrease simultaneously. In Figure 3-14(b), for instance, when N_x moves a little in the direction shown by an arrow, its distances to three anchor nodes decrease simultaneously; as a result, N_x is considered outside the triangle $\Delta A_1A_2A_3$ (triangle taken from Figure 3-13 above).



b) Inside the triangle $\Delta A_1A_2A_4$ a) Outside the triangle $\Delta A_1A_2A_3$

Figure 3-14: Perfect PIT Test

However, in terms of implementation, the Perfect PIT test has two shortcomings: First, it is impossible to test in all directions, because there are infinite directions around the unknown node N_x. Second, the Perfect PIT test requires that unknown nodes can move, however, unknown nodes may be fixed in some applications; even if they move, their movement may not be only for the goal of localization.

For that reason, as a substitute to Perfect PIT, an Approximate PIT (APIT) test is preferred. The APIT test assumes unknown nodes are static. Even if unknown nodes cannot move, the APIT method imagines they could move, and regards their neighbour nodes as their positions after moving. In the same way as N_x, its neighbour unknown nodes have also received signals sent from anchors, and have noted down the corresponding RSSI values. This means N_x can communicate with these neighbour unknown nodes and receive their RSSI values. The received RSSI values are used to compare the relative distance to determine whether a node is further to an anchor than the other node.

In order to verify whether N_x is inside the triangle or not, the Perfect PIT test controls N_x to move a very tiny step and then observes the change of its distances to anchor nodes; on the other side, in APIT test, the static N_x virtually moves to its neighbour unknown nodes.

Unlike tiny moves in Perfect PIT test, the big moves to neighbour unknown nodes can cause test errors in APIT.

Third step, *APIT aggregation:* At the end of the individual APIT tests, through a grid SCAN algorithm APIT aggregates the results (inside/outside decisions of which some may be wrong). Inside decision is regarded as the truth and incremented; on the other side, outside decisions are regarded as false and the grid area is decremented. Finally using this information the maximum overlapping area is estimated.

Fourth step, *COG position calculation:* Finally, the estimated location of the unknown node is regarded as the centre of an overlap formed from the triangles where the unknown node N_x resides.

However, the APIT algorithm is not a preferred localization algorithm in wireless networks because of the following shortcomings: i) it requires long-range anchor node stations, which requires expensive high-power transmitters. ii) It is not flexible to adjust itself to the mobility or static nature of nodes. iii) It usually requires more anchor nodes than the average number of anchors used in localization. iv) The APIT test can cause serious errors as described above and the localization accuracy in this method is affected by a node's presence whether it is within the triangular regions or not.

3.1.1.5 Mid-perpendicular Algorithm

Mid-perpendicular algorithm [116] [117] was proposed to improve the accuracy of Centroid and CPE algorithms. The core principle of this algorithm is to find the centre of the anchor nodes communication overlap and take this centre as the estimated position of the unknown node.

The authors first presented the algorithm when an unknown node has only 3 neighbour anchor nodes then generalized the algorithm to work with more neighbour anchor nodes. Like Centroid and CPE algorithms, the Mid-perpendicular algorithm also assumes the communication ranges of anchor nodes are all the same. As shown in Figure 3-15, the unknown node N_x has three neighbour anchor nodes (A_1, A_2 and A_3) which mean N_x locates in the communication overlap region of anchor nodes A_1, A_2 and A_3. This overlap is marked as the shaded part in the figure.

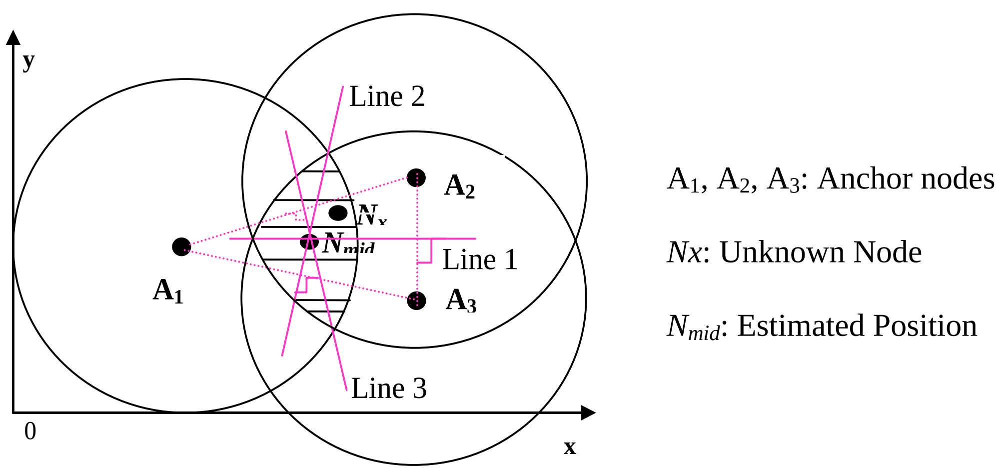


Figure 3-15: Mid-Perpendicular when an Acute Triangle (with 3 neighbour anchor nodes)

To find the centre of this communication overlap region, the authors use the concept of mid- perpendicular. As shown in Figure 3-15 "$Line_1$" is the mid-perpendicular of the line connecting the anchors A_2 and A_3. Hence, $Line_1$ passes through the middle point between A_2 and A_3, and it crosses the line (which connects A_2 and A_3) at a right angle. According to the symmetry, $Line_1$ goes through the centre of the overlap region. Similarly, $Line_2$ is the mid-perpendicular of the line connecting A_1 and A_3 while $Line_3$ is the mid-perpendicular of the line connecting A_1 and A_2. Both $Line_2$ and $Line_3$ go through the centre of the communication overlap region. Thus, the cross point of the three mid-perpendicular lines ($Line_1$, $Line_2$ and $Line_3$) can be taken as the centre of the overlap. They denote this cross point as N_{mid}, which is the estimated position of unknown node by this algorithm. In fact, the cross point can be computed using only two mid-perpendicular lines; for instance, $Line_1$ and $Line_2$. If the coordinates of the three anchors (A_1, A_2 and A_3) are respectively (x_1, y_1), (x_2, y_2), and (x_3, y_3), then $Line_1$, which is the mid-perpendicular of line A_2A_3, can be presented as in Equation (3.12).

$$y - \frac{y_2 + y_3}{2} = \left(x - \frac{x_2 + x_3}{2} \right) \frac{x_2 - x_3}{y_3 - y_2} \tag{3.12}$$

Line2, which is the mid-perpendicular of line A_1A_3, can be expressed as:

$$y - \frac{y_1 + y_3}{2} = \left(x - \frac{x_1 + x_3}{2} \right) \frac{x_1 - x_3}{y_3 - y_1} \tag{3.13}$$

The cross point, $N_{\text{mid},}$ of the above two mid-perpendicular lines with its coordinates (x_{mid}, y_{mid}), can be computed as:

$$\begin{cases} x_{mid} = \dfrac{\left(x_1^2 - x_2^2\right)\left(y_3 - y_1\right) + \left(x_1^2 - x_3^2\right)\left(y_1 - y_2\right) + \left(y_1 - y_2\right)\left(y_2 - y_3\right)\left(y_3 - y_1\right)}{2[y_1\left(x_2 - x_3\right) + y_2\left(x_3 - x_1\right) + y_3\left(x_1 - x_2\right)]} \\ \\ y_{mid} = \dfrac{\left(y_1^2 - y_2^2\right)\left(x_3 - x_1\right) + \left(y_1^2 - y_3^2\right)\left(x_1 - x_2\right) + \left(x_1 - x_2\right)\left(x_2 - x_3\right)\left(x_3 - x_1\right)}{2[y_1\left(x_2 - x_3\right) + y_2\left(x_3 - x_1\right) + y_3\left(x_1 - x_2\right)]} \end{cases} \tag{3.14}$$

Equation (3.14) a complex one, works when N_x's 3 neighbour anchor nodes (A_1, A_2 and A_3) form an acute triangle, where all the angles are less than 90 degrees.

Nevertheless, if the 3 neighbour anchor nodes form a right angled triangle or an obtuse angled triangle, then the computation of N_{mid} is simpler than the Equation (3.14). For example, Figure 3-16 shows the scenario when A_1, A_2 and A_3 form a right angled triangle where $\angle A_1A_3A_2$ is 90 degrees, and the side A_1A_2 is the longest side of the triangle. From this figure we can see the cross point of the three mid-perpendiculars is just the middle point of the longest side A_1A_2. When the three neighbour anchor nodes form a right angled triangle, N_{mid} is the middle point of the longest side in the triangle which can be represented in Equation (3.15), where side A_iA_k is the longest side of the triangle $\Delta A_1A_2A_3$ with the coordinates (x_i, y_i) and (x_k, y_k) for A_i and A_k, respectively:

$$\left(x_{mid}, y_{mid}\right) = \frac{\left(x_i + x_k, y_i + y_k\right)}{2}, \tag{3.15}$$

where, A_i *and* A_k form the longest side of the right angled triangle.

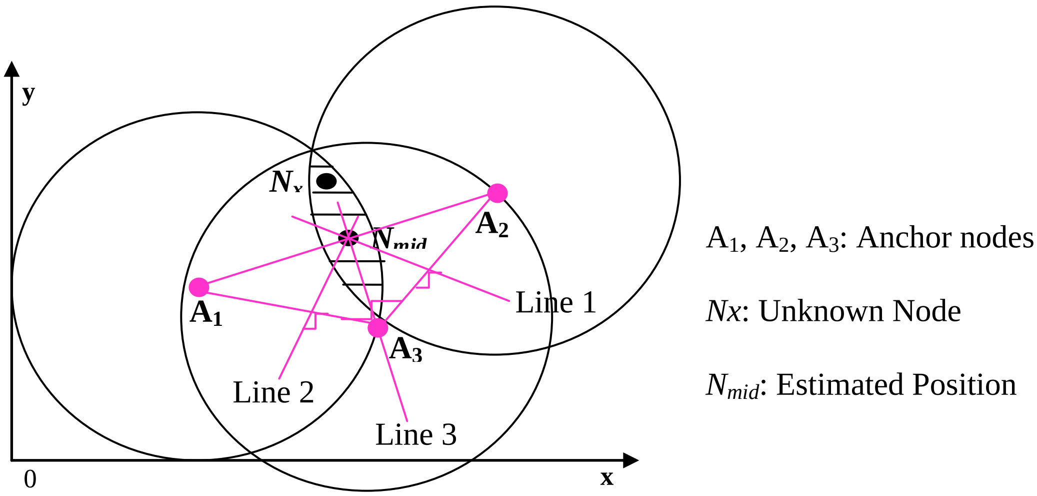


Figure 3-16. Mid-Perpendicular when Right Angled Triangle (with 3 Neighbour Anchor Nodes)

Moreover, the same equation also works in the case of an obtuse angled triangle; for instance, A_1, A_2 and A_3 form an obtuse angled triangle, as shown in Figure 3-17.

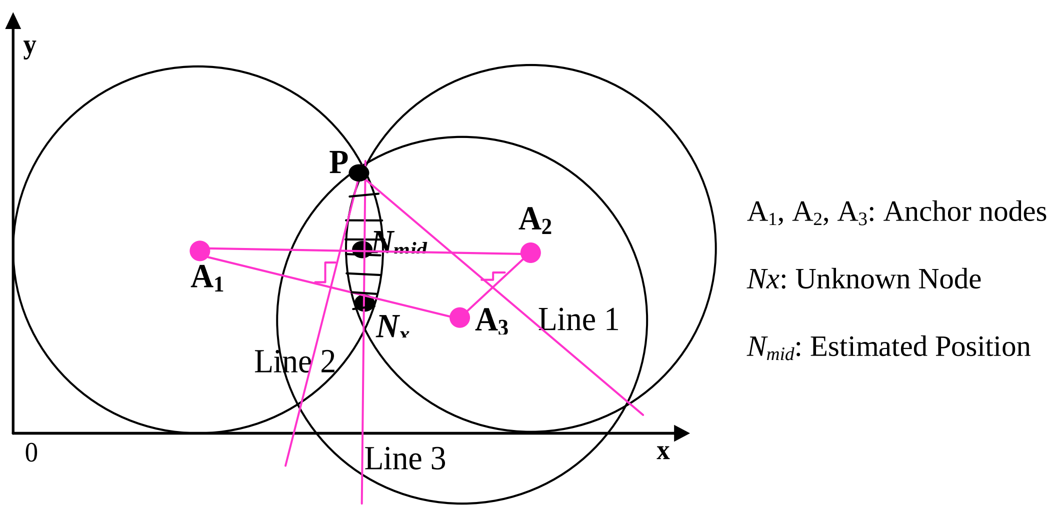


Figure 3-17: Mid-perpendicular algorithm when an Obtuse Angled Triangle (with 3 neighbour anchor nodes)

Angle $\angle A_1A_3A_2$ is larger than 90 degrees, and the side A_1A_2 is the longest side of the triangle. From this figure we can see the cross point of the three mid-perpendicular lines, indicated as "*P*", is going to be out of the overlap region. Instead, the middle point of side A_1A_2, denoted as N_{mid}, becomes the centre of the overlap. Hence, when the three neighbour anchors form an obtuse triangle, N_{mid} is still the middle point of the longest side in the triangle.

To compute N_{mid}, when N_{x} has only three neighbour anchor nodes, the authors generalized the above equations as in Equation (3.16):

$$N_{mid}\left(x_{mid}, y_{mid}\right) = \begin{cases} \text{Equation (3.14), if three anchors form an acute triangle} \\ \text{Equation (3.15), others} \end{cases} \tag{3.16}$$

The Equation (3.16) is a generalization to estimate the location of unknown node when it has 3 neighbour anchor nodes, but when N_{x} has in total m neighbour anchor nodes (m>3) a more complex calculation is proposed by the authors as follows: Among the m neighbour anchor nodes A_1, A_2, … A_{m}, any three anchors can give one estimated position "N_{mid}" using the Equation (3.16); therefore, as many as C_3^m locations of N_x can be generated. The authors considered the average of all these positions as the final estimated position of the unknown node, N_x.

The programme procedure of Mid-perpendicular algorithm is presented in Figure 3-18 below:

Algorithm: "Mid-perpendicular"

1. Suppose the normal node N has m neighbour anchors $A_1, A_2 \ldots, A_m$.
2. $x_{mid} \leftarrow 0$; $y_{mid} \leftarrow 0$;
3. **for** $i \leftarrow 1$ **to** $(m\text{-}2)$
4. A_i is chosen. (x_i, y_i) is the position of A_i.
5.   **for** $j \leftarrow (i+1)$ **to** $(m\text{-}1)$
6. A_j is chosen. (x_j, y_j) is the position of A_j.
7.   **for** $k \leftarrow (j+1)$ **to** m
8. A_k is chosen. (x_k, y_k) is the position of A_k.
9. $(x_{mid}, y_{mid}) \leftarrow$ calculated as Equation (3.16) based on the anchors A_i, A_j, A_k
10. $\overline{x}_{mid} \leftarrow \overline{x}_{mid} + x_{mid}$; $\overline{y}_{mid} \leftarrow \overline{y}_{mid} + y_{mid}$
11. $\overline{x}_{mid} \leftarrow \overline{x}_{mid} / C_m^3$; $\overline{y}_{mid} \leftarrow \overline{y}_{mid} / C_m^3$
12.   **return** $\overline{x}_{mid}$ and $\overline{y}_{mid}$

Figure 3-18: Procedure of Mid-perpendicular algorithm

However, for small wireless energy constrained devices, this method is complex and expensive (as discussed in the complexity of algorithms in Section 4.4.3). On top of the complexity of the algorithms, it also requires to decide what type of triangle any three possible anchor nodes form which is C_3^m triangles, where m is the number of neighbour anchor nodes. This adds another complexity on an already computationally complex algorithm for small wireless nodes. This indicates a need for an algorithm which is computationally simple but still gives relatively better accuracy.

3.2.2 DV-hop Based Methods

Hop-counting technique also called Distance Vector – hop (DV-Hop) method was first proposed by D. Niculescu and B. Nath [118]. In this method, each unknown node looks for anchor nodes for their estimated hop sizes and position and then attempts to get the smallest hop count to each anchor nodes using assigned routing protocol. Then, it estimates its distances to each anchor nodes using the hop counts to them. Finally, the unknown node uses trilateration to compute its estimated location using the estimated distances to three appropriate neighbour anchor nodes.

Although the above local method algorithms (Centroid, Original CPE, Improves CPE, APIT and Mid- Perpendicular) all require a unknown node has at least 3 neighbour anchor nodes, in practical scenarios, the number of unknown nodes is always more than anchor nodes. There may be a case where unknown nodes may have less than 3 neighbour anchor nodes, or even no neighbour anchors. In Figure 3-19; for example, in a small network topology there are 3 anchor nodes (A_1, A_2, A_3) and 5 unknown nodes (N_1, N_2, N_3, N_4, N_x). The connectivity between nodes is displayed by lines of dashes. We can see that the concerned unknown nodes N_x and N_4 have no neighbour anchor nodes, while the other unknown nodes all have only one neighbour anchors. This shows that none of the algorithms already introduced in previous subsections (including the range-based schemes) can localize these unknown nodes. Nevertheless, in such cases DV-hop algorithm can be used. Its localization steps are presented as follows:

The estimation of the unknown node's location in DV-Hop algorithm involves four steps:
Step 1: Flooding throughout the network each anchor node broadcasts a beacon packet with the node location and a hop-count value initialized to one. Each receiving node

retains only the minimum hop-count value per each anchor node of all packets it receives while packets with higher hop count values to a particular anchor node are ignored, defined as invalid information. Next the retained valid packets are flooded outward in the network incrementing hop-count values by one at every intermediate hops. In this manner, the entire nodes in the network get the smallest hop-count to every anchor node.

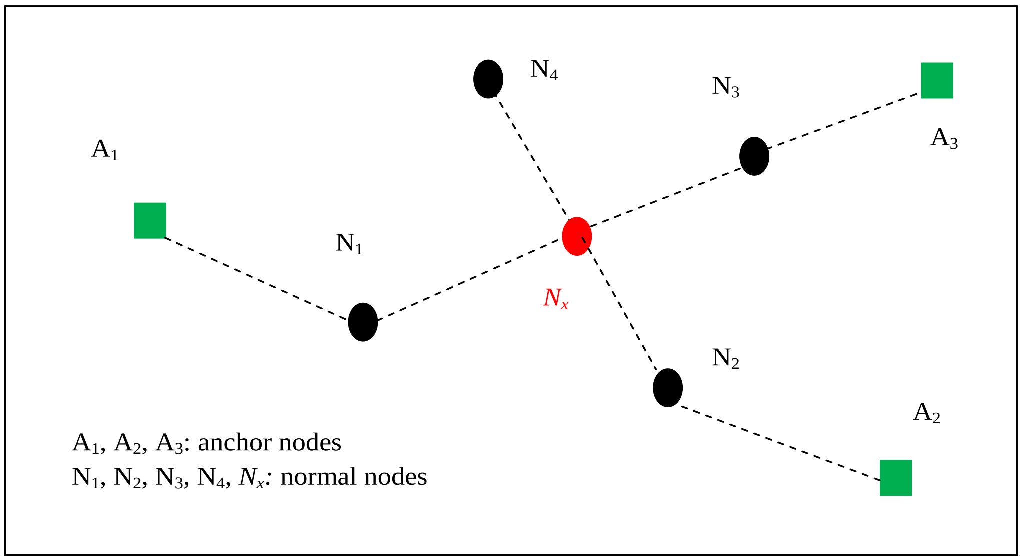


Figure 3-19: Example of network topology for DV-hop algorithm

For example, if we look at the case in Figure 3-19, at *Step 1*, the anchor node A_1 broadcasts a message carrying its position (x_1, y_1) and a hop count field initialized as 0. Upon receiving this message, the unknown node N_1, first increments the hop count field by 1. Right now, the hop count message is renewed by $N_{1;}$ as a result, the value of the hop count field message is 1. Then, N_1 records the information in its database, noting down A_1's position (x_1, y_1) and setting $hop_{1,N1}$ (here, $hop_{1,N1}$ is N_1's hop count to A_1) to be 1. After this, N_1 broadcast the renewed message and N_x receives the broadcast message. As N_1, N_x carry out the same process, including renewing the message, recording the message information into N_x's database, and relaying the message. As a result, N_x can know its hop count to A_1 (denoted as $hop_{1,Nx}$) as 2. In the same way, N_x can be aware of its hop counts to other anchors, as well as other anchors' positions.

Step 2: Calculating Average Hop Size: Having the hop-count value to other anchor nodes, an anchor node estimates an average hop size and floods it to the entire network. Then, unknown nodes receiving the hop-size, they multiply the hop-size by the hop-count value to estimate their distance to the anchor node. The average hop-size is computed by anchor node i using the following Equation:

$$Hop\_Size = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_{i,j}} \tag{3.17}$$

where (x_i, y_i), (x_j, y_j) are coordinates of anchor nodes i and beacon node j, $h_{i,j}$ is the hop count between node i and node j. Each anchor node broadcasts its *hop-size* to the network using controlled flooding.

Step 3: Receive Hop-size Information: Unknown nodes receive and save *hop-size* information and they transmit the *hop-size* to their neighbour nodes.

Step 4: Chooses three Anchor Nodes: Each unknown node in the network chooses three anchor nodes which are closest to it and computes the distance to the chosen anchor nodes. Finally, use trilateration to estimate the position of the unknown node.

Despite the fact that the DV-hop algorithm can localize the unknown nodes which have less than three neighbour anchor nodes, its coarse localization accuracy needs to be improved. Consequently, there are many follow-up studies of DV-Hop method, but in the following subsections only the representative DV-hop algorithms are presented such as DDV-hop (Differential DV-hop), Self-adaptive DV-hop, and Robust DV-hop.

3.1.1.1 DDV-hop Algorithm

The DV-hop follow-up work called DDV-hop (Differential DV-hop) algorithm changes *Step 2* and *Step 3* of the original DV-hop algorithm [119].

DDV-hop algorithm in Step #2, each anchor A_i in addition to its hop-size also called distance-per-hop (dph_i), it also broadcasts the differential error of dph_i to the entire network. The differential error, denoted as $diff\_err_i$, is computed as:

$$diff\_err_i = \frac{\sum_{j \neq i} \left| dph_i - \frac{d_{i,j}}{hop_{i,j}} \right|}{m_d - 1} \tag{3.18}$$

where m_d is the number of anchor nodes, $hop_{i,j}$ is the hop count between A_i and every other anchor A_j, and $d_{i,j}$ is the distance between A_i and A_j. Here, $hop_{i,j}$ is obtained through *Step 1*, while $d_{i,j}$ is calculated as $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

In *Step 3*, DDV-hop differs from original DV-hop on how to compute the estimated distance between an unknown node N_x and each anchor A_i. In the original DV-hop algorithm, when an unknown node N_x receives the distance-per-hop value of A_i, N_x immediately computes its estimated distance to A_i as $dph_i \times hop_{i,Nx}$, but in DDV-hop algorithm, N_x uses its own distance-per-hop value denoted as dph_{ddv} instead of dph_i. Here, dph_{ddv} is computed as the weighted sum of all anchor nodes' distance- per-hop. The weighting coefficient of dph_i, referred as λ_i, is determined by the differential errors of anchor nodes' distance-per-hop as follows:

$$\lambda_i = \frac{diff\_err_i}{\sum_{k=i}^{m^*} \left| diff\_err_k \right|} \tag{3.19}$$

where m^* is the number of differential errors received by N_x. Then, dph_{ddv} is computed as:

$$dph_{ddv} = \sum_{i=1}^{m^*} \left(\lambda_i \times dph_i \right) \tag{3.20}$$

To compute a more accurate average distance per hop for unknown nodes, a weighted method is used based on the differential error of each anchor node's distance-per-hop. The authors proposed this algorithm to improve the accuracy in asymmetry (not regularly) distributed wireless sensor networks. In such networks, their simulation results show DDV-hop algorithm increases accuracy by about 18%-20% from DV-hop algorithm; of course, on the cost of high network traffic because the new data "differential error" (*diff_err*i) should be broadcasted by every anchor node in the network adding another cost on the scarce battery of nodes.

3.1.1.2 Self-Adaptive DV-hop Algorithms

In Self-Adaptive DV-hop algorithms one that requires RSSI information and one which does not were also proposed as original DV-hop algorithm follow-ups [120]. Here, one without RSSI information Self-Adaptive DV-hop algorithms is presented. This algorithm changes *Step 3* of the original DV-hop algorithm, still having the same network overhead.

When an unknown node N_x computes its estimated distance to A_i at *Step 3*, distance-per-hop value denoted as dph_{adp}, which is computed as the weighted sum of anchor nodes' distance-per-hop, is used instead of the original DV-hop's distance-per-hop. Unlike in DDV-hop algorithm, in Self-Adaptive algorithm the weighting coefficient (λ_i^a) of dph_i (each anchor A_i's distance-per-hop), is determined based on N_x's hop counts to A_i; the more hops between N_x and A_i, the smaller value assigned to λ_i^a which is computed as follows:

$$\lambda_i^a = \frac{\left(\sum_{k=1}^{m_d} hop_{k,Nx}\right) - hop_{i,Nx}}{(m_d - 1)\sum_{k=1}^{m_d} hop_{k,Nx}} \tag{3.21}$$

Then, dph_{adp} is calculated as:

$$dph_{ddv} = \sum\left(\lambda_i^a \times dph_i\right) \tag{3.22}$$

The governing principle in this work is that the nearest anchor node to an unknown node always has the most accurate average distance-per-hop which helps to compute a new distance-per-hop. Simulation results show the accuracy of the Self-Adaptive algorithm is 30% better than DV-hop algorithm. However, the simulation scenario is very special: the anchor nodes are distributed at the corners of the simulation area, and the unknown nodes are regularly distributed inside the area.

3.1.1.3 Robust DV-hop Algorithm

Robust DV-hop (RDV) algorithm is another follow-up localization algorithm of the original DV-hop algorithm [121]. RDV-hop algorithm computes the distance-per-hop value between N_x and A_i, (referred as $dph_{Nx,i}$.) as the weighted sum of the distance-per-hop values between A_i and every other anchor A_k. Here, the distance-per-hop between A_i and A_k is denoted as $dph_{i,k}$, which is calculated as:

$$dph_{i,k} = \frac{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}}{hop_{i,k}} \tag{3.23}$$

where (x_i, y_i) and (x_k, y_k) are the positions of A_i and A_k, respectively. $hop_{i,k}$ is the hop count between A_i and A_k. Consequently, N_x needs to know the hop count between any two anchors. This necessitates each anchor to broadcast at *Step 2* its hop counts throughout the network. The weighting coefficient $\lambda_{i,k}$, is computed as follows:

$$\lambda_{i,k} = \frac{1}{(hop_{i,Nx} + hop_{Nx,k}) - hop_{i,k} + 1} \tag{3.24}$$

Subsequently, $dph_{Nx,i}$ can be computed as:

$$dph_{Nx,i} = \frac{\sum_{k \neq i} (\lambda_{i,k} \times dph_{i,k})}{\sum_{k \neq i} \lambda_{i,k}} \tag{3.25}$$

Here, if N_x is on the shortest path between A_i and A_k, this weighting coefficient will have the maximum value. The core principle of this algorithm is if an unknown node N_x places on the shortest path between two anchor nodes, then the distance-per-hop between the two anchors will be the most accurate for this unknown node. This concept helps this algorithm to obtain a more accurate average distance-per-hop for each unknown node. The simulation results show that Robust DV-hop algorithm has from 10% to 40% an accuracy improvement compared with original DV-hop algorithm depending on the distribution of sensor nodes. However, all the three nodes' distributions in the simulation are very special: the first is an isotropic (regular) distribution, the second is C-shaped distribution, and the third is X-shaped distribution. Moreover, this algorithm has high network traffic compared with the original DV-hop algorithm because at *Step 2*, in addition to the distance per hop, each anchor node needs to broadcast its hop counts to other anchor nodes throughout the network which incurs high network traffic.

When we look at these follow-up algorithms of original DV-hop, they try to get a more accurate weighted distance-per-hop value for each unknown node which requires additional information like differential error in [119]. This increases the algorithms computational complexity and broadcasting this computed additional information causes additional bottleneck to the network traffic consuming more energy of wireless nodes' scarce battery.

3.2.3 Summary of Range-Free Geo-localization Methods

This section tries to summarize the introduced popular range-free geo-localization algorithms discussed in the previous subsections; these algorithms are compared and contrasted as shown in Table 3-2.

The Centroid and the Improved CPE algorithms both have low network overhead and low calculation complexity, but with relatively low accuracy. Although the original CPE algorithm gives better accuracy, it is complex and centralized causing high network traffic and computational cost. The APIT algorithm is not a preferred algorithm because it requires special anchor nodes with high powered transmitters and also depends on the RSSI information of which signal varies in different environments. Mid-perpendicular algorithm also suffers its own shortcomings: i) the algorithm is a computationally complex algorithm to be used for energy scarce small wireless devices ii) it also require to detect or measure angle of a triangle formed from three anchor nodes to decide which equation has to be used between its two angle based equations which farther complicates the algorithm.

On the other hand, unlike the above local range-free methods requiring at least 3 neighbour anchor nodes for an unknown node to estimate its location, DV-hop based algorithms do not impose such a requirement. However, they have the following short comings when compared with local methods: i) they cause high network overhead; ii) they give poor accuracy which rarely satisfies the location accuracy demanded by many location based applications and services.

Table 3-2: Comparison of Range-Free Geo-localization Methods

Range-free Algorithms	Strength	Weakness
Centroid	Low overhead cost	-An unknown node needs at least 3 neighbour anchor nodes -Low precision
Original CPE	Good precision	-An unknown node needs at least 3 neighbour anchor nodes -Centralized, -High overhead cost
Improved CPE	Low overhead cost	-An unknown node needs at least 3 neighbour anchor nodes -Low precision
APIT	No ideal radio assumption	- Low precision -High power consumption, -RSSI needed
Mid-perpendicular	Good accuracy in regularly deployed	-An unknown node needs at least 3 neighbour anchor nodes -High computational complexity
DV-hop	No constraint on the number of neighbour anchor nodes	-Low accuracy, -High network overhead
DDV-hop		
Self-adaptive DV-hop		
Robust DV-hop		

From a general geo-localization perspective, the major strengths and weaknesses of range-based and range-free schemes are compared and contrasted in Table 3-3. When we compare the two broadly classified geo-localization methods, the range-free localization methods do not need any additional ranging hardware and can be taken as cost-effective solutions for battery scarce wireless network nodes; however, they are less precise than range-based localization methods. Hence, this work focuses on how to increase their localization accuracy.

Table 3-3: Comparison of Range-based and Range-free Geo-localization Algorithms

Range-Based	Range-Free
Better accuracy	Lower accuracy
High energy consumption	Low energy consumption
Need additional ranging devices	Does not need additional ranging devices
Easily affected by multi-path fading and noise	More robust
Use measured distance/angle/time to estimate location	Use radio range node connectivity metric
Non-smart methods	Smart methods

On the other hand, range-based localization methods are more accurate than range-free localization methods but consume high energy as they need dedicated expensive, specialized and battery intensive additional infrastructure for ranging, except RSSI. However, RSSI information is very sensitive to the environment; as a result, it gets easily affected by obstructions like walls, for instance, in buildings.

Therefore, we focus on the range-free localization algorithm. From range-free we focus on the local methods assuming unknown node has at least two neighbour anchor nodes in the crowd. In Chapter 4, we have introduced our novel range-free smart geo-localization algorithms in wireless networks which increase localization accuracy when compared with the state-of-the-art algorithms with reasonable computational complexity.

Deliberately left blank

CHAPTER 4 OUR SMART NOVEL RANGE-FREE GEO-LOCALIZATION ALGORITHMS FOR WIRELESS NETWORKS

This chapter presents our three but related geo-localization algorithms. Each algorithm is discussed in a separate subsection. Computational complexity of the algorithms is also discussed and compared with other related algorithms at the end of the chapter.

Since unknown node resides in the communication ranges overlap of neighbour anchor nodes, we hypothesis that to increase accuracy and to be a reliable range-free geo-localization algorithm, range-free local method geo-localization algorithms must always estimate the location of unknown nodes inside all neighbour anchor nodes' communication range overlap region. Determining the vertices of the smallest communication overlap polygon which we call the True Intersection Points (TIPs) and estimating the position of the unknown node N_x at the centre of these TIPs, 1) always guarantees the estimated position is inside and at the centre of the communication overlap region of anchor nodes; 2) increases accuracy because the TIPs are near to the unknown node, N_x when compared to the anchor nodes.

4.1 Centre of the Smallest Communication Overlap Polygon (CSCOP) Geo-localization Algorithm

Regardless of their accuracy, Centroid, Original CPE, Improved CPE, and Mid-perpendicular are popular range-free localization algorithms. However, accuracy is an important parameter for most location based applications (LBAs). Looking critically, these popular algorithms give relatively low accuracy when compared to the location accuracy demanded by most LBAs. To solve this shortcoming, we propose our localization algorithm called Centre of the Smallest Communication Overlap Polygon (CSCOP).

The core principle of CSCOP localization algorithm is: i) first, find the Smallest Communication Overlap Polygon (SCOP) where the unknown node resides, ii) then, estimate always the location of the unknown node at the centre of the SCOP. Other related algorithms have attempted to do this but no one has successfully managed it always. This solves the shortcomings of other related algorithms by estimating the location of the unknown node always inside SCOP. The problem is how to pinpoint the SCOP.

4.1.1 Pinpointing the Smallest Communication Overlap Polygon (SCOP)

Like Centroid, Original CPE, Improved CPE, Mid-perpendicular, and other related algorithms, we assume all anchor nodes have the same communication range. However, unlike other related algorithms which assume an unknown node N_x has at least 3 neighbour anchor nodes, CSCOP assumes at least 2. To illustrate the principle of pinpointing the SCOP, let us look at Figure 4-1 below:

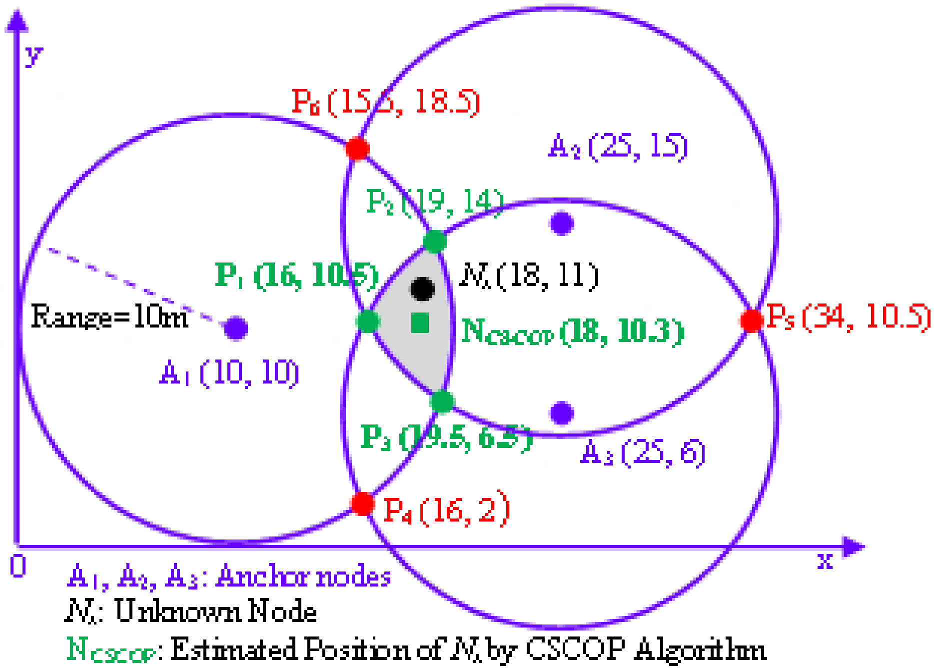


Figure 4-1: Pinpointing SCOP

In Figure 4-1, the unknown node N_x has 3 neighbour anchor nodes: (A_1, A_2, and A_3). As these are its neighbour anchor nodes, unknown node N_x actually locates inside the SCOP (the shaded area in the figure) of these anchor nodes. Hence, any range-free localization algorithm, to be reliable and with better location accuracy, has to always estimate the location of the unknown node inside this shaded region which we call the SCOP. So, we have to find SCOP first.

To pinpoint the SCOP, we perform the following 2 steps: first, find Intersection Points (IPs) of communication ranges of anchor nodes, second, determine True Intersection Points (TIPs) which make the vertices of the SCOP from False Intersection Points (FIPs) which do not make the vertices of the SCOP among IPs.

4.1.1.1 Finding Intersection Points (IPs)

In the Figure 4-1, we have 3 anchor nodes, which means 3 circles with radii equal to their communication range *R*. To find the intersection Points (IPs) of the circles, we solve each circle's IPs with every other circle in the iteration of C_2^n using Equation (4.1), where (x_i, y_i) and (x_j, y_j) are the positions of the two anchor nodes involved in the calculation of IPs and *R* is communication range of anchor nodes (or radii of a circles).

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 = R^2 \\ (x - x_j)^2 + (y - y_j)^2 = R^2 \end{cases} \tag{4.1}$$

The IPs of 2 circles can be 1, 2 or infinity. If it is infinity, the two circles are identical which means they are one and the same circle. Hence, 2 different circles intersect either at 1 point (as in Figure 4-2(a)) or at 2 points (as in Figure 4-3(a)). For example, if we take A_1 and A_2 anchor nodes in Figure 4-1, they intersect at points P_3 (19.5, 6.5) and P6 (15.5, 18.5). Totally, in Figure 4-1, using Equation (4.1), we have 6 IPs (P_1, P_2, P_3, P_4, P_5, and P_6) of anchor nodes A_1, A_2 and A_3.

4.1.1.2 Determining True Intersection Points (TIPs)

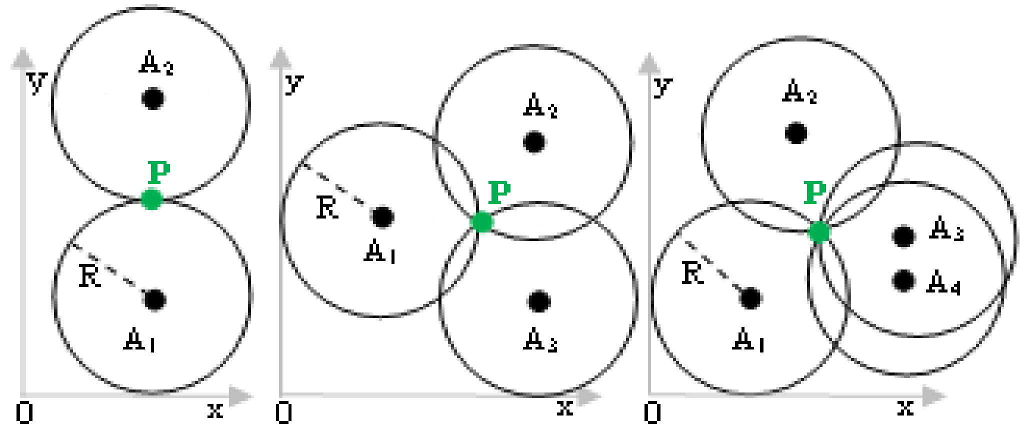
To determine TIPs which are the vertices of SCOP we check the distance between each IPs to each anchor nodes' position whether it is less than or equal to *R* (radius) using Equation (4.2), where $(x_{IPi}, y_{IPj})$and $(x_i, y_j)$are the coordinates of the IPs and anchor nodes' location respectively and *R* is communication range of anchor nodes.

$$(x_{IPi} - x_i)^2 + (y_{IPj} - y_j)^2 \leq R^2 \tag{4.2}$$

If IP's distance to each anchor node is less than or equal to R, that Intersection Point is taken as a TIP. If not, that point is considered as a False Intersection Point (FIP). For instance, using equation (2) among 6 IPs in Figure 4.1, P_1, P_2, and P_3 are classified as TIPs and P_4, P_5, and P_6 are classified as FIPs. Since we are interested in the TIPs as they form the vertices of the SCOP, we take only P_1, P_2, and P_3.

In Figure 4-1, when we compare positions of anchor nodes (A_1, A_2 and A_3) with positions of TIPs (P_1, P_2 and P_3), the positions of the latter are nearer to the unknown node, N_x. This increases location accuracy of the algorithm because estimation of the location of unknown node, N_x is based on TIPs. Moreover, SCOP (TIPs as its vertices) can be used as the smallest search area for the unknown node while its centre is the estimated location of the unknown node.

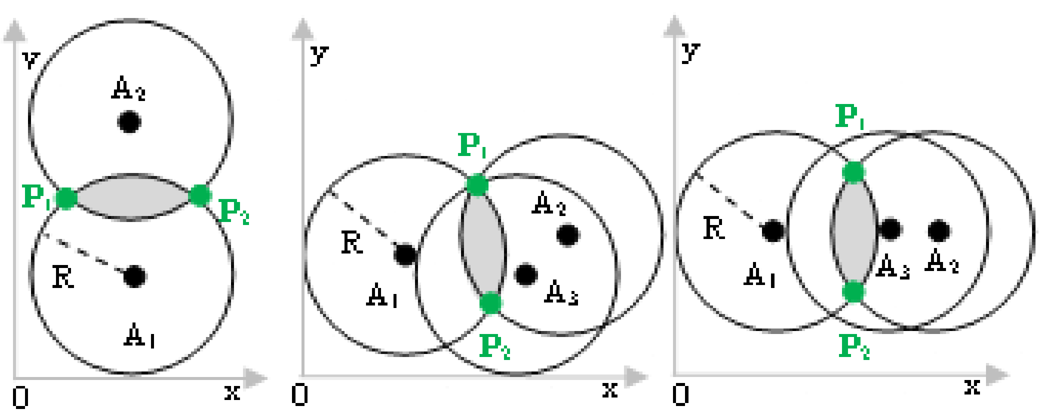
Now, let us look at the possible range of TIPs in different scenarios in the following figures: Figure 4-2 shows that two (Figure 4-2(a)) or more anchor nodes (Figure 4-2 (b) and (c)) can have a single TIP which is at the same time the SCOP of the anchor nodes. In such a scenario, the estimated and the real location of the unknown node N_x is the same, 0% location error. This can be true in theory but in reality it is difficult to achieve 100% accuracy.



a) b) c)

Figure 4-2: Senario-1, SCOP with one TIP regardless of increase in number of anchor nodes

Figure 4-3(a), 4-3 (b), and 4-3 (c) show the number of TIPs can be just two, regardless of the number of neighbour anchor nodes. In Figure 4-3(c), the two TIPs are formed from the intersection points of the two farthest anchor nodes: A_1 and A_2, but not A_3.



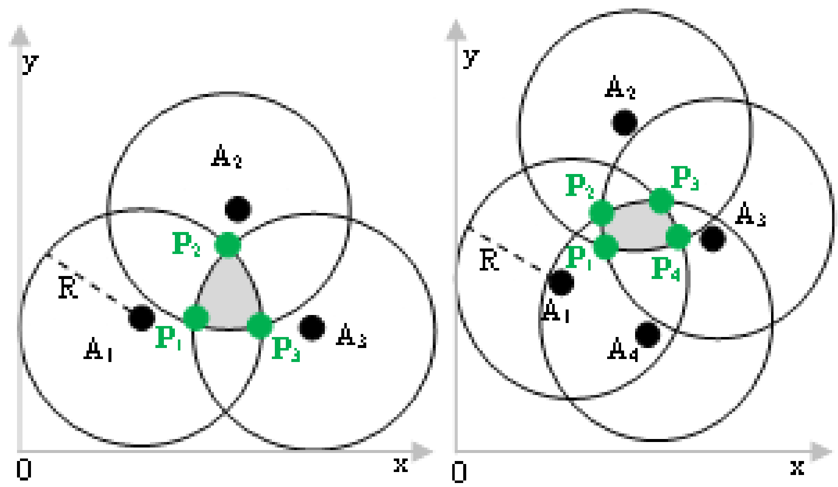
a) b) c)

Figure 4-3: Senario-2, SCOP as two TIPs

More importantly, in Figure 4-3(c), we can note that this algorithm excludes anchor nodes that does not contribute to TIPs, for example anchor node A_3. Moreover, Figure 4-

2(a), Figure 4-3(a) and 4-3(c) show us this algorithm can also work with only 2 anchor nodes.

Unlike Figure 4-2 and Figure 4-3, Figure 4-4 shows us when the number of neighbour anchor nodes increase in number, the number of TIPs may also increase. For example, when the number of anchor nodes increased from 3 in Figure 4-4(a) and to 4 in Figure 4-4(b), the number of TIPs has also increased from 3 to 4.



a) **b)**

Figure 4-4: Senario-3, SCOP with more than 2 TIPs

Looking at Figure 4-2 to Figure 4-4, the TIPs can be: i) just a point as shown in Figure 4-2 (Senario-1), ii) two points as in Figure 4-3 (Senario-2) or iii) more than two points as in Figure 4-4 (Senario-3). Hence, TIPs may range from 1 to *k* where *k* is the number of anchor nodes. In other words, the number of TIPs is less than or equal to the number of anchor nodes.

4.1.2 The Centre of the Smallest Communication Overlap Polygon

Once we pinpoint SCOP by identifying the TIPs as its vertices, then we can estimate the location of the unknown node as the centre of the SCOP. Assuming finally we have *n* number of TIPs identified by using Equation (4.2) and these *n* numbers of TIPs are P_1, $P_{2...}P_n$. Then, unknown node N_x locates itself at the centroid of these *n* numbers of TIPs by using Equation (4.3):

$$\begin{cases} x_{CSCOP} = \left(x_{TIP1} + x_{TIP2} + ... + x_{TIPn}\right)/n \\ y_{CSCOP} = \left(y_{TIP1} + y_{TIP2} + ... + y_{TIPn}\right)/n \end{cases} \tag{4.3}$$

where (x_{TIP1}, y_{TIP1}), (x_{TIP2}, y_{TIP2}). . . (x_{TIPn}, y_{TIPn}) are coordinates of TIPs and *n* is the number of TIPs. The result x_{CSCOP} and y_{CSCOP} is the position of the unknown node, N_x.

Figure 4-5 summarizes the programme procedure of CSCOP localization algorithm.

Algorithm: Centre of the Smallest Communication Overlap Polygon (CSCOP)

```
During a period t, unknown node N_x obtains the positions of k neighbour anchor nodes (A_1, A_2 …, A_k).
(x_i, y_i) is a coordinate point, IP is Intersection Point, TIP  is True Intersection Point
x_cscop ← 0; y_cscop ← 0
for i ← 1 to (k-1)
    A_i is chosen. (x_i, y_i) is the position of A_i.
    for j ← (i+1) to k
            A_j is chosen. (x_j, y_j) is the position of A_j.
            IP[(x_i, y_i)] ← calculated as  (4.1) based on anchors A_i and A_j
            C1 = C1+1
    end for
end for
for i ← 1 to C1
     IP_i is chosen (x_i, y_i) is coordinate of IP_i
     for j← 1to k
        A_j is chosen (x_i, y_i) is the position of A_j
    TIP[(x_i, y_i)] ← calculated and checked as  (4.2) based on IPs computed
    C2 = C2 + 1
end for
for i ← 1 to C2
    do x_cscop ← ( x_cscop+TIP (x_i)) ; y_cscop ← ( y_cscop + TIP(y_i))
end for
 x_cscop ← x_cscop /C2; y_cscop ← y_cscop / C2
 return x_cscop and y_cscop
```

Figure 4-5: Procedure of CSCOP localization algorithm

To reduce computational cost of CSCOP algorithm, selective Anchor Nodes CSCOP algorithm is introduced in the following sub-section. Unlike CSCOP algorithm, this algorithm, bases only on the selected neighbour anchor nodes for estimation of unknown node. The detail of the algorithm is described in the following sub-section.

4.2 Selective Anchor Node Centre of the Smallest Communication Overlap Polygon Localization Algorithm

In this section, we introduce selective anchor node CSCOP algorithm. The core principle behind this selective anchor node CSCOP algorithm is selecting the anchor nodes which are the main contributors of the smallest communication overlap polygon (SCOP). As

CSCOP algorithm works starting with 2 neighbour anchor nodes, selective 2 and selective 3 anchor nodes CSCOP algorithms are presented respectively.

4.2.1 Selective two Anchor Nodes CSCOP Localization Algorithm

Here, we select two appropriate neighbour anchor nodes which better contribute to the formation of SCOP. In Figure 4-6, we have 3 neighbour anchor nodes of normal node N_x: A_1, A_2, and A_3. In this figure we see the SCOP (shaded area) of the 3 neighbour anchor nodes is actually formed from the communication overlap of the two neighbour anchor nodes: A_1 and A_3.

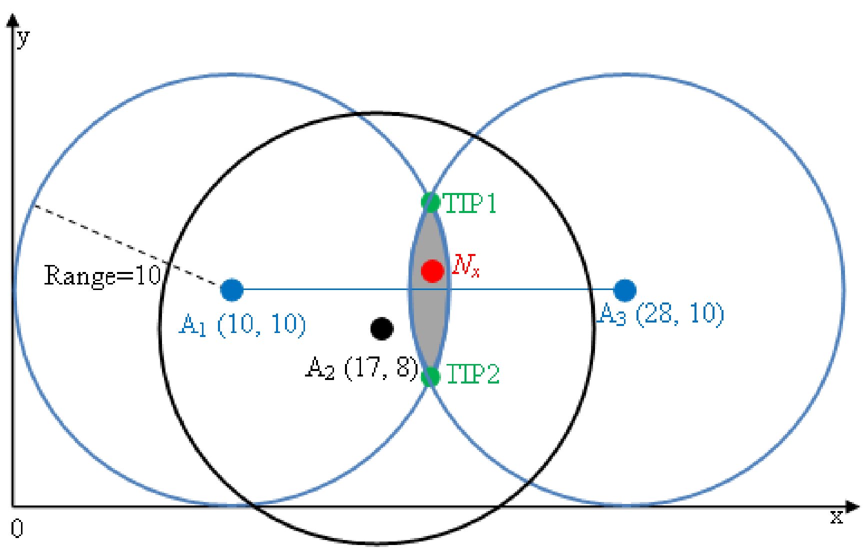


Figure 4-6: Principle of Selective two Anchor Nodes CSCOP algorithm

More distinctively, we can characterize these 2 neighbour anchor nodes as follows: they have the longest distance between them compared with any other two anchor node pairs in the entire neighbour anchor nodes distribution. In this figure, A_1 and A_3 have the longest distance between them. We can find these two anchor nodes by using Equation (4.4) below which is derived from the Pythagorean Theorem. To find the distance between two points (x_1, y_1) and (x_2, y_2), all that one needs to do is use the coordinates of these ordered pairs and apply the distance formula. If there are k neighbour anchor nodes, there will be C_2^k distances in total. By comparing these distances, the unknown node, N_x can find out the two farthest neighbour anchor nodes; for example, represented as A_i and A_j,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.4}$$

where, d is distance between two points; (x_1, y_1) and (x_2, y_2) are coordinate points of two different points.

Like other related range-free algorithms (Centroid, CPE, Mid-perpendicular, and CSCOP), Selective Anchor Node CSCOP also assumes neighbour anchor nodes have the same communication range and periodically beacon their location. To generalize, following the same steps, we can select 2 appropriate anchor nodes among k neighbour anchor nodes, when k is greater than two. Here is the programme procedure for the Selective two Anchor Nodes CSCOP localization algorithm:

Algorithm: Selective two Anchor Nodes CSCOP

Suppose N_x has k neighbour anchor nodes $A_1, A_2 \ldots, A_k$.
for $\boldsymbol{i} \leftarrow$ **1 to (*k*-1)**
    A_i is chosen. (x_i, y_i) is the position of A_i.
    for $j \leftarrow (i+1)$ **to** k
        A_j is chosen. (x_j, y_j) is the position of A_j.
      Find the farthest anchor nodes using (4.4); suppose they are A_i, A_j
end
$(x_{mid}, y_{mid}) \leftarrow$calculated using CSCOP algorithm based only on selected two anchor nodes: A_i, A_j
return x_{mid} and y_{mid}

Figure 4-7: Programme procedure of Selective two Anchor Nodes CSCOP algorithm

4.2.2 Selective three Anchor Nodes CSCOP Localization Algorithm

In Figure 4-8, we have 4 neighbour anchor nodes to unknown node, N_x: A_1, A_2, A_3 and A_4. In this figure we see the SCOP (shaded area) formed by the 4 neighbour anchor nodes is actually formed from the communication overlap of the three neighbour anchor nodes: A_1, A_2 and A_3. In general, the three farthest neighbour anchor nodes (A_1, A_2 and A_3) are the main contributors to the smallest communication overlap polygon.

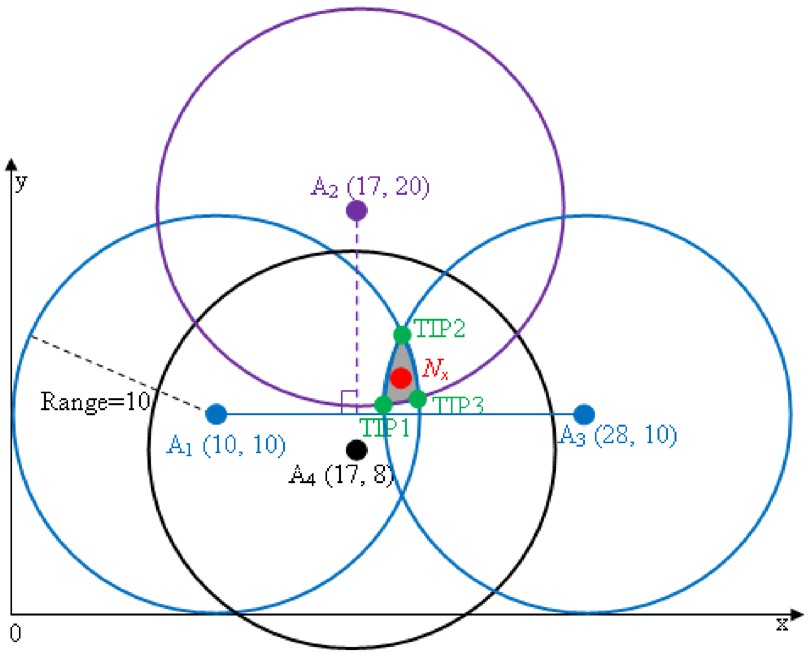


Figure 4-8: Principle of Selective three Anchor Nodes CSCOP Algorithm

Characteristically, we can characterize these 3 neighbour anchor nodes as follows: 1) the first two are identified as we have discussed in Selective two Anchor Nodes CSCOP algorithm steps. 2) The third neighbour anchor node is one with longest perpendicular distance from the line connecting the two farthest anchor nodes. In the figure, the two farthest neighbour anchor nodes are A_1 and A_3. Other anchor nodes in the figure are A_2 and A_4. Comparing their perpendicular distance to a line connecting A_1 and A_3, clearly, A_2 (with 10m distance) has a longer distance than A_4 (with 2m distance). So, the third neighbour anchor node which contributes more to the formation of the smallest communication overlap polygon is A_2. Following the same procedure, we can select the three best neighbour anchor nodes which contribute most to the smallest communication overlap polygon among *k* number of neighbour anchor nodes when *k* is greater than 3.

Here is the programme procedure of Selective three Anchor Nodes CSCOP Algorithm. In its programme procedure, first, unknown node, N_x computes the distance between any two neighbour anchor nodes using distance Equation (4.4) above.

Then, N_x looks for the third neighbour anchor node which is farthest perpendicular distanced to the line connecting A_i and A_j among all other neighbour anchor nodes except A_i and A_j using Equation (4.5). Assuming the equation of the line connecting the two farthest points is $Ax + By + C = 0$ and a point is (m, n), then, a perpendicular distance between the point and the line can be calculated using Equation (4.5), where d is distance between the line and the point.

$$d = \frac{|Am + Bn + C|}{\sqrt{A^2 + B^2}} \tag{4.5}$$

Finally, the unknown node, N_x can compute its position using CSCOP algorithm based only on the three selected neighbour anchor nodes: A_i, A_j and A_m.

Figure 4-9 below illustrates the programme procedure of Selective three Anchor Nodes CSCOP algorithm:

Algorithm: Selective three Anchor Nodes CSCOP

1. Suppose N_x has k neighbour anchor nodes $A_1, A_2 \ldots, A_k$.
2. **for** $\boldsymbol{i} \leftarrow$ **1 to (*k*-1)**
3. A_i is chosen. (x_i, y_i) is the position of A_i.
4. **for** $j \leftarrow (i+1)$ **to** k
5. A_j is chosen. (x_j, y_j) is the position of A_j.
6. Find the farthest anchor nodes using (4.4); suppose they are $A_i,\ A_j$
7. ***end***
8. ***end***
9. **for** $m \leftarrow 1$ **to** k
10. A_m is chosen. (x_i, y_i) is the position of A_m
11. Find the anchor node with farthest perpendicular distance to the line $\overline{A_i A_j}$ using (4.5); assume it is A_m
12. **end**
13. $(x_{mid}, y_{mid}) \leftarrow$calculated using CSCOP algorithm based only on selected three anchor nodes: A_i, A_j and A_m
14. **return** x_{mid} and y_{mid}

Figure 4-9: Programme procedure of Selective three Anchor Nodes CSCOP Algorithm

The related algorithms in the domain and our algorithms presented so far (CSCOP and Selective Anchor Nodes CSCOP) only work in homogeneous radio range of anchor nodes. As a result, these algorithms assume "anchor nodes have the same communication range". The question here is in the reality there are anchor nodes with different communication ranges, so how can we break this traditional assumption and have range-free geo-localization algorithm that works with heterogeneous communication range of anchor nodes. To address this problem we have introduced an algorithm which is resilient to anchor nodes' communication range difference in the following sub-section. This is a break through work in the domain knowledge.

4.3 Immune to Radio Range Difference (IRRD) Geo-localization Algorithm

Range-free local geo-localization algorithms: Centroid, Original CPE, Improved CPE, Mid-perpendicular, CSCOP and Selective CSCOP all assume "anchor nodes have the same communication ranges". To break this traditional assumption, we proposed a geo-localization algorithm which is Immune to Radio Range Difference (IRRD). This is significant breakthrough work in the domain area.

This algorithm works in the same steps as CSCOP algorithm but in a different environment. While CSCOP assumes anchor nodes have the same communication range, this algorithm breaks this traditional assumption and makes CSCOP to work with anchor nodes having homogeneous and/or heterogeneous communication ranges.

Like CSCOP localization algorithm, it works in 2 steps: In first step, it finds the SCOP where the unknown node resides; in the second step, it estimates the position of the unknown node, N_x at the centre of this pinpointed SCOP. Now, the problem is how to make CSCOP immune to anchor nodes' communication range difference.

4.3.1 Determining the Smallest Communication Overlap Polygon (SCOP)

To illustrate how this algorithm works with anchor nodes that have communication range difference, let us look at Figure 4-10 below:

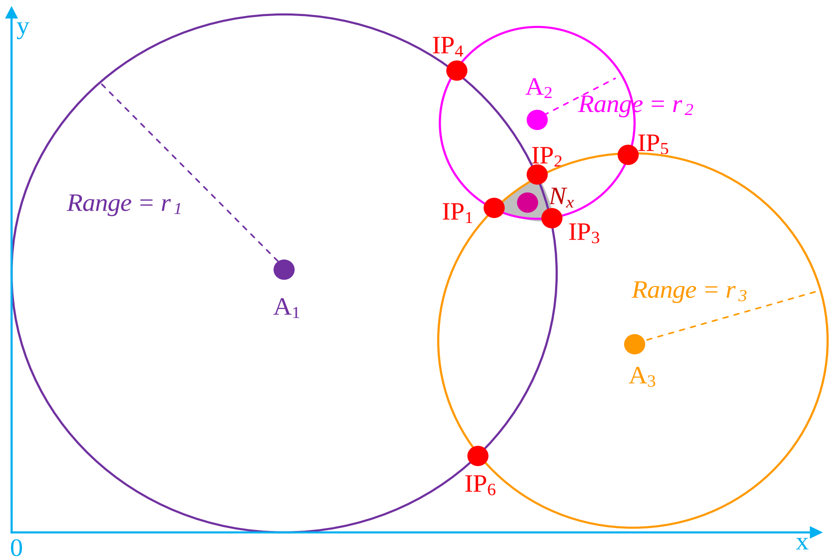


Figure 4-10: Finding Intersection Points (IPs)

In Figure 4-10, the unknown node N_x has 3 neighbour anchor nodes: (A_1, A_2, and A_3) with their different communication ranges (r_1, r_2, and r_3, respectively). Now, unknown node N_x's neighbour anchor nodes have different communication ranges (r_1, r_2, and r_3). Although unknown node's neighbour anchor nodes have different communication ranges, they are its neighbour anchor nodes. As a result, it locates inside the SCOP of these neighbour anchor nodes' (the shaded area in the figure). Hence, we have to find SCOP first.

Finding SCOP works with two steps: first, we find Intersection Points (IPs), then we find True Intersection Points (TIPs) among these IPs.

4.3.1.1 Finding Intersection Points (IPs)

The first step in pinpointing the SCOP is finding IPs. Figure 4-10 shows 3 neighbour anchor nodes for the unknown node, N_x. In other words, we have 3 circles (A_1, A_2, and A_3)

with radii r_1, r_2, and r_3, respectively. Hence, to get IPs of these circles, we solve for IPs of each circle with every other circle using Equation (4.6) C_2^n times, where (x_i, y_i) and (x_j, y_j) are the positions of the two anchor nodes involved in the calculation of IPs; r_i and r_j are the communication ranges (radii) of the two anchor nodes involved, respectively.

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 = r_i^2 \\ (x - x_j)^2 + (y - y_j)^2 = r_j^2 \end{cases} \tag{4.6}$$

IPs of any given 2 circles could be 1, 2, or infinity. If it is infinity, the two circles are identical which means they are one and the same circle. Thus, any 2 different circles intersect either at 1 point as in Figure 4-12(*a*) or at 2 points as in Figure 4-12(*b*). For instance, in Figure 4-10, if we take A_1 and A_2, they intersect at IP_3 and IP_4. In Figure 4-10, we have six IPs: IP_1, IP_2, IP_3, IP_4, IP_5, and IP_6.

More importantly, the next question is how we identify True Intersection Points (TIPs) from False Intersection Points (FIPs) among these IPs in IRRD algorithm.

4.3.1.2 Determining True Intersection Points (TIPs)

The second step in pinpointing the SCOP is identifying TIPs from FIPs. The TIPs are the vertices of SCOP. Those IPs which form vertices of SCOP are identified as TIPs whereas those which do not are False Intersection Points (FIPs) using Equation (4.7), where (x_{IPi}, y_{IPi}) and (x_j, y_j) are the co-ordinates of the IPs and anchor nodes' location, respectively; r_j^2 is array of communication range (r_1, r_2, …, r_j) of anchor nodes (A_1, A_2,…, A_j).

$$(x_{IPi} - x_j)^2 + (y_{IPi} - y_j)^2 \le r_j^2 \tag{4.7}$$

To identify TIPs, we compare the distances between each IPs to every anchor's position whether it is less than or equal to r_j (respective communication range of anchor nodes) or not. If IP's distance to each anchor is less than or equal to r_j, that IP is identified as a TIP. If not, it is regarded as a FIP. For example, as we can look at Figure 4-11, among those six IPs in Figure 4-10, IP_1, IP_2, and IP_3 are identified as TIPs whereas IP_4, IP_5, and IP_6 are considered as FIPs. Finally, we take only TIPs (TIP_1, TIP_2, and TIP_3) because they form the vertices of the SCOP.

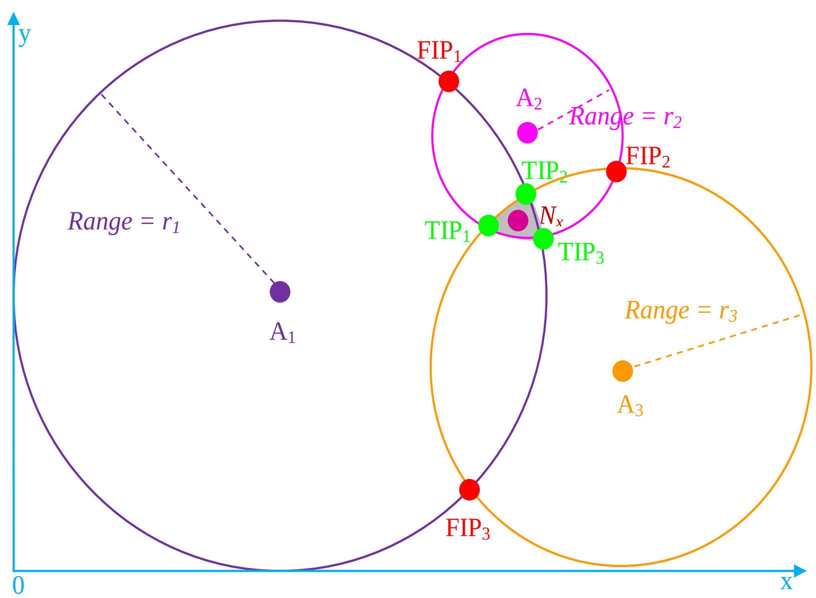


Figure 4-11: Identifying True Intersection Points (TIPs) from False Intersection Points (FIPs)

4.3.2 The Centre of the Smallest Communication Overlap Polygon

Once we identified TIPs as its vertices of SCOP, then we estimate the position of the unknown node, N_x at the centre of these TIPs (or SCOP). Assuming finally we have m number of TIPs: P_1, P_2....P_m identified by using (2), then, unknown node N_x locates itself at the centre of these TIPs using (4.8), where $(x_{TIP1}, y_{TIP1}), (x_{TIP2}, y_{TIP2}) \ldots (x_{TIPm}, y_{TIPm})$ are co-ordinates of TIPs and m is the number of TIPs. The result x_{IRRD} and y_{IRRD} is the location of the unknown node, $N_{\underline{x}}$.

$$\begin{cases} x_{IRRD} = (x_{TIP1} + x_{TIP2} + \ldots + x_{TIPm})/m \\ y_{IRRD} = (y_{TIP1} + y_{TIP2} + \ldots + y_{TIPm})/m \end{cases} \tag{4.8}$$

Let us look at the likely range of TIPs in IRRD using the following figures: Figure 4-12(*a*) confirms anchor nodes can form a single TIP which is the SCOP at the same time. This indicates the estimated and the real position of the unknown node is one and the same (0% location error) which holds true in theory but difficult to achieve in reality. On the other hand, it can be just 2 as indicated in Figure 4-12(*b*); just 3 as in Figure 4-11, or *k* as in Figure 4-12(*c*), where *k* is number of anchor nodes involved. Figure 4-12(*b*) also indicates the algorithm can work with just 2 anchor nodes.

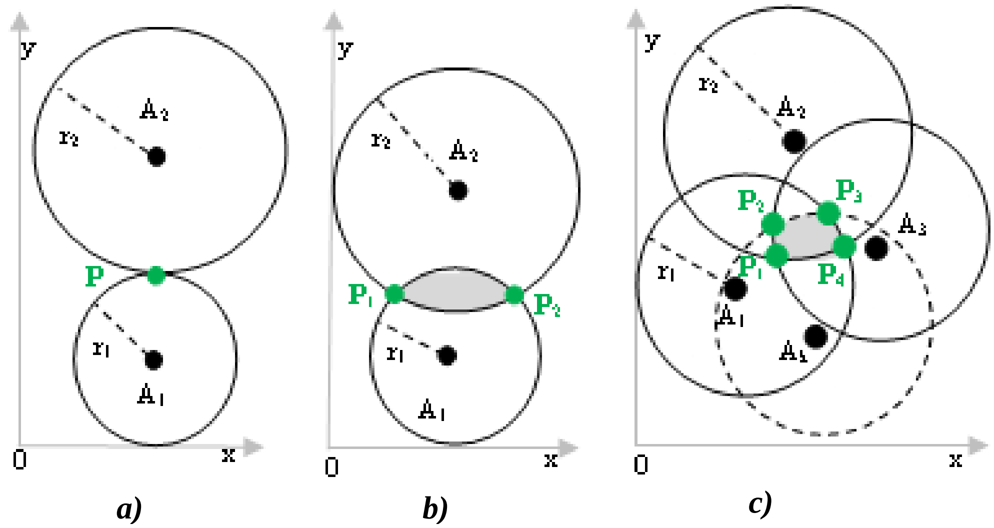


Figure 4-12: SCOP as one, two, *k* TIPs

In conclusion, unlike other related algorithms, Immune to Radio Range Difference (IRRD) algorithm as its name suggests works with both heterogeneous and homogeneous communication ranges of anchor nodes. Furthermore, this algorithm, like CSCOP also defines the smallest frontier where the unknown node resides which is the SCOP, formed from TIPs as its vertices.

Figure 4-13 summarizes the programme procedure of Immune to Radio Range Difference (IRRD) localization algorithm.

Algorithm: Immune to Radio Range Difference (IRRD) Algorithm

During a period t, unknown node N_x obtains the positions of k neighbour anchors (A_1, $A_2 \ldots, A_k$) and their communication range, r_j.
(x_i, y_i) is a coordinate point, Intersection Point (IP), True Intersection Point (TIP)
$x_{cscop} \leftarrow 0$; $y_{cscop} \leftarrow 0$
for $i \leftarrow 1$ to (k-1)
    A_i is chosen. (x_i, y_i) is the position of A_i.
    for $j \leftarrow (i+1)$ *to* k
        A_j is chosen. (x_j, y_j) is the position of A_j.
        IP[(x_i, y_i)] $\leftarrow$ calculated as (4.6) based on anchors A_i and A_j
            C1 = C1+1
    end for
end for
for $i \leftarrow 1$ *to* C1
        IP_i is chosen (x_i, y_i) is coordinate of IP_i
      for j$\leftarrow$ *1to k*
          k_j is chosen (x_i, y_i) is the position of k_j
    TIP[(x_i, y_i)] $\leftarrow$ calculated and checked as (4.7) based on IPs computed and communication range r_j
    C2 = C2 + 1
end for
for $i \leftarrow 1$ to C2
    do $x_{cscop} \leftarrow$ (x_{cscop}+TIP (x_i)) ; $y_{cscop} \leftarrow$ (y_{cscop} + TIP(y_i))
end for
$x_{cscop} \leftarrow x_{cscop}$ /C2; $y_{cscop} \leftarrow y_{cscop}$ / C2
return x_{cscop} and y_{cscop}

Figure 4-13: Programme procedure of Immune to Radio Range Difference (IRRD) geo-localization algorithm

4.4 Evaluation of Computational Complexity of the Algorithms

This sub-section analyses computational complexity of the localization algorithms: Centroid Convex Position Estimation (CPE), Mid-perpendicular, our proposed CSCOP and Selective Anchor Nodes CSCOP and IRRD.

Recent studies have shown that software has a major impact on the energy consumption of the device it is executed on [122] [123] [124] [125]. Quantifying energy consumption of hardware through sensor devices is relatively easy but analyzing software and algorithms is not because quantifying energy is hardware dependant and one has to rely on captured data [122] [123] [124] [125]. Significant efforts have been devoted to develop an algorithm power and energy consumption models in many literatures but there are no analytic models applicable and comprehensively validated yet [122] [123] [124] [125]. Hence, the generic cost analysis of algorithms is in Big O notation.

Analysis of the computational complexity of an algorithm requires determining the amount of resources (like time, storage and energy) for its execution. As direct measure of algorithm's energy consumption is highly dependent on the specific hardware circuitry, its energy cost is usually expressed in terms of computational complexity. Algorithm computational complexity is usually expressed using "Big O" notation which suppresses lower order terms and multiplicative constants [126]. For instance, if an algorithm with input size of n has at most $3n^6 + 5n$ elementary operations, its computational complexity is $O(n^6)$. This example shows that "Big O" notation suppresses lower order terms and multiplicative constants and only takes the highest exponent. However, in addition to "Big O" notation representation, we have looked at the detailed elementary computational analysis of the algorithms.

4.4.1 Computational Complexity of Centroid Algorithm

Centroid algorithm computational complexity analysis is based on its equation in [76]. In the equation, unknown node's x-axis coordinate computation requires (n-1) "+" and one "/". We compute y-axis coordinate similarly. Therefore, the detailed number of elementary operations without suppressing for Centroid algorithm is 2(n-1) "+" and 2 "/" where n is number of anchor nodes. Hence, the computational complexity of Centroid algorithm in "Big O" notation representation is $O(n)$.

4.4.2 Computational Complexity of Improved CPE Algorithm

Improved CPE algorithm computational complexity analysis is based on its equation in [113] [114]. This equation finds x-axis co-ordinate using three elementary operations: "+", "/" and "comparison". To find the maximum and minimum values among $x_1, x_2, ... x_n$ of neighbour anchor nodes, first we compare x_1 and x_2. Assuming $x_1 < x_2$ without loss of generality, temporary minimum and maximum values are set to be x_1 and x_2, respectively by first comparison. In the subsequent comparisons, we compare each x_i with temporary maximum and minimum values. If x_i is greater than temporary maximum, it is taken as temporary maximum value else it will be compared with temporary minimum value. If it is less than the temporary minimum value, it will be taken as temporary minimum. For each x_i, we need one or two comparison operations; hence, we have 3/2 average number of comparison operations. In total, the number of comparison operations required to get $\min_{i-1}^{n}(x_i)$ and $\max_{i-1}^{n}$ for Improved CPE algorithm is 1+3/2(n-2) = ½(3*n*-4). Besides these comparisons, to compute x-coordinate, Improved CPE needs one "+" and one "/" operations. We compute for y-coordinate in the same way. Thus, the total number of elementary operations needed to compute Improved CPE is (3*n* - 4) "comparison", two "+" and two "/"where *n* is number of anchor nodes. Hence, in "*Big O*" notation, the computational complexity of CPE Algorithm is the same as Centroid algorithm, which is $O(n)$.

4.4.3 Computational Complexity of Mid-perpendicular Algorithm

To analyze the computational complexity of the Mid-perpendicular localization algorithm, we base its equation in [116] [117]. Its equation has two cases. In the first case where the three neighbour anchor nodes form an acute triangle, computing x-coordinate of the unknown node requires four elementary operations: "+", "-", "*", and "/" by 4, 8, 10, and 1 times respectively. Similarly, computing y-coordinate also requires the same operation. In the first case, in total it requires 8 "+", 16 "-", 20 "*", and 2 "/". Whereas in the second case where the three anchor nodes form either a right-angled triangle or an obtuse-angled triangle, the equation demands only one "+" and one "/" elementary operation to compute x and y coordinate. As a result, the average number of the elementary operations required for the two cases when the number of anchor nodes equals three is 5 "+", 8 "-", 10 "*" and 2 "/".

When neighbour anchor nodes are greater than 3, among n neighbour anchor nodes, the authors propose to select any three of them to compute an unknown node's position. This leads us to have a total of C_3^n positions. Finally, the average of all C_3^n positions is taken as the final estimated position of unknown node, N_x. Therefore, Mid-perpendicular algorithm has $6C_3^n - 1$"+", $8C_3^n$"$-$", $10C_3^n$"$\times$", and $2C_3^n + 1$ "/" elementary operations, where n is number of anchor nodes. So, computational complexity of Mid-perpendicular algorithm is $O(n^3)$.

4.4.4 Computational Complexity of our CSCOP Algorithm

Computational complexity analysis of our CSCOP algorithm bases its equation at sub-section 4.1. Assuming we have n number of neighbour anchor nodes. First, to get IPs of these neighbour anchor nodes, we select any two of them and compute for their IPs using Equation (4.1). Here we need the following three elementary operations: 2 "+", 4 "$-$"

and 4 “*”. Totally, we have C_2^n circle pairs in solving IPs. Which means, we have $2C_2^n$ “+”, $4C_2^n$ “−”, and $4C_2^n$ “*”.

Next, we find TIPs using Equation (4.2). Here, we compare the distance between each IPs to every anchor node with communication range (R) of anchor nodes. In the algorithm, number of IPs is less than or equal to two times the number of anchor nodes. Hence, if the number of anchor nodes is n, the maximum number of IPs is $2n$. For our computational complexity analysis, we take the worst case for the number of IPs which is $2n$. In a single comparison, we have one “+”, 2 “−” and 2 “*” elementary operations. Totally, we have $2n^2$ comparisons and $2n^2$ “+”, $4n^2$ “−” and $4n^2$ “*”elementary operations in this step.

Finally, in the location estimation process using (4.3), we calculate the centre of TIPs. In this step unknown node’s x-axis coordinate computation requires (n-1) “+” and one “/”. Similarly, we compute y-axis coordinate. The number of elementary operations for this step is 2(n-1) “+” and 2 “/”.

Hence, the detailed number of elementary operations for our CSCOP algorithm is $2C_2^n$ $+2n^2+2(n-1)$ “+”, $4C_2^n+4n^2$ “−”, $4C_2^n+4n^2$ “*”, 2 “/”and $2n^2$ comparisons. Thus, computational complexity of CSCOP algorithm in “*Big O*” is $O(n^2)$, less than Mid-perpendicular algorithm’s computational complexity.

4.4.5 Computational Complexity of our Selective CSCOP Algorithms

In this sub-section we look computational complexity of our two selective anchor node CSCOP algorithms: Selective tow Anchor Nodes CSCOP and Selective three Anchor Nodes CSCOP.

4.4.5.1 Computational Complexity of our Selective two Anchor Nodes CSCOP Algorithm

Computational complexity analysis of our Selective two Anchor Nodes CSCOP algorithm bases its equation at sub-section 4.2.1. Assuming that there are *n* neighbour anchor nodes, procedure of Selective two Anchor Nodes CSCOP localization algorithm requires computing the two farthest anchor nodes among its *n* neighbour anchor nodes. This process requires comparing distance between any two anchor nodes demanding C_2^n "+", $2C_2^n$ "-", $2C_2^n$ "$\times$" and C_2^n "comparison". At last, the unknown node N_x, calculates its location using CSCOP localization algorithm based on the two selected anchor nodes. Hence, in Selective two Anchor Nodes CSCOP localization algorithm the overall number of elementary operations involved is C_2^n+3 "+", $2(C_2^n+2)$ "-", $2(C_2^n+2)$ "$\times$", 1 "/", C_2^n "Comparisons". Therefore, the computational complexity for Selective two Anchor Nodes CSCOP localization algorithm in "Big O" notation can be expressed as $O(n^2)$.

4.4.5.2 Computational Complexity of our Selective three Anchor Nodes CSCOP Algorithm

Computational complexity analysis of our Selective three Anchor Nodes CSCOP algorithm bases its equation at sub-section 4.2.2. Assuming that there are *n* neighbour anchor nodes, procedure of Selective three Anchor Nodes CSCOP localization algorithm requires computing the two farthest anchor nodes among its *n* neighbour anchor nodes.

This process requires comparing distance between any two anchor nodes demanding C_2^n "+", $2C_2^n$ "-", $2C_2^n$ "$\times$" and C_2^n "comparison". The next procedure of Selective three Anchor Nodes CSCOP localization algorithm is finding the third anchor node with longest perpendicular distance to the line connecting the two farthest anchor nodes. This procedure needs $3(n-2)$ "+", $6(n-2)$ "$\times$" $(n-2)$ "/" and $(n-3)$ "comparison". At last, the unknown node N_x, calculates its location using CSCOP localization algorithm based on the three selected anchor nodes. Hence, in Selective three Anchor Nodes CSCOP localization algorithm the overall number of elementary operations involved is (C_2^n+3n+10) "+", 2(C_2^n+9) "-", 4n+2 "$\times$", n *"/",* and *(C_2^n+n-3)* "Comparisons". Therefore, the computational complexity for Selective three Anchor Nodes CSCOP localization algorithm in "Big O" notation can be expressed as $O(n^2)$.

4.4.6 Computational Complexity of our IRRD Algorithm

Computational complexity analysis of our algorithm, IRRD, bases its equation at sub-section 4.3 and it has the same computational complexity as CSCOP algorithm because both follow the same procedure. The only additional cost is memory cost to save the array of different radio ranges of anchor nodes.

Table 4-1 summarizes the computational complexities of the algorithms discussed above both in detailed number of elementary operations required and in "*Big O*" notation. "*Big O*" notation in its algorithm complexity analysis suppresses lower order terms and multiplicative constants. To complement this we have considered a detailed number of elementary operations for the comparison.

When we compare the computational complexity of the algorithms, both Centroid and

Improved CPE algorithms have the lowest computational complexity followed by our CSCOP localization algorithm which focuses on the accuracy improvement. Mid-perpendicular algorithm has the highest computational complexity compared to others.

Table 4-1: Summary of Computational Complexity of the Algorithms

Localization Algorithms	Number of Elementary Operations	Computational Complexity
Centroid	2(n-1) "+", 2 "/"	$O(n)$
Improved CPE	2 "+", 2 "/", (3n-4) "Comparisons"	$O(n)$
Mid-perpendicular	$6C_3^n - 1$ "+", $8C_3^n - 1$ "−", $10C_3^n$ "×", and $2C_3^n - 1$ "/"	$O(n^3)$
CSCOP	2(C_2^n +n^2+(n-1)) "+", $4C_2^n$ + 4n^2 "−", $4C_2^n$ + 4n^2 "*", 2 "/" and 2n^2 "Comparisons"	$O(n^2)$
Selective two Anchors CSCOP	C_2^n +3 "+", 2(C_2^n +2) "-", 2(C_2^n +2) "×", 1 "/", C_2^n "Comparisons"	$O(n^2)$
Selective three Anchors CSCOP	(C_2^n +3n+10) "+", 2(C_2^n +9) "-", 4n+2 "×", *n "/", and (*C_2^n *+n-3)* "Comparisons"	$O(n^2)$
IRRD	2(C_2^n +n^2+(n-1)) "+", $4C_2^n$ + 4n^2 "−", $4C_2^n$ + 4n^2 "*", 2 "/" and 2n^2 "Comparisons"	$O(n^2)$

Deliberately left blank

CHAPTER 5 IMPLEMENTATIONON AND PERFORMANCE EVALUATION

This chapter first discusses the research methodology used in the implementation then evaluates and compares the accuracy performance of our geo-localization algorithms with the state-of-the-art algorithms in the domain. Performance evaluation of the algorithms is evaluated from both sparse and dense crowd network perspective as discussed in the following sub-sections:

5.1 Research Methodology

The related work review showed there are two kinds of geo-localization methods: range-based and range-free. The former is more accurate but energy hungry method which is not appropriate for battery scarce small wireless devices while the latter is less energy hungry but with coarse accuracy. As a result, this work focuses on the latter one because it consumes less energy and properly addresses the inherent short battery life problem of wireless devices.

In the review of the related work, the range-free method is further classified into two: local methods and hop counting (or DV-Hop) methods. In the local methods, an unknown node collects the location information of its neighbour anchor nodes directly to estimate its own location. On the other side, in DV-Hop method, each unknown node looks for anchor nodes to give their estimated hop sizes and then attempts to get the smallest hop count to these anchor nodes. Then, every unknown node estimates its distances to these anchor nodes using the hop count. Finally, the unknown nodes apply trilateration to estimate their location depending on the estimated distances to three appropriate neighbour anchor nodes. Assuming we can find neighbour anchor nodes for a given unknown node in the crowd, we focused on the first method.

After identifying the gaps and the shortcomings on the domain, we have tried to model the research problem mathematically and geometrically.

Then, from the conceptualized mathematical and geometrical models, we have developed our three but related novel geo-localization algorithms: i) Centre of the Smallest Communication Overlap Polygon (CSCOP), ii) Simplified CSCOP, iii) Immune to Radio Range Difference (IRRD).

These algorithms are related algorithms adding additional significant features on the original algorithm, CSCOP. From the set of Intersection Points (IPs), True Intersection Points (TIPs) which constitute the vertices of the Smallest Communication Overlap Polygon are identified leaving aside False Intersection Points (FIPs) which do not constitute the vertices of the Smallest Communication Overlap Polygon (this is discussed in detail in Chapter 4).

Finally, the developed algorithms are evaluated using MATLAB simulation in both sparse and dense crowd scenarios. The results of our algorithms are compared with other state-of-the-art algorithms in the area. In the simulation MATLAB is used with ideal scenarios: ideal radio propagation without interference or path loss, no frame collisions and no mobility for nodes. To make anchor nodes within a radio communication range (neighbour anchor nodes) of the unknown node, the simulation area is configured as a square with side length "2 × range" and the unknown node is placed at the centre of the simulation area. The radius is taken as the communication range of nodes. We assumed the unknown node N_x has communicated with its neighbour anchor nodes and knows their locations. It is also assumed that anchor nodes shared location has no error.

Using respective algorithms: Centroid [76], Improved CPE [113] [114], Mid-perpendicular [116] [117] and our algorithms, unknown node N_x computes its estimated location. The accuracy of these algorithms is quantified by the metric "location error" and "location error % radio range" where the location error is the distance between N_x's estimated position and the real position. The "location error % radio range" is obtained as the percentage of location error by the node radio range. Lower location error always indicates better accuracy; the unit is in meters.

Table 5-1 lists the general parameters used in the simulation. In the table, "Random simulation number" refers to the number of simulations with different geographic distributions of anchor nodes. Although the number of neighbour anchor nodes ranges from 2 to k for CSCOP and Selective two Anchor Nodes CSCOP and 3 to k for others, we considered a maximum of 8 anchor nodes in a scenario of sparse crowd and a maximum of 30 anchor nodes in scenario of dense crowd for our simulation purpose. To reduce subjectivity, for each number of anchor nodes, the random simulation number is set to be 5,000.

Table 5-1: General Simulation Parameters for Performance Evaluation

List of Parameters	Values
Simulation Area	40m × 40m Square Area
Node Radio Communication Range	20 metres
Radio Propagation	Ideal, no path loss, no interference
Real position of N_x	N_x(20m, 20m)
Number of Anchor nodes "*k*"	2 or more for CSCOP and Selective two Anchor Nodes CSCOP; 3 or more for others
Random Simulation Number	5,000

To have a comprehensive view on the localization accuracy performance of the algorithms, the simulation results on i) average, ii) maximum, iii) minimum location error values and iv) error probability distribution are compared and presented in tables and figures in the following sub-sections.

5.2 Performance Evaluation of Centre of the Smallest Communication Overlap Polygon (CSCOP) Localization Algorithm

5.2.1 Sparse Crowd

In this section, the accuracy performance of our localization algorithm, CSCOP, is compared with the state-of-the-art algorithms in a scenario of sparse crowd. To make sparse crowd in our simulation, anchor nodes are uniform randomly distributed 10m away from each other in a 40m by 40m square simulation area. Other simulation parameters used are those listed in Table 5.1 and described in sub-section 5.1 above.

i. Average Location Error of CSCOP in case of Sparse Crowd

Table 5-2 shows average location error of Centroid, improved CPE, Mid-perpendicular and our CSCOP algorithms in case of sparse crowd. The table shows unlike other algorithms CSCOP algorithm works starting with two anchor nodes. While Centroid algorithm performs the least followed by improved CPE and Mid-perpendicular, CSCOP algorithm out performs other algorithms, especially its performance increases significantly when the number of anchor nodes increase. For example, the average location error of our CSCOP algorithm in case of sparse crowd with 8 anchor nodes involved is 8.3552 (in % radio range) which means $\left(\frac{x}{20m} \times 100\right)$ = 8.3552, x = 1.6710m error in our simulation scenario.

Table 5-2: Average Location Error (% Radio Range) of CSCOP in Case of Sparse Crowd

No of Anchor Nodes	Centroid	Improved CPE	Midperpendicular	CSCOP
2	-	-	-	34.2183
3	31.3390	30.6425	29.4815	29.5689
4	24.5119	23.5846	22.0497	22.0055
5	20.2617	19.3202	17.8475	16.9917
6	16.8469	16.1074	14.8366	13.2046
7	14.3131	13.3125	12.6475	10.4468
8	12.9570	11.8614	11.4229	8.3552

Figure 5-1 presents the average location error performance of these algorithms in case of sparse crowd. In the figure the y-axis is average location error (% radio range) and the x-axis is number of anchor nodes involved in the localization process. The figure shows our new CSCOP algorithm out-performs other state-of-the-art algorithms in the localization accuracy. The figure also shows that unlike other algorithms it also works with two anchor nodes.

More importantly, as the number of anchor nodes increase, its performance is advanced incrementally when compared with the others. For instance, it showed high accuracy improvement in the case of more anchor involvement (5, 6, 7, and 8 anchors) than from less anchor involvement (3 and 4 anchors) because as more anchor nodes are involved, SCOP is reduced in size (or TIPs, which are vertices of SCOP, are nearer to the unknown node). For example, when we go from Figure 4-4(a) to Figure 4-4(b) in chapter 4, we notice the SCOP (shaded region) decreases as the number of anchor nodes increase from 3 to 4. This indicates CSCOP is a reliable algorithm for crowd geo-localization. In general, the figure shows the accuracy performance improvement from Centroid algorithm to our CSCOP algorithm is more significant than from Centroid to other state-of-the-art algorithms.

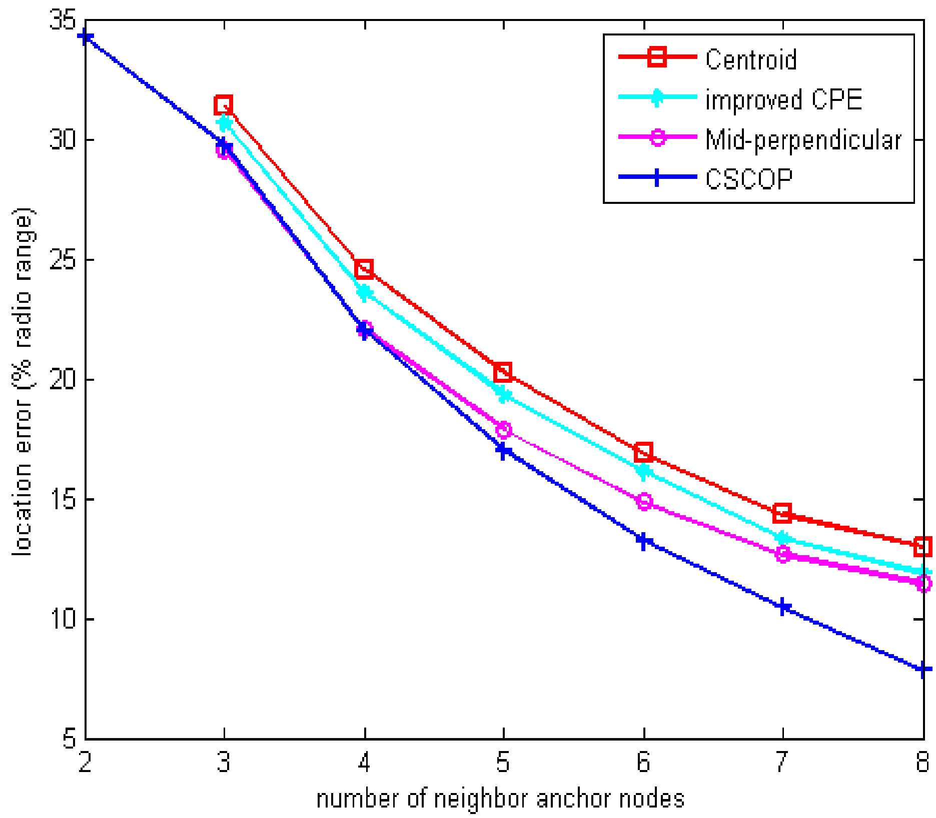


Figure 5-1: Average location error (% radio range) of CSCOP in sparse crowd

ii. Minimum-Maximum Location Error of CSCOP in Case of Sparse Crowd

The maximum and minimum location error value of these algorithms is presented in Figure 5-2. The y-axis is location error (in % radio range) and the x-axis is number of neighbour anchor nodes involved in the process of location estimation. The figure indicates the minimum location error for all four algorithms is zero or near to zero. On the other hand, the maximum location error is relatively higher; almost two times the average value. To know where most of the location error probability lies (whether around minimum, average or maximum), we need to look for error probability distribution.

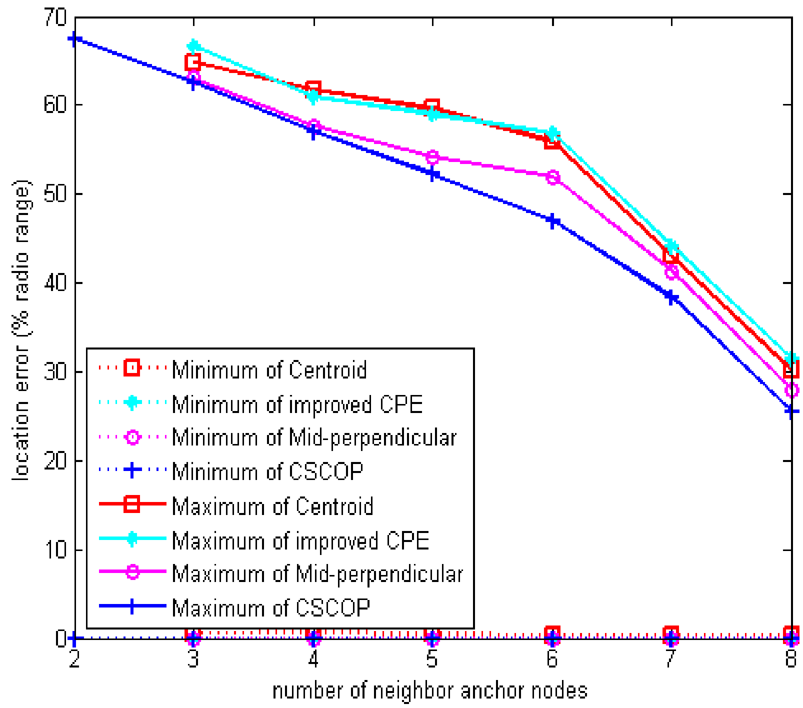


Figure 5-2: Maximum and minimum location error in (% of radio range) of CSCOP in sparse crowd

iii. Probability Distribution of CSCOP Location Error in case of Sparse Crowd

Figure 5-3 shows the location error probability distribution of the algorithms in the case of 8 anchor nodes in sparse crowd. It is generated using the MATLAB statistics toolbox normal fit function. In the figure the x-axis is location error (in % radio range) and y-axis is probability of error occurrence.

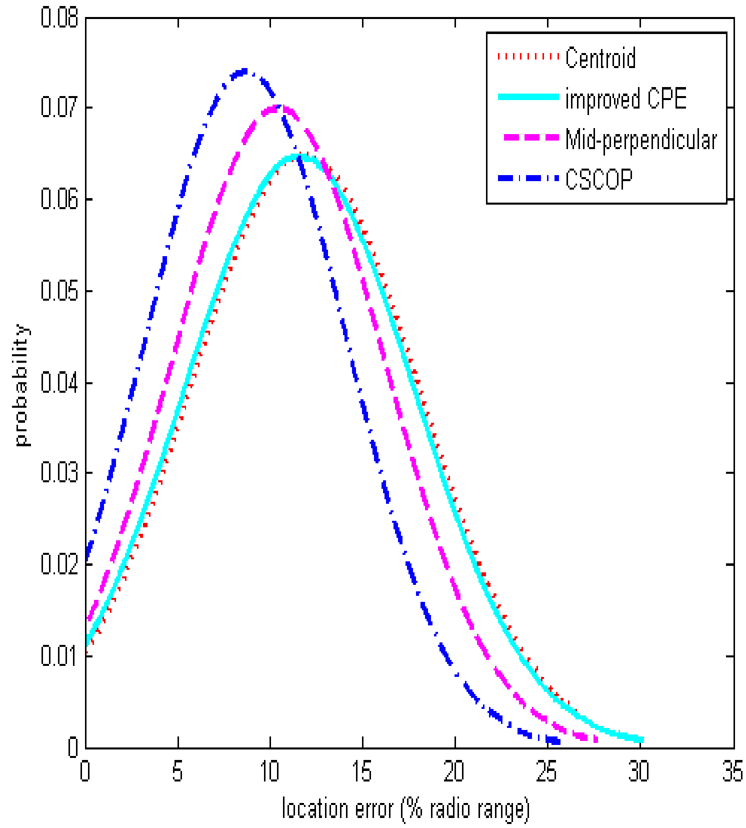


Figure 5-3: Probability distribution of CSCOP location error in sparse crowd (8 neighbour anchor nodes)

The figure indicates CSCOP has the least location error distribution and highest probability (peak of the curve) occurrence of the average error. The probability occurrence of average error values of the algorithms (when number of anchor nodes is 8) are peaks of the respective curves. When we compare these peaks, our CSCOP algorithm has the highest peak. Hence, its average error is occurring with a higher probability than others. When compared with others, it also has highest probability occurrence of minimum location error (which is zero error). Its maximum location error is occurring with a probability near zero. These make it a more reliable geo-localization algorithm when compared with others.

In general, location accuracy performance evaluation shows our algorithm (CSCOP) outperforms other state-of-the-art algorithms. This is because unlike other related algorithms, CSCOP always first pinpoints SCOP and then locates the unknown node inside the SCOP, improving accuracy, which makes it a relatively reliable and accurate localization algorithm. Our algorithm improves the accuracy since the vertices of SCOP (or TIPs) are nearer to the real position. As the number of anchor nodes increase, we get TIPs nearer to the unknown node which at the same time increases location accuracy.

Furthermore, this algorithm, in addition to improving location estimation of the unknown node N_x, also defines the smallest boundary where this node resides which is the SCOP, formed from TIPs as its vertices. This can help to define the smallest search area in search and rescue operation by having TIPs as the boundaries where to look the unknown node.

The presented algorithm is an accuracy-cost efficient localization algorithm for wireless networks. It finds many applications ranging from sensor to cellular networks; more importantly to crowd/swarm networks because when compared with other state-of-the-art algorithms, its performance increases most significantly as the number of neighbour anchor nodes increase.

5.2.2 Dense Crowd

In this sub-section, the accuracy performance of our localization algorithm, CSCOP, is compared with the state-of-the-art algorithms in a scenario of dense crowd. To make dense crowd, anchor nodes are uniform randomly distributed 2m away from each other in a 40m by 40m square simulation area. Other parameters used in the simulation are those stated in Table 5-1 and discussed in sub-section 5.1.

i) Average Location Error of CSCOP in case of Dense Crowd

Table 5-3 shows the average location error of the geo-localization algorithms in case of dense crowd. The simulation results in the table show that while Centroid localization algorithm performs least followed by Mid-perpendicular and improved CPE, our CSCOP localization algorithm performs best when compared with others in case of dense crowd. Improved CPE localization algorithm performs better than Mid-perpendicular localization algorithm in case of dense crowd unlike in case of sparse crowd where the later performs better than the former. The performance increase from Centroid algorithm to our CSCOP algorithm is more significant than from Centroid to other algorithms in both sparse and dense crowd network. For example, its location error in case of 30 neighbour anchor nodes is 3.7176 in % radio range which means $\left(\frac{x}{20m}\times 100\right)$=3.7176, x = 0.7435m error, which is much smaller error when compared with its 1.6710m error in case of dense crowd which is more than 100% increase in accuracy performance.

Table 5-3: Average Location Error (% Radio Range) of CSCOP in Case of Dense Crowd

No of Anchor Nodes	Centroid	Improved CPE	Midperpendicular	CSCOP
5	28.2199	25.5849	25.3491	24.7057
10	19.4334	16.0718	17.4321	14.2601
15	15.3683	12.0264	13.7545	9.9689
20	13.0334	9.8088	11.6695	7.5719
25	11.3366	8.3948	10.1190	5.4587
30	10.1586	7.1124	9.1874	3.7176

Figure 5-4 presents the average location error performance of these algorithms in case of dense crowd. In the figure the y-axis is average location error (% radio range) and the x-axis is number of anchor nodes involved in the process of geo-localization. The figure shows our new CSCOP algorithm out-performs other state-of-the-art algorithms.

Especially, when the number of anchor nodes increase, its performance increases more significantly than other state-of-the-art algorithms. When we compare average location error performance of our algorithm CSCOP in case of sparse crowd Figure 6-1 and dense crowd Figure 6-4, it performs better in case of dense crowd.

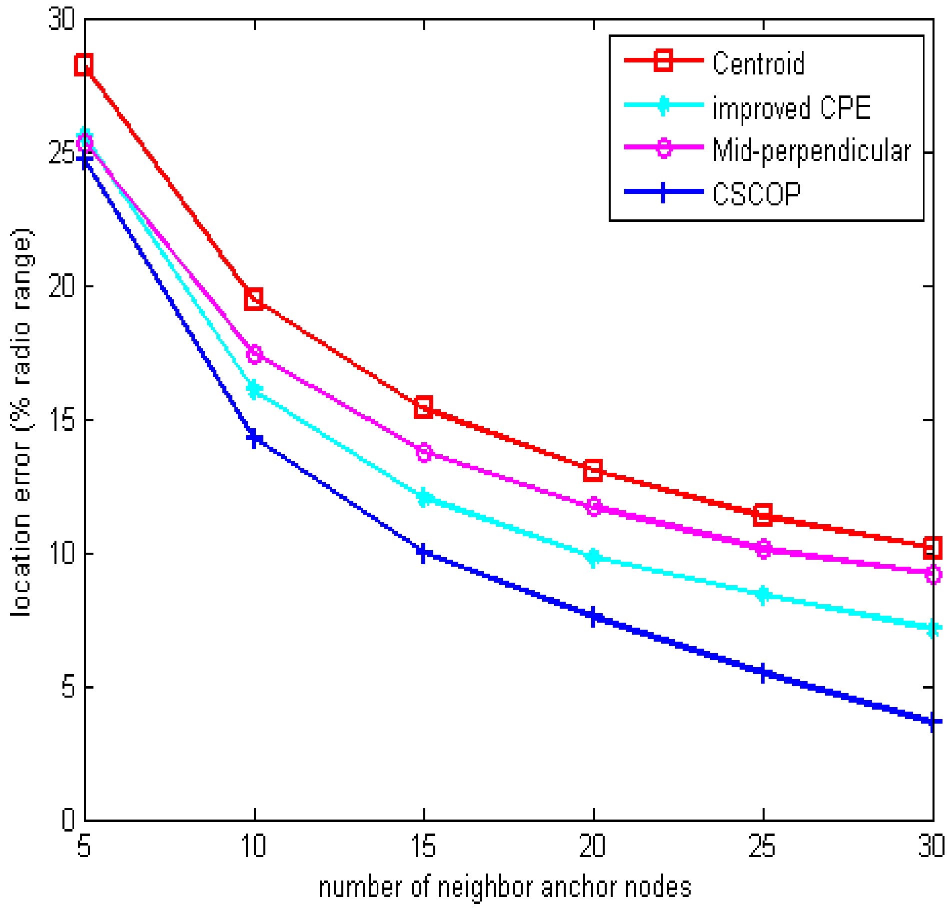


Figure 5-4: Average location error (% radio range) of CSCOP in case of dense crowd

When we compare improved CPE and Mid-perpendicular algorithms, unlike in case of sparse crowd where the later outperforms the former, the former outperforms the later in case of dense crowd. Nevertheless, our CSCOP algorithm significantly out performs other algorithms in both sparse and dense crowd scenarios.

ii) Probability Distribution of Location Error of CSCOP in Case of Dense Crowd

Figure 5-5 shows the location error probability distribution of the algorithms in the case of dense crowd with 30 anchor nodes involved. The figure is generated using the

MATLAB statistics toolbox normal fit function. In the figure the x-axis is location error (in % radio range) and y-axis is probability of error occurrence.

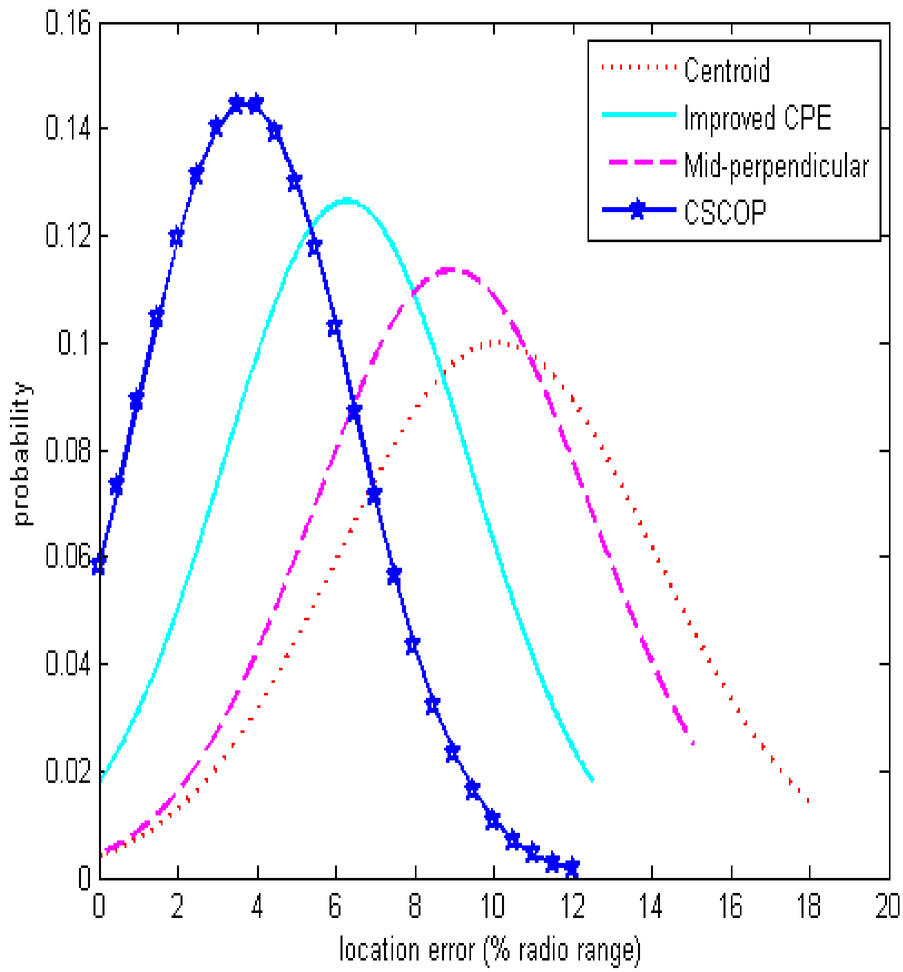


Figure 5-5: Probability distribution of location error of CSCOP in case of dense crowd (30 neighbour anchor nodes)

The figure shows CSCOP has the least location error distribution and highest probability (peak of the curve) occurrence of the average error when compared to other algorithms. The probability occurrence of average error values of the algorithms (when number of anchor nodes is 30) are peaks of the respective curves. When we compare these peaks, our CSCOP algorithm has the highest peak. Thus, its average error is occurring with a higher probability than others. The performance increase from Centroid to our CSCOP algorithm is more significant than from Centroid to other state-of-the-art algorithms.

More importantly, the figure also shows CSCOP has highest probability occurrence of minimum location error (which is zero error) when compared with others. For example, when we look at the figure, the probability of occurrence of zero error for Centroid and Mid-perpendicular algorithms is near to zero probability where as for improved CPE and for our CSCOP is near to 0.02 and 0.06, respectively. Hence, the figure shows the probability occurrence of zero error of CSCOP increases by 200% when compared to improved CPE and increases by more than 500% when compared with Centroid and Mid-perpendicular.

The figure also shows, unlike other algorithms, its maximum location error is occurring with a probability near zero. The figure in addition shows CSCOP has smallest maximum location error when compared with other algorithms. These all make it a more reliable geo-localization algorithm when compared with other state-of-the-art algorithms.

5.3 Performance Evaluation of Selective Anchor Node CSCOP Localization Algorithm

5.3.1 Sparse Crowd

In this section performance accuracy of both our Selective two and Selective three Anchor Nodes CSCOP localization algorithms are compared with the state-of-the-art algorithms (Centroid, Improved CPE, Mid-perpendicular, and CSCOP) in the case of sparse crowd. To make sparse crowd, anchor nodes are uniform randomly distributed 10m away from each other in a 40m by 40m square simulation area. Other parameters used in the simulation are those listed in Table 5.1 and sub-section 5.1 above.

To evaluate the accuracy of these localization algorithms, a comprehensive analysis of their average location error is essential. Moreover, probability distribution of location error which shows where most of the error occurs (either at average error value, maximum error value or minimum error value) has to be evaluated.

i) Average Location Error of Selective Anchor Nodes CSCOP Algorithms in Case of Sparse Crowd

Figure 5-6 compares the average location error performance of Selective two and Selective three CSCOP algorithms in sparse crowd. In the figure, y-axis reperesents the location error in percent communication radius (% radio range) of anchor nodes while x-axis refers to number of neighbour anchor nodes involved in localization of unknoen node, N_x. In the figure, Selective three Anchor Nodes CSCOP algorithm performs better than Selective two Anchor CSCOP algorithm, especially when number of anchor nodes increase.

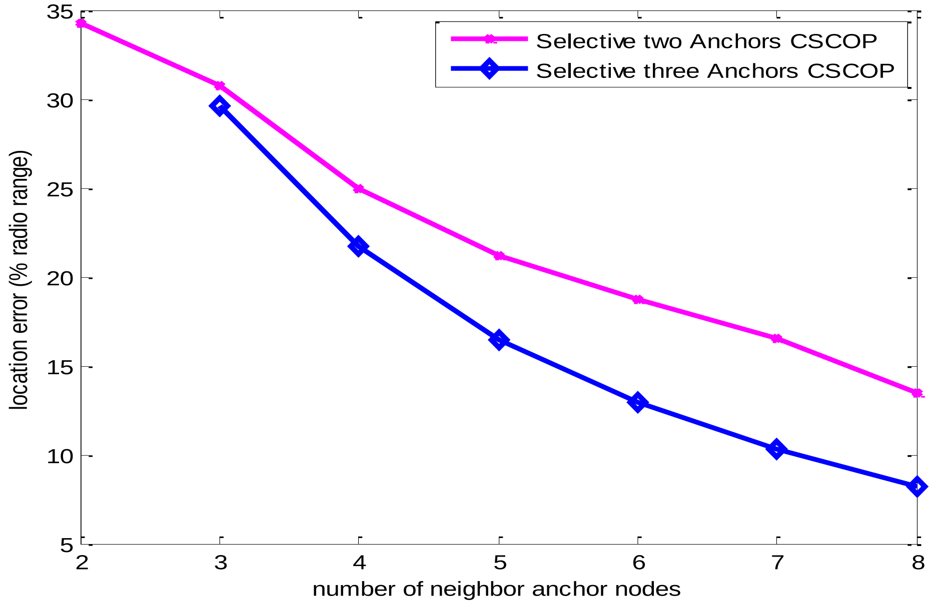


Figure 5-6: Average location error (% radio range) of Selective two and Selective three CSCOP in case of sparse crowd

Figure 5-7 shows the average location error performance of the algorithms. In the figure, Selective three Anchor Nodes CSCOP algorithm performs better than any other algorithms. It has a big difference in performance compared to Selective two Anchor Nodes CSCOP algorithm. Both CSCOP and Selective three Anchor Nodes CSCOP perform better than the others. Comparing the two; however, the latter performs slightly better than CSCOP in sparse crowd.

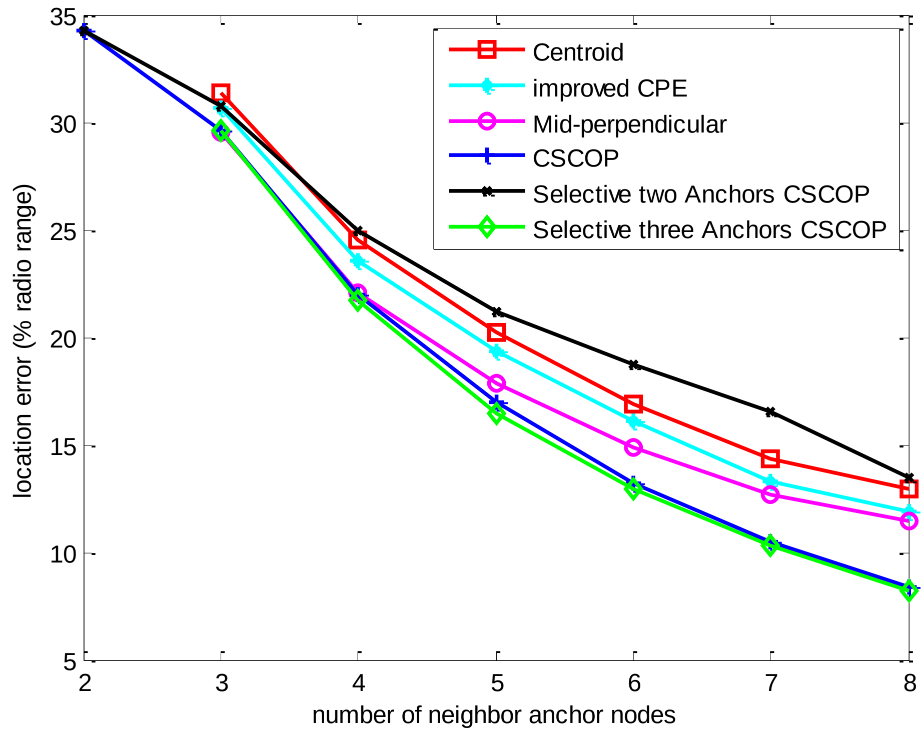


Figure 5-7: Average location error (% radio range) of Selective CSCOP in case of sparse crowd

The figure also shows unlike other algorithms which look to at least 3 anchor nodes, both Selective two Anchors CSCOP and CSCOP algorithms work starting with two anchor nodes. As the number of anchor nodes increase (5, 6, 7, 8), the performance of both CSCOP and Selective three Anchor Nodes also increases when compared with others. Comparing the two, still the latter performce slightly better.

ii) Probability Distribution of Location Error of Selective Anchor Nodes CSCOP in Case of Sparse Crowd

Figure 5-8 presents the probability distribution of location error of the algorithms. It is generated by using the MATLAB statistics toolbox normal fit function. In the figure, the x-axis is location error (in % radio range) and y-axis is probability occurrence of error.

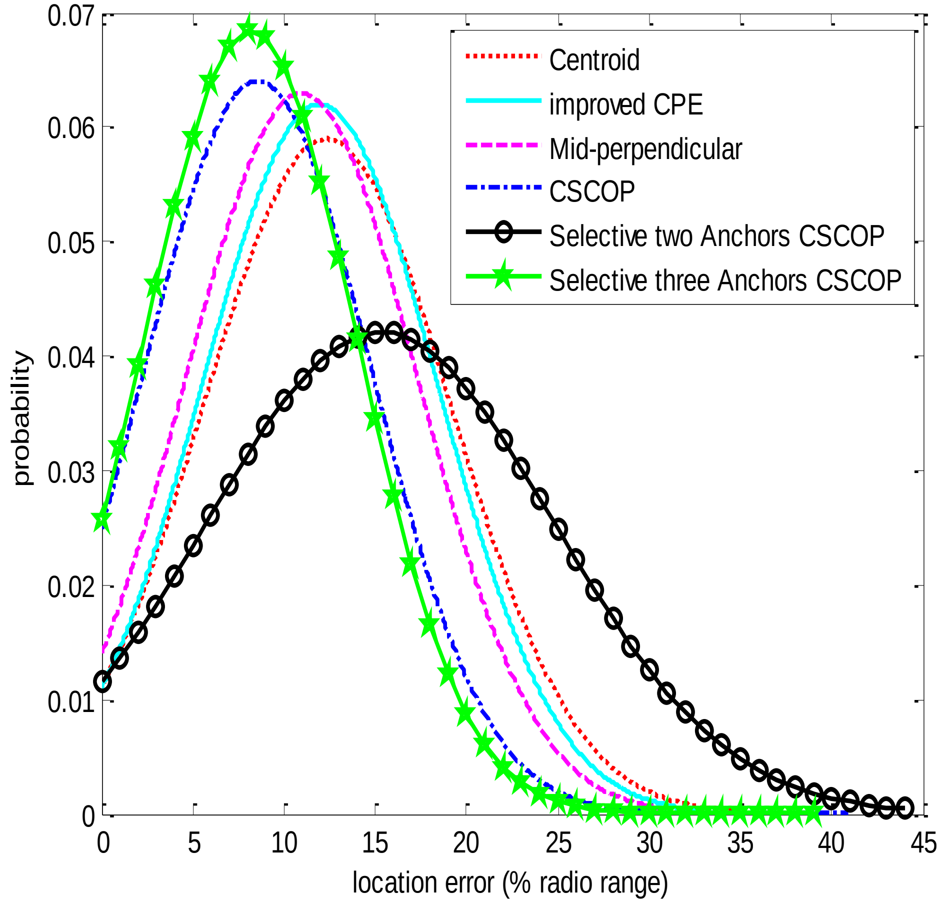


Figure 5-8: Probability distribution of location error of Selective CSCOP in case of sparse crowd (8 neighbour anchor nodes)

In the figure, Selective three Anchor Nodes CSCOP has the least location error distribution. When we compare state-of-the-art algorithms probability distribution of location error, our Selective three Anchor Nodes CSCOP has the highest probability (peak of its curve) with least location error value. The error value at this peak belongs to its average error which is occurring with highest probability while its maximum location error is occurring with a probability near to zero.

Moreover, it also has highest (equal to CSCOP) probability of minimum location error (which is zero error) making it more reliable than others in sparse crowd. Contrary to this, our Selective two Anchor nodes CSCOP has poorer performance than the others, except, in minimum location error probability occurrence.

In general, simulation results show Selective three Anchor Nodes CSCOP algorithm performs better than any other state-of-the-art algorithms in localization accuracy optimizing the trade-off between accuracy and computational cost. As the SCOP is near to the unknown node it increases accuracy. It also defines the smallest search area where the target is situated. Its centre is the estimated location of the unknown node.

5.3.2 Dense Crowd

In this section, the accuracy performance of our both Selective two and Selective three Anchor Node CSCOP localization algorithms are compared with the state-of-the-art algorithms in scenario of dense crowd network. To make dense crowd, anchor nodes are uniform randomly distributed 2m away from each other in a 40m by 40m square simulation area. Other simulation parameters used in the simulation are the same as indicated in Table 5-1.

i) *Average Location Error of Selective Anchors CSCOP Algorithm in Case of Dense Crowd*

Table 5-4 shows average location error of algorithms in case of dense crowd. The table shows our CSCOP algorithm outperforms other state-of-the-art algorithms followed by our Selective three CSCOP geo-localization algorithm. Our Selective two CSCOP performs least when compared with other algorithms. Improved CPE performs better than Mid-perpendicular algorithm in case of dense crowd unlike in case of sparse crowd where the later performs better than the former. When we arrange the performance of algorithms' average location accuracy from list to top performance in case of dense crowd is as follows: Selective two CSCOP, Centroid, Mid-perpendicular, Improved CPE, Selective three CSCOP, CSCOP.

Table 5-4: Average Location Error of Selective CSCOP in Case of Dense Crowd

No of Anchor Nodes	Centroid	Improved CPE	Mid-perpendicular	CSCOP	Selective two CSCOP	Selective three CSCOP
5	28.2199	25.5849	25.3491	24.7057	26.4328	24.0061
10	19.4334	16.0718	17.4321	14.2601	18.1927	13.3837
15	15.3683	12.0264	13.7545	9.9689	15.2434	9.6941
20	13.0334	9.8088	11.6695	7.5719	12.8481	7.6716
25	11.3366	8.3948	10.1190	5.4587	11.6720	6.3661
30	10.1586	7.1124	9.1874	3.7176	10.0889	4.6727

Figure 5-9, shows the average location error of our Selective Anchor Nodes CSCOP algorithms in relation to other algorithms. In the figure the x-axis is the number of anchor nodes involved in the geo-localization process and the y-axis is location error of algorithms (in % radio range).

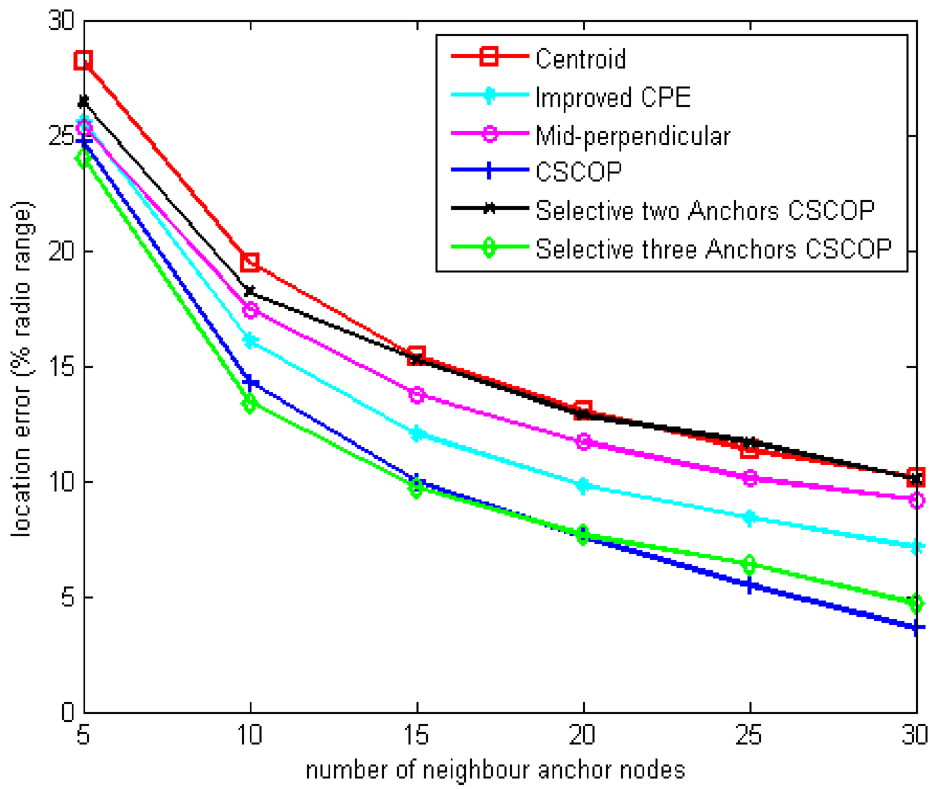


Figure 5-9: Average location error (% radio range) of Selective Anchor Nodes CSCOP in case of dense crowd

In the figure when we compare the performance of our Selective two Anchors CSCOP and Centroid algorithms, at the beginning with less number of anchor nodes our Selective two anchors CSCOP performs better but gradually as the number of anchor nodes involved increase, both perform almost the same. When we compare Mid-perpendicular and

Improved CPE algorithms, the latter outperforms the former unlike the case in sparse crowd where the former outperforms the latter. When we compare our CSCOP and Selective three CSCOP, the latter performs slightly better at the beginning but as the number of anchor nodes involved increase in number, the former performs better than the latter. Generally, when we compare all algorithms in this dense crowd scenario, our CSCOP and Selective three Anchor Nodes CSCOP outperform other algorithms.

ii) Probability Distribution of Location Error of Selective Anchors CSCOP in Case of Dense Crowd

Figure 5-10 shows probability distribution of location error of Selective CSCOP algorithms in relation with other algorithms. In the figure, x-axis is the location error (in % radio range) and y-axis is probability occurrence of location error.

The figure shows our Selective two Anchors CSCOP performs least than other algorithms but it has better probability of occurrence of zero error when compared with both Mid-perpendicular and Centroid algorithms. In the figure, our Selective three Anchors CSCOP performs considerably better than Selective two Anchors CSCOP, Centroid, Mid-perpendicular and Improved CPE algorithms but slightly less than our CSCOP algorithm.

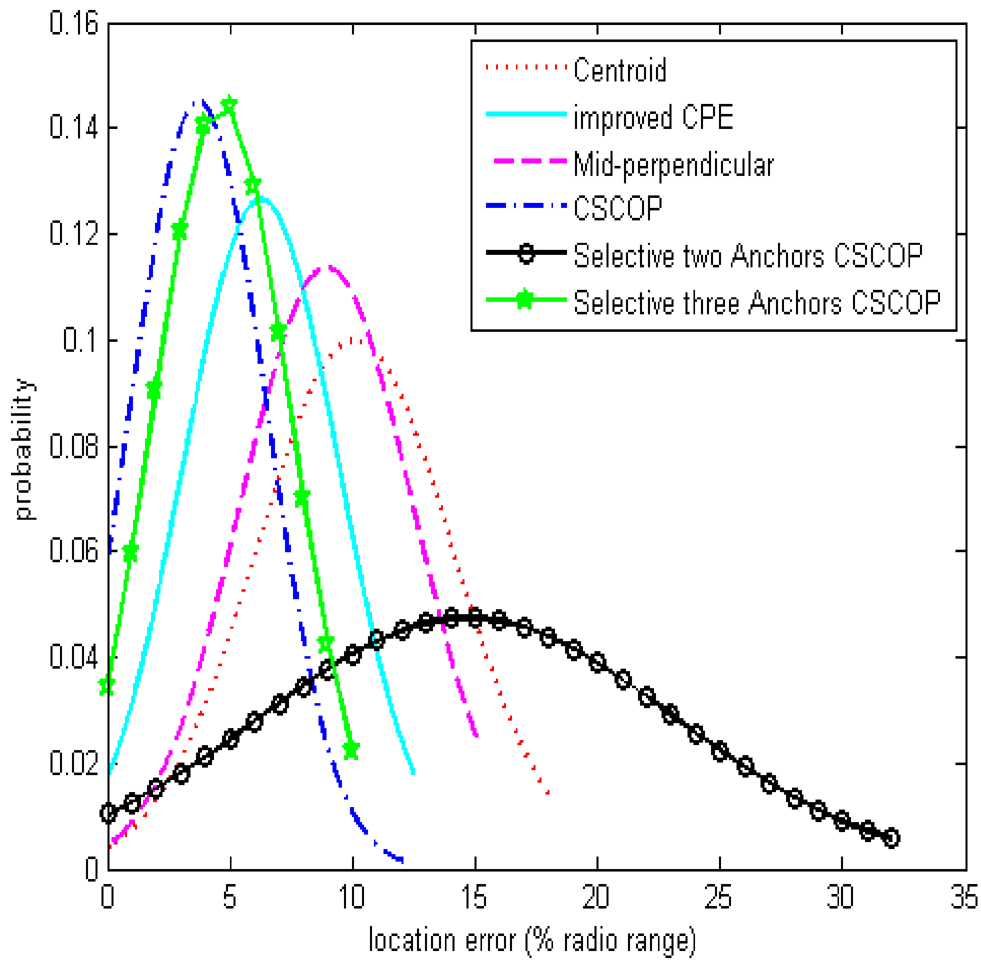


Figure 5-10: Probability distribution of location error of Selective CSCOP in case of dense crowd (30 neighbour anchor nodes)

To sum up, our CSCOP and Selective three Anchor Nodes CSCOP out performs other state-of-the-art algorithms in the area in both sparse and dense crowd networks. Especially, when the number of anchor nodes increase in number, the performance increase of these algorithms is more significant when compared with other algorithms. When we compare both; however, while the latter performs slightly better in case of sparse crowd, the former performs slightly better in case of dense crowd network but this is in trade off the computational cost. On the other hand, Selective two Anchor CSCOP performs least. When we look other algorithms performance, while Mid-perpendicular performs better than Improved CPE in case of sparse crowd, the latter performs better than the former in case of dense crowd. Therefore, our CSCOP and Selective three Anchor Nodes CSCOP are recommended algorithms in both sparse and dense crowd while the latter with additional plus advantage in reducing computational complexity by selecting only three anchor nodes and the former with better accuracy performance in case of dense crowd. CSCOP algorithm is also selective anchor node algorithm. Although it does not select only three anchor nodes like Selective three anchor Nodes CSCOP, it selects only anchor nodes which contribute to TIPs among all neighbour anchor nodes.

Deliberately left blank

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Since its introduction, the wireless network is becoming more prevalent despite its challenges. Among these challenges is geo-localization of wireless nodes because small wireless devices are battery constrained. However, in today's society location information is becoming as important as time information as LBAs increase in type and number: object tracking, smart home, hospital surveillance, environment surveillance social networking, shopping, public transport, group tour and travel, etc. As a result, problem of wireless network node localization has given significant attention by many different researchers.

Wireless network node localization solutions can be broadly classified into two: range based and range free. The difference lies whether ranging devices are required or not. A range-based method, as its name indicates, uses additional dedicated ranging devices, like timers, signal strength receivers, directional antennas and/or antenna arrays to locate nodes. This method may give a fine-grained result putting strict requirements on signal measurements and time synchronization; however, these requirements result in high power consumption of scarce battery.

On the other hand, range-free localization methods do not require additional battery hungry ranging devices for distance/angle measurements among nodes; instead they use node connectivity and appropriate range-free localization algorithms. In this method, the nodes which are aware of their locations are called anchor nodes while others are called unknown (or location unaware) nodes. Since range-free localization methods do not need energy hungry ranging devices, they do not consume high battery power which makes them more appropriate to energy constrained wireless nodes localization; however, they usually give coarse-grained accuracy. Due to their poor precision, research focuses on techniques to improve their accuracy. Range-free localization methods can be further classified into two groups: local and hop-counting (DV-Hop) methods.

In the local methods, an unknown node collects the location information of its neighbour anchor nodes directly to estimate its own location. For instance, in the Centroid algorithm which is proposed in [76], each wireless node estimates its location as the centroid of the locations of its neighbouring anchor nodes. This algorithm can give good accuracy if anchor nodes are regularly positioned [116, 117]. However, if the anchors are not positioned regularly, it gives low accuracy.

A second type of range-free localization method is hop-counting (also known as DV-Hop) method and it was first proposed by Niculescu and Nath in [118]. In this method each unknown node looks for anchor nodes to give their estimated hop sizes and then attempts to get the smallest hop count to each anchor nodes. Then, every unknown node estimates its distances to each anchor nodes using the hop count. Finally, the unknown nodes apply trilateration to estimate their location depending on the estimated distances to three appropriate nearest anchor nodes. There are several follow-up studies on DV-Hop method [119] but our focus is on the first type of method.

Compared with range-based methods, range-free methods are more cost-effective because no additional ranging devices are needed. Due to this, we focused our research on the range-free method and as we are interested in the crowd, our research focuses on the local method of range-free method.

Since the start of a local method of range-free localization research, many local range-free localization algorithms have been proposed. These include: Centroid, Convex Position Estimation (CPE), Improved CPE, Approximate Point-in-Triangulation (APIT) and Mid-perpendicular. Among these, Centroid, Improved CPE, Mid-perpendicular are well known algorithms. Nevertheless, these algorithms give coarse accuracy which does not meet location accuracy demands of different LBAs. Some are also computationally complex and costly; for example, Mid-perpendicular algorithm. Hence, we are interested in new localization algorithm which improves location accuracy and computational complexity. The following three but related novel range-free algorithms are the major contributions of this dissertation:

i) Centre of the Smallest Communication Overlap Polygon (CSCOP) algorithm:

- CSCOP's location accuracy performance outperforms other state-of-the-art algorithms in both sparse and dense crowd scenarios, is proposed. The proposed algorithm first finds the vertices of the smallest communication overlap polygon which are TIPs. Then, it estimates the position of the unknown node at the centre of TIPs. Since TIPs are close to the unknown node, it increases the accuracy of the algorithm. Besides, the simulation result shows that as the number of neighbour anchor nodes increase in number, the performance of this proposed algorithm increases more significantly than other state-of-the-art algorithms. This quality makes it appropriate localization algorithm for crowd wireless network. Thus, it is a novel way of reassessing a known problem successfully which was not addressed well by other related work.

- As CSCOP is range free localization method it did not require battery hungry ranging devices for distance/angle/time measurement, as a result, it is a battery efficient method compared with range-based methods like GPS.
- While other related algorithms require at least 3 neighbour anchor nodes to estimate the location of the unknown node, the proposed algorithm requires at least 2. This helps the unknown node to estimate its location in a less anchor node density.
- Applications of the proposed algorithm range from sensor networks to cellular networks. More specifically, in crowd wireless networks like group tour and travel, airports, stadiums, city centres, etc.
- Algorithm Computational Complexity: Both Centroid and improved CPE algorithms have relatively low computational complexity $O(n)$ level when compared with our CSCOP algorithm which increases to $O(n^2)$ level. On the other hand, Mid-perpendicular algorithm has a computational complexity as high as $O(n^3)$. Here, n is the number of neighbour anchor nodes for the unknown node.

iii) Selective Anchor Nodes and Selective three Anchor Nodes CSCOP algorithms:

- Here, we are interested in decreasing computational complexity of CSCOP by selecting appropriate anchor nodes among all neighbour anchor nodes. Then, once we select appropriate anchor nodes, we apply CSCOP algorithm only on the selected anchor nodes. In Selective two CSCOP, we select the two farthest neighbour anchor nodes using simple distance formula. These anchor nodes make the smallest overlap when compared to other pair of neighbour anchor nodes. In Selective three Anchor Nodes, we select three anchor nodes which are major contributors to the smallest communication overlap polygon than any other three neighbour anchor nodes. The first two anchor nodes are selected using the same step as Selective two Anchor Nodes algorithm. The third anchor node is perpendicularly farthest distanced node from a line connecting the two farthest nodes. Finally, we apply CSCOP algorithm on the selected anchor nodes only to estimate position of the unknown node. This decreases computational cost of CSCOP as it only applies on the selected anchor nodes instead of all neighbour anchor nodes. As there is significant difference in location precision between them, we propose Selective three Anchors CSCOP than Selective two Anchors CSCOP.

iv) **Immune to Radio Range Difference (IRRD) algorithm:**

- Here, we are interested in extending CSCOP algorithm to work in both 1) homogeneous radio communication range and 2) heterogeneous radio communication range of nodes in wireless networks unlike other related algorithms. As we have wireless nodes with different radio communication ranges, we need to have an algorithm that works in a wireless network of different radio communication ranges. This will make the proposed algorithm resilient localization algorithm for radio communication rang difference of wireless nodes; more importantly, this is a break through work in the domain since it breaks the traditional assumption: “all anchor nodes have the same radio range” and it is original contribution to the domain knowledge.

- Our algorithms (CSCOP, Selective two CSCOP and IRRD) work starting with two anchor nodes unlike other related algorithms which require at least three anchor nodes.
- To evaluate localization accuracy of these algorithms, we have conducted extensive MATLAB simulations in both sparse and dense crowd scenarios. The results show our algorithm out-perform other state-of-the-art algorithms: Cntroid, Improved CPE and Mid-perpendicular algorithms in both sparse and dense crowd geo-localization. Especially, when number of anchor nodes involved increase, our algorithms (CSCOP and Selective three Anchor Nodes CSCOP) significantly outperform other algorithms in both sparse and dense crowd. When we compare our algorithms performance in sparse and in dense crowd, they perform better in case of dense crowed. For example, the average location error of our CSCOP algorithm in case of sparse crowd with 8 anchor nodes involved is 8.3552 (in % radio range) which means $\left(\frac{x}{20m}\times 100\right)$ = 8.3552, x = 1.6710m error and its location error in case of dense crowd with 30 neighbour anchor nodes is 3.7176 (in % radio range) which means $\left(\frac{x}{20m}\times 100\right)$=3.7176, x = 0.7435m error in our simulation scenario, where communication range is 20m.
- Generally, application of our algorithms ranges from crowds of cellular networks to sensor networks.

6.2 Future Work

As follow up work, we point the following future work directions:

- In the implementation of the algorithms we used MATLAB simulation with ideal scenarios: ideal radio propagation without interference or path loss, no frame collisions, no mobility but this is not true in the reality. Moreover, local methods of range-free geo-localization algorithms: Centroid, improved CPE, Mid-perpendicular, CSCOP and Selective Anchor Nodes CSCOP assume the radio communication range of nodes' is identical and spherical. In reality, however, radio propagation could be irregular due to factors like outdoor and indoor environment, antenna, etc. As a result, it will be interesting to investigate the algorithms further in the real radio propagation scenarios.
- To obtain the performance of our algorithms in real environment it will be interesting to develop protocols and prototypes. This could help to improve further proposed algorisms.
- Since the proposed methods are implemented in no mobility metric, it will be interesting to see their performance introducing mobility of nodes.
- Local methods of range-free geo-localization algorithms: Centroid, improved CPE, Mid-perpendicular CSCOP, Selective Anchor Nodes CSCOP and IRRD work when there is neighbour anchor nodes for the unknown node where as DV-hop methods do not impose this requirement, but the later usually give low accuracy. Hence, it will be interesting to combine of both methods to optimize their short comings.

- Security: This work does not considered security issue of shared location information of users; however, location information shared has to be anonyms to the identity of the user for security. Hence, the issue of security of individuals when sharing location information is a future work direction.
- Cloud Perspective: This work has seen the problem from distributed point of view; however, as cloud usage will be common in the future; crowd geo-localization in the cloud context is also a future work direction for us.

Annexes

Annex 1: Publications

So far the following three papers from this thesis work were published in international IEEE conference proceedings. One paper for international high double standard journal IEEE/ACM is on progress receiving comments for refinement.

International Conference Proceedings

1 Demilew, S.A.; Ejigu, D.; Da-Costa, G.; Pierson, J.-M., "Novel reliable range-free geo-localization algorithm in wireless networks: Centre of the smallest communication overlap polygon (CSCOP)," in *Communications and Networking (BlackSeaCom), 2015 IEEE International Black Sea Conference on* , vol., no., pp.181-185, 18-21 May 2015
doi: 10.1109/BlackSeaCom.2015.7185111

2 S. A. Demilew, D. Ejigu, G. Da-Costa and J. M. Pierson, "Range-free selective anchor node center of the smallest communication overlap polygon localization algorithm in wireless networks," *IEEE AFRICON*, *2015*, Addis Ababa, 2015, pp. 1-5. doi: 10.1109/AFRCON.2015.7332049

3 S. A. Demilew, D. Ejigu, G. Da-Costa and J. M. Pierson, "Novel range-free immune to radio range difference (IRRD) geo-localization algorithm in wireless networks," *IEEE AFRICON*, *2015*, Addis Ababa, 2015, pp. 1-4. doi: 10.1109/AFRCON.2015.7331966

International Journals

4 S. A. Demilew, G. Da-Costa, J. Pierson, D. Ejigu, "*Energy Aware Crowd Geo-localization," International journal of IEEE/ACM Transactions on Networking, 2016. (*Submitted and received comment for refinement)

Annex 2: MatLab Source Code

```
%%%% Setting Parameters %%%%
Hsize = 40; % Size of one side in square simulation area
Crange = 20; % communication range of nodes
Dintl = 0.5; % Distance of interval of one side
Nintl = Hsize / Dintl; % divide the Hsize into Nintl intervals
Narea = Nintl^2; % Narea: number of intervals in the total area
Rnum=2000000; % times of simulations temporary
mindist = 2;
Nanc=30; % number of anchors
Pnom=[20,20]; %true position of normal node
locerr = zeros(1,6); % location error, 6 algorithms
locnum = 0; result1=[]; result2=[]; result3=[]; result4=[]; result5=
[]; result6 = [];
Rcount=1; % initial: number of simulations
while Rcount < Rnum
    Panc = zeros(Nanc,2);
    %intialization: Nanc anchors randomly distributed
    nodechosen=1:Narea;
    Rseq = randi(Narea,[1,Nanc]);
    % Nanc random numbers between [1:Narea],but not surely different
    % from each other
    for seqcount=1:Nanc % exchange, because may exist two same
                        % numbers in Rseq
        nodetemp=nodechosen(seqcount);
        nodechosen(seqcount)=nodechosen(Rseq(seqcount));
        nodechosen(Rseq(seqcount))=nodetemp;
    end
    Rseq=nodechosen(1:Nanc)'; % now sure that any two numbers in Rseq
                              % are different
    Rseq = sort(Rseq);
    Panc(:,1)=(0.5+floor((Rseq-1)/Nintl))*Dintl;
    Panc(:,2)=(0.5+mod(Rseq-1,Nintl))*Dintl;
    %whether the normal node has 3 neighbour anchors
    onehopnum =0;
    for Ncount=1:Nanc
        disanctemp = sqrt((Pnom-Panc(Ncount,1:2))*(Pnom-
Panc(Ncount,1:2))');
        if disanctemp <= Crange
            onehopnum=onehopnum+1;
        end
    end
    onehopcount1=1;
    while onehopcount1 < Nanc
        ancpos1=Panc(onehopcount1,1:2);
        onehopcount2 = onehopcount1 + 1;
        while onehopcount2 <= Nanc
            ancpos2=Panc(onehopcount2,1:2);
            distanc1to2=sqrt((ancpos1-ancpos2)*(ancpos1-ancpos2)');
            if distanc1to2 < mindist
                onehopnum = 0;
                onehopcount2=Nanc+1;
                onehopcount1=Nanc+1;
            end
            onehopcount2 = onehopcount2 + 1;
        end
        onehopcount1 = onehopcount1 + 1;
    end
```

```
%%%%%% performe all algorithms %%%%%%
%first need to detect whether there are Nanc (neighbour anchors)
if onehopnum==Nanc
    locnum=locnum+1;

    %%% Centroid Localization Algorithm %%%
    Pnomtemp=mean(Panc,1);
    errtemp=sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
    errtemp1=errtemp;
    locerr(1,1)=locerr(1,1)+errtemp;
    result1=[result1;errtemp/Crange*100];

    %%% improved CPE Localization Algorithm %%%
    CPEleft=max(Panc(:,1)); CPEright=min(Panc(:,1));
    CPEup=min((Panc(:,2))); CPEdown=max((Panc(:,2)));
    Pnomtemp=[(CPEleft+CPEright)/2, (CPEup+CPEdown)/2];
    errtemp=sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
    errtemp2=errtemp;
    locerr(1,2)=locerr(1,2)+errtemp;
    result2=[result2;errtemp/Crange*100];

    %%% Mid-perpendicular Localization Algorithm [116][117] %%%
    allresult=[];
    for onehopcount1=1:(onehopnum-2)
        ancpos1=Panc(onehopcount1,1:2);
        for onehopcount2=(onehopcount1+1):(onehopnum-1)
            ancpos2=Panc(onehopcount2,1:2);
            for onehopcount3=(onehopcount2+1):onehopnum
                ancpos3=Panc(onehopcount3,1:2);

xa=ancpos1(1);xb=ancpos2(1);xc=ancpos3(1);ya=ancpos1(2);yb=ancpos2(2)
;yc=ancpos3 (2);
                dist1=sqrt((ancpos1-ancpos2)*(ancpos1-ancpos2)');
                dist2=sqrt((ancpos2-ancpos3)*(ancpos2-ancpos3)');
                dist3=sqrt((ancpos3-ancpos1)*(ancpos3-ancpos1)');
                if dist1^2>=dist2^2+dist3^2
                    resultemp=mean([ancpos1;ancpos2],1); %
                                  %temporary result for position
                elseif dist2^2>=dist1^2+dist3^2
                    resultemp=mean([ancpos2;ancpos3],1);
                elseif dist3^2>=dist1^2+dist2^2
                    resultemp=mean([ancpos3;ancpos1],1);
                else
                    resultemp(1)=((xa^2-xb^2)*(yc-ya)+(xa^2-
xc^2)*(ya-yb)+(ya-yb)*(yb-yc)*(yc-ya))/((xa-xb)*yc+(xc-xa)*yb+(xb-
xc)*ya)/2;
                    resultemp(2)=((ya^2-yb^2)*(xc-xa)+(ya^2-
yc^2)*(xa-xb)+(xa-xb)*(xb-xc)*(xc-xa))/((ya-yb)*xc+(yc-ya)*xb+(yb-
yc)*xa)/2;
                end
                allresult=[allresult; resultemp];
            end
        end
    end
    Pnomtemp=mean(allresult,1);
    errtemp=sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
    errtemp3=errtemp;
    locerr(1,3)=locerr(1,3)+errtemp;
    result3=[result3;errtemp/Crange*100];
```

```
%%% CSCOP Localization Algorithm %%%
IP=[]; % initializing Intersection Points of anchor nodes'
       % communication circles
for ii= 1:(Nanc-1)
    c1= Panc(ii,1:2);    % center of the first circle of
                         % anchor node
    for jj=(ii+1):Nanc
        c2 = Panc(jj,1:2);  % center of the second circle of
                            % anchor node
        r1 = Crange;     % radius of the first circle
        r2 = Crange;     % radius of the second circle
        D = norm(c1-c2); % distance between circles
        cosAlpha = (r1^2+D^2-r2^2)/(2*r1*D);
        u_AB = (c2 - c1)/D; % unit vector from first to
                            % second center
        pu_AB = [u_AB(2), -u_AB(1)]; % perpendicular vector
                                     % to unit vector


     % use the cosine of alpha to calculate the length of the
     % vector along and perpendicular to AB that leads to the
     % intersection point
        intersect_1 = c1 + u_AB * (r1*cosAlpha) + pu_AB *
(r1*sqrt(1-cosAlpha^2));
        intersect_2 = c1 + u_AB * (r1*cosAlpha) - pu_AB *
(r1*sqrt(1-cosAlpha^2));
        IP = [IP;intersect_1;intersect_2];% array of
                                  %Intersection Points of anchor
                                  % nodes communication circles
    end
end

TIP = []; % intialize True Intersection Points
D = zeros(length(IP),length(Panc)); % zeros matrix of
                                    % distance between
                                    % Intersection Points to
                                    % each anchor nodes
test = zeros(length(IP),length(Panc));% zeros matrix of
                                      % logical value in
                                      %identification of TIPs
for ii = 1:length(IP)
    IPtemp = IP(ii,:);
    for jj = 1:length(Panc)
        Panctemp = Panc(jj,:);
        D(ii,jj) = norm(IPtemp-Panctemp); %Distance computed
                                          %from intersection
                                          % points to anchor
                                          % node points
        D(ii,jj) = floor(D(ii,jj));
        test(ii,jj) = D(ii,jj)<= Crange; % logical value
                                         % matrix
        if all(test(ii,:))
            TIPtemp = IP(ii,:); % True Intersection Point
                                % selected
                                % if it meets criteria
            TIP = [TIP;TIPtemp]; % Accumulates array of True
                                 % Intersection Points
        end
    end
```

```
        end
        Pnomtemp = mean(TIP,1);
        errtemp = sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
        errtemp4 = errtemp;
        locerr(1, 4) = locerr(1, 4)+ errtemp;
        result4 = [result4; errtemp4/Crange*100];


    %%% Selective two Anchor Nodes CSCOP Localization Algorithm %%%%
        longanc=zeros(1,3); % longest line connecting any two anchors
        for onehopcount1 = 1:(onehopnum-1)
            ancpos1=Panc(onehopcount1,1:2);
            for onehopcount2=(onehopcount1+1):onehopnum
                ancpos2=Panc(onehopcount2,1:2);
                distanc1to2=sqrt((ancpos1-ancpos2)*(ancpos1-
ancpos2)');
                if distanc1to2 > longanc(3)
                    longanc(3)=distanc1to2;
                    longanc(1)=onehopcount1;
                    longanc(2)=onehopcount2;
                end
            end
        end
        ancpos1=Panc(longanc(1),1:2);
        ancpos2=Panc(longanc(2),1:2);
        %%%
        %TIP=[]; % initializing Intersection Points of anchor nodes'
                 % communication circles
        c1= ancpos1;     % centre of the first circle of anchor node
        c2 = ancpos2;    % centre of the second circle of anchor node
        r1 = Crange;     % radius of the first circle
        r2 = Crange;     % radius of the second circle
        D = norm(c1-c2); % distance between circles
        cosAlpha = (r1^2+D^2-r2^2)/(2*r1*D);
        u_AB = (c2 - c1)/D; % unit vector from first to second centre
        pu_AB = [u_AB(2), -u_AB(1)]; % perpendicular vector to unit
                                     % vector

        % use the cosine of alpha to calculate the length of the
        % vector along and perpendicular to AB that leads to the
        % intersection point
        intersect_1 = c1 + u_AB * (r1*cosAlpha) + pu_AB * (r1*sqrt(1-
cosAlpha^2));
        intersect_2 = c1 + u_AB * (r1*cosAlpha) - pu_AB * (r1*sqrt(1-
cosAlpha^2));
        TIP = [intersect_1;intersect_2];% array of Intersection
                                        % Points of anchor
                                        % nodes communication circles
        Pnomtemp = mean(TIP,1);
        errtemp = sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
        errtemp5 = errtemp;
        locerr(1, 5) = locerr(1, 5)+ errtemp;
        result5 = [result5; errtemp5/Crange*100];


    %%%Selective three Anchor Nodes CSCOP Localization Algorithm%%%%
        longanc=zeros(1,3); % longest line connecting any two anchors
        for onehopcount1 = 1:(onehopnum-1)
            ancpos1=Panc(onehopcount1,1:2);
            for onehopcount2=(onehopcount1+1):onehopnum
```

```
                ancpos2=Panc(onehopcount2,1:2);
                distanc1to2=sqrt((ancpos1-ancpos2)*(ancpos1-
ancpos2)');
                if distanc1to2 > longanc(3)
                    longanc(3)=distanc1to2;
                    longanc(1)=onehopcount1;
                    longanc(2)=onehopcount2;
                end
            end
        end
        ancpos2=Panc(longanc(1),1:2);
        ancpos3=Panc(longanc(2),1:2);
        %%%
        dist2=longanc(3);
        xb=ancpos2(1); xc=ancpos3(1); yb=ancpos2(2); yc=ancpos3(2);
        longanc(3)=0;
        for onehopcount1=1:onehopnum
            if onehopcount1 ~= longanc(1) && onehopcount1 ~=
longanc(2)
                ancpos1 = Panc(onehopcount1,1:2);
                distemp = abs(ancpos1(1)*(yb-yc)+ancpos1(2)*(xc-
xb)+yc*(xb-xc)+xc*(yc-yb))/dist2;
                if distemp > longanc(3)
                    longanc(3)=distemp;
                    anctemp = onehopcount1;
                end
            end
        end
        ancpos1=Panc(anctemp,1:2);

        selectedanc=[ancpos2;ancpos3;ancpos1];

        IP=[]; % initializing Intersection Points of anchor nodes'
               % communication circles
        for ii= 1:(length(selectedanc)-1)
            c1= selectedanc(ii,1:2);    % centre of the first circle
                                        % of anchor node
            for jj=(ii+1):length(selectedanc)
                c2 = selectedanc(jj,1:2);  % centre of the second
                                           % circle of anchor node
                r1 = Crange;     % radius of the first circle
                r2 = Crange;     % radius of the second circle
                D = norm(c1-c2); % distance between circles
                cosAlpha = (r1^2+D^2-r2^2)/(2*r1*D);
                u_AB = (c2 - c1)/D; % unit vector from first to
                                    % second centre
                pu_AB = [u_AB(2), -u_AB(1)]; % perpendicular vector
                                             % to unit vector

               % use the cosine of alpha to calculate the length of the
               % vector along and perpendicular to AB that leads to the
               % intersection point
                intersect_1 = c1 + u_AB * (r1*cosAlpha) + pu_AB *
(r1*sqrt(1-cosAlpha^2));
                intersect_2 = c1 + u_AB * (r1*cosAlpha) - pu_AB *
(r1*sqrt(1-cosAlpha^2));
                IP = [IP;intersect_1;intersect_2];% array of
                % Intersection Points of anchor
                % nodes communication circles
            end
        end
```

```
        TIP = []; % initialize True Intersection Points
        D = zeros(length(IP),length(selectedanc)); % zeros matrix of
        %distance between Intersection Points to each anchor nodes
        test = zeros(length(IP),length(selectedanc));% zeros matrix
        %of logical value in identification of TIPs
        for ii = 1:length(IP)
            IPtemp = IP(ii,:);
            for jj = 1:length(selectedanc)
                Panctemp = selectedanc(jj,:);
                D(ii,jj) = norm(IPtemp-Panctemp); %Distance computed
                %from intersection points to anchor node points
                D(ii,jj) = floor(D(ii,jj));
                test(ii,jj) = D(ii,jj)<= Crange; % logical value
                                                  % matrix
                if all(test(ii,:))
                    TIPtemp = IP(ii,:); % True Intersection Point
                                        % selected
                                        % if it meets criteria
                    TIP = [TIP;TIPtemp]; % Accumulates array of True
                                         % Intersection Points
                end
            end

        end
        Pnomtemp = mean(TIP,1);
        errtemp = sqrt((Pnomtemp-Pnom)*(Pnomtemp-Pnom)');
        errtemp6 = errtemp;
        locerr(1, 6) = locerr(1, 6)+ errtemp;
        result6 = [result6; errtemp6/Crange*100];
    end

    Rcount = Rcount +1;
    if locnum == 5000
        Rcount = Rnum +1;
    end
end
if locnum~=0
    locerr=locerr/locnum;
    locerr=locerr/Crange*100; %percentage of radio range
end

%%%%% The END %%%%%
```

References

[1] "Skyhook Wireless," Accessed in March 2016. [Online]. Available: http://www. skyhookwireless.com

[2] Zhenyun Zhuang Kyu-Han, Kim Jatinder Pal Singh. "Improving Energy Efficiency of Location Sensing on Smartphones," *MobiSys'*10, June 15–18, 2010, San Francisco, California, USA.

[3] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," *in INFOCOM 2009, IEEE, April* 2009, pp. 2716 –2720.

[4] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions of Information Systems*, vol. 10, no. 1, pp. 91–102, 1992. [Online]. Available: http://dx.doi.org/ 10.1145/128756.128759 Indoor Localization and Tracking: Methods, Technologies and Research Challenges 71

[5] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications Magazine*, Vol. 4, No. 5, pp. 42-47, 1997.

[6] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proceedings of the Annual International Conference on Mobile Computing and Networking*, MOBICOM, pp. 32-43, 2000.

[7] J. J. Caffery and G. L. Stuber, "Overview of radiolocation in CDMA cellular system," *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 38–45, Apr. 1998.

[8] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, pp. 775–784, 2000. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2000.832252

[9] Y. Gwon, R. Jain, and T. Kawahara, “Robust Indoor Location Estimation of Stationary and Mobile Users,” *Proceedings of IEEE INFOCOM*, Vol. 2, pp, 2004. 1032-1043.

[10] T. S. Rappaport, *Wireless Communications, Principles and Practice*, 2nd ed. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[11] R. Fontana, E. Richley, and J. Barney, “Commercialization of an Ultra Wideband Precision Asset Location System,” *IEEE Conference on Ultra Wideband Systems and Technologies*, pp, 2003. 369-373. Galileo, http://europa.eu.int/comm/dgs/energy transport/ galileo/intro/index en.htm.

[12] Lee *et al.:* Multihop range-free localization with approximate shortest path in anisotropic wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2014. 2014:80.

[13] Garmin Ltd., “Gps Guide for Beginners,” *Technical Report Part No. 190-00224-00*, 2005.

[14] M. G. Kuhn, “An Asymmetric Security Mechanism for Navigation Signals,” *Proceedings of the Information Hiding Workshop*, 2004.

[15] R. Anderson, and M. Kuhn, “Tamper Resistancea Cautionary Note,” *Second USENIX Workshop on Electronic Commerce Proceedings*, pp. 1-11, 1996.

[16] F. van Diggelen and C. Abraham, “Indoor GPS Technology," CTIA Wireless-Agenda, Dallas, May 2001.

[17] M. Weyn and F. Schrooyen, “A WiFi-Assisted-GPS Positioning Concept, "in Proceeding of the Third European Conference on the Use of Modern Information and Communication Technologies, Gent, Belgium, March 2008.

[18] Maarten Weyn. “Opportunistic Seamless Localization.” Ph.D. dissertation, Universiteit Antwerpen te verdedigen Department of Informatica, 2011.

[19] Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. In MobiSys, 2008.

[20] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In SenSys, 2009.

[21] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In MobiSys, 2010.

[22] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, *WLAN Positioning Systems Principles and Applications in Location Based Services*. Cambridge University Press, 2012.

[23] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” IEEE Transaction of Systems, Man and

Cybernetics, Part C: Applications and Reviews, vol. 37, no. 6, pp. 1067–1080, 2007. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2007.905750

[24] R. Hansen, R. Wind, C. S. Jensen, and B. Thomsen, “Algorithmic strategies for adapting to environmental changes in 802.11 location fingerprinting,” in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Zurich, Switzerland, pp. 1–10, 2010. [Online]. Available: http://dx.doi.org/10.1109/IPIN.2010.5648270

[25] M. B. Kjærgaard, “Indoor location fingerprinting with heterogeneous clients,” Pervasive and Mobile Computing, vol. 7, no. 1, pp. 31–43, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.pmcj.2010.04.005

[26] R. Hansen, B. Thomsen, L. L. Thomsen, and F. S. Adamsen, “SmartCampusAAU - an open platform enabling indoor positioning and navigation,” in *Proceedings of 14th IEEE International Conference on Mobile Data Management*, vol. 2, Milan, Italy, pp. 33–38, 2013. [Online]. Available: http://dx.doi.org/ 10.1109/MDM.2013.62

[27] C. Laoudias, D. Zeinalipour-yazti, and C. G. Panayiotou, “Crowdsourced indoor localization for diverse devices through radiomap fusion,” in *Proceedings of 4th Intl. Conference on Indoor Positioning and Indoor Navigation (IPIN ’13)*, Montbéliard, France, 2013.

[28] Ekahau - Real-Time Location System. [Online]. Available: http://www.ekahau.com/real-time-location-system/technology/ [Accessed in January 2016].

[29] Navizon indoor location solutions. [Online]. Available: http://www.navizon.com/indoors-solutions [Accessed on February 2016]. 70 D.H. STOJANOVIĆ, N.M. STOJANOVIĆ

[30] A. K. M. M. Hossain and W.-S. Soh, "A comprehensive study of Bluetooth signal parameters for localization," in *Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, pp. 1–5, 2007. [Online]. Available: http://dx.doi.org/ 10.1109/PIMRC.2007.4394215

[31] C. Frost, C. S. Jensen, K. S. Luckow, B. Thomsen, and R. Hansen, "Bluetooth indoor positioning system using fingerprinting," in *Mobile Lightweight Wireless Systems*, vol. 81, J. Ser, E. A. Jorswieck, J. Miguez, M. Matinmikko, D. P. Palomar, S. Salcedo-Sanz, and S. Gil-Lopez, Eds., Springer Berlin Heidelberg, 2012, pp. 136–150. [Online]. Available: http://dx.doi.org/ 10.1007/978-3-642-29479-2_11

[32] ZONITH Indoor Positioning System. [Online]. Available: http://www.zonith.com/products/ips/ [Accessed on April 2015].

[33] Topaz location system. [Online]. Available: http://www.tadlys.co.il/pages/Product_content.asp? iGlobalId=2 [Accessed on April 2015].

[34] A. Cavallini (Gaia-Matrix), "iBeacons Bible 1.0." [Online]. Available: http://meetingofideas.files. wordpress.com/2013/12/ibeacons-bible-1-0.pdf [Accessed on April 2015].

[35] X. Hu, L. Cheng, and G. Zhang, A Zigbee-based localization algorithm for indoor environments, Computer Science and Network, vol. 3, pp. 1776-1781, 2011. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6182313

[36] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 13–32, 2009. [Online]. Available: http://dx.doi.org/10.1109/SURV.2009.090103

[37] J. Larranaga, L. Muguira, J.-M. Lopez-Garde, and J.-I. Vazquez, "An environment adaptive ZigBee-based indoor positioning algorithm," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Zurich, Switzerland, pp. 1–8, 2010. [Online]. Available: http://dx.doi.org/ 10.1109/IPIN.2010.5647828

[38] J. Gonzalez, J. L. Blanco, C. Galindo, A. Ortiz-de-Galisteo, J. A. Fernandez-Madrigal, F. A. Moreno, and J. L. Martinez, Combination of UWB and GPS for indoor-outdoor vehicle localization, 2007 IEEE International Symposium on Intelligent Signal Processing, pp. 1-6, 2007. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4447550

[39] S. Gezici and H. V. Poor, "Position estimation via ultra-wide-band signals," Proceeding of the IEEE, vol. 97, no. 2, pp. 386–403, 2009. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2008.2008840

[40] M. Eric, M. Dukic, and D. Vucic, "Method for direct self-localization of IR UWB node(s) in indoor scenario," in *Proceedings of IEEE International*

Conference on Ultra-Wideband (ICUWB), Bologna, Italy, pp. 570–574, 2011. [Online]. Available: http://dx.doi.org/10.1109/ICUWB.2011.6058911

[41] M. Eric, R. Zetik, and D. Vucic, “An approach for determination of antenna positions in distributed antenna system used for UWB indoor self-localization: Experimental results,” in *Proceedings of 21st Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia, pp. 204–207, 2013. [Online]. Available: http://dx.doi.org/10.1109/TELFOR.2013.6716208

[42] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra Wideband Positioning Systems Theoretical Limits Ranging Algorithms And Protocols*. Cambridge University Press, 2011.

[43] Ubisense Real-time Location System. [Online]. Available: http://www.ubisense.net/en/ [Accessed on March 2014].

[44] Time Domain’s PulsON® module. [Online]. Available: http://www.timedomain.com/ [Accessed on March 2014].

[45] H. Koyuncu and S. H. Yang, “A survey of Indoor positioning and object locating systems,” International Journal of Computer Science and Network Security, vol. 10, no. 5, pp. 121–128, 2010.

[46] F. Seco, C. Plagemann, A. R. Jimenez, and W. Burgard, “Improving RFID-based indoor positioning accuracy using Gaussian processes,” in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Zurich, Switzerland, pp. 1–8, 2010. [Online]. Available: http://dx.doi.org/ 10.1109/IPIN.2010.5647095

[47] J. Hightower and G. Borriello, "SpotON : An indoor 3D location sensing technology based on RF signal strength," UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, 2000.

[48] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," Wireless Networks, vol. 10, no. 6, pp. 701–710, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:WINE.0000044029.06344.dd

[49] T. Ito, T. Sato, K. Tulathimutte, M. Sugimoto, and H. Hashizume, "A Scalable Ultrasonic-Based Localization System Using the Phase Accordance Method," in *Recent Advances in Sensing Technology*, vol. 49, S. C. Mukhopadhyay, G. Sen Gupta, and R. Y.-M. Huang, Eds., Springer Berlin Heidelberg, pp. 35–54, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00578-7_3

[50] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "FootPath: accurate map-based indoor navigation using smartphones," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Guimaraes, Portugal, pp. 1–8, 2011. [Online]. Available: http://dx.doi.org/10.1109/IPIN.2011.6071934

[51] B.-S. Cho, W. Moon, W.-J. Seo, and K.-R. Baek, "A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding," Journal of Mechanical Science and Technology, vol. 25, no. 11, pp. 2907–2917, 2011. [Online]. Available: http://dx.doi.org/10.1007/s12206-011-0805-1

[52] B. Barshan and H. F. Durrant-Whyte, “Inertial navigation systems for mobile robots,” IEEE Transactions on Robotics and Automation, vol. 11, no. 3, pp. 328–342, 1995. [Online]. Available: http://dx.doi.org/ 10.1109/70.388775

[53] M. Hardegger, S. Mazilu, D. Caraci, F. Hess, D. Roggen, and G. Troster, “ActionSLAM on a smartphone: at-home tracking with a fully wearable system,” in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Montbéliard, France, 2013.

[54] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, “Place lab: Device positioning using radio beacons in the wild,” *in Pervasive Computing, ser. Lecture Notes in Computer Science*, 2005, vol. 3468, pp. 301–306.

[55] Priyantha, Bodhi, Dimitrios Lymberopoulos, and Jie Liu. "*Littlerock: Enabling energy-efficient continuous sensing on mobile phones.*" Pervasive Computing, IEEE 10.2 (2011): 12-15.

[56] Nath, Suman. "*ACE: exploiting correlation for energy-efficient and continuous context sensing.*" Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, 2012.

[57] K. Lin, A. Kaman, D. Lymberopoulos, and F. Zhao, “Energy-accuracy aware localization for mobile devices,” *in ACM MobySys* 2010.

[58] A. K¨upper, Location Based Services Fundamentals and Operation. John Wiley & Sons Ltd, 2005.

[59] U. Bareth and A. Kupper, “Energy-efficient position tracking in proactive location-based services for smartphone environments,” in *Computer Software*

and Applications Conference (COMPSAC), 2011 IEEE 35th Annual, July 2011, pp. 516 –521.

[60] M. C. Gonzalez, C. A. Hidalgo, and A.L. Barabasi, "Understanding individual human mobility patterns," *Nature,* vol. 453, pp. 779–782, 2012. [Online] Available: http://dx.doi.org/10.1038/nature06958

[61] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, "Entracked: energy-efficient robust position tracking for mobile devices," in *Proceedings of the 7th international conference on Mobile systems, applications, and services, ser. MobiSys* '09. New York, NY, USA: ACM, 2009, pp. 221–234. [Online]. Available: http://doi.acm.org/10.1145/1555816.1555839

[62] J. H. Winters, "On the capacity of radio communication systems with diversity in Rayleigh fading environment," *IEEE J. Sel. Areas Commun*., vol. SAC-5, pp. 871–878, Jun. 1987.

[63] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun.* Mag., vol. 42, pp. 74–80, Oct. 2004.

[64] A. Bletsas, H. Shin, and M. Z. Win, "Cooperative communications with outage-optimal opportunistic relaying," *IEEE Trans*. Wireless Commun., vol. 6, pp. 3450–3460, Sep. 2007.

[65] F. J. Gonzalez-Castano and J. Garcia-Reinoso, "Bluetooth location networks," *in Proc. IEEE Global Telecomm*. Conf., Taipei, Taiwan, R.O.C., Nov. 2002, pp. 233–237.

[66] W. Chen and X. Meng, "A cooperative localization scheme for Zigbee-based wireless sensor networks," *in Proc. IEEE Int*. Conf. Netw., Sep. 2006, vol. 2.

[67] R. Pabst, B. H. Walke, D. C. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D.

D. Falconer, and G. P. Fettweis, “Relay-based deployment concepts for wireless and mobile broadband radio,” *IEEE Commun*. Mag., vol. 42, pp. 80–89, Jan. 2004.

[68] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative Localization in Wireless Networks,” *in Proceedings of the IEEE*, Vol. 97, February 2009.

[69] J. Bachrach, and C. Taylor, “Chapter 9: “Location in Sensor Networks”. In Hand-book of Sensor Networks - Algorithms and Architectures. Edited by: Stojmenovic, I., 2005 John Wiley & Sons.

[70] V. Chandrasekhar, et al. “Localization in underwater sensor networks: survey and challenges”. In Proceedings of the 1st ACM international Workshop on Underwater Networks (WUWNet ‘06), Los Angeles, CA, USA, Sep. 25, 2006, pp. 33-40.

[71] G. Mao, B. Fidan, B. Anderson, “Wireless sensor network localization techniques”. Computer Networks, vol. 51, no. 10, Jul. 2007, pp. 2529-2553.

[72] I. Amundson, and X. Koutsoukos, “A survey on localization for mobile wireless sensor networks”. In Proceedings of the 2nd international Conference on Mobile Entity Localization and Tracking in GPS-Less Environments, Orlando, FL, USA, Sep. 30, 2009, pp. 235-254.

[73] Y. Liu, Z. Yang, X. Wang, et al. “Location, Localization, Localizability”. Journal of Computer Science and Technology, vol. 25, no. 2, Mar. 2010, pp. 247-297.

[74] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, “Global Positioning System: Theory and Practice”, Fourth Edition, Springer Wien, New York, 1997.

[75] F. Benbadis, T. Friedman, M. de Amorim, S. Fdida, “GPS-free positioning system for wireless sensor networks”. In Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks (WOCN ’05), Mar. 2005, pp. 541-545.

[76] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low-Cost Outdoor Localization for Very Small Devices, IEEE Personal Communications, 7(5) (2000) 28-34.

[77] A. Savvides, H. Park, and M. Srivastava, “The bits and flops of the n-hop multilateration primitive for node localization problems”. In Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications (WSNA ’02), Atlanta, Georgia, USA, Sep. 28, 2002, pp. 112-121.

[78] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements”. In Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems (SenSys ‘04), Baltimore, MD, USA, Nov. 03-05, 2004. pp. 50-61.

[79] Z. Yang, and Y. Liu, “Quality of Trilateration: Confidence-Based Iterative Localization”. IEEE Transactions on Parallel and Distributed Systems. vol. 21, no. 5, May 2010, pp. 631-640.

[80] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks”. In Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking (MobiCom ‘99), Seattle, Washington, USA, Aug. 15-19, 1999, pp. 263-270.

[81] C. Savarese, J. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks". In Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference, Berkeley, CA, USA, Jun. 10-15, 2002, pp. 317-327.

[82] X. Xu, N. Rao, and S. Sahni, "A computational geometry method for localization using differences of distances". ACM Transactions on Sensor Networks, vol. 6, no. 2, Feb. 2010, pp. 1-25.

[83] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors". In Proceedings of the 7th Annual International Computing and Networking (MobiCom '01), Rome, Italy, July 2001, pp. 166-179.

[84] J. Winters, "Smart antenna techniques and their application to wireless ad hoc Networks". IEEE Wireless Communications, vol. 13, no. 4, Aug. 2006, pp. 77-83.

[85] J. Hightower and G. Borriello. "A survey and taxonomy of location systems for ubiquitous computing". Technical Report UW-CSE, University of Washington, Computer Science and Engineering, Seattle, WA, USA, August 2001.

[86] D. McCrady, L. Doyle, H. Forstrom, et al. "Mobile ranging using low accuracy clocks". IEEE Transactions on Microwave Theory and Techniques, vol. 48, no. 6, Jun. 2000, pp. 951-958.

[87] S. Lanzisera, D. Lin, K. Pister, "RF time of flight ranging for wireless sensor network localization". International Workshop on Intelligent Solutions in Embedded Systems, Vienna, Austria, Jun. 2006, pp. 1-12.

[88] R. Fontana, S. Gunderson, “Ultra-wideband precision asset location system”. In Proceedings of the 2002 IEEE Conference on Ultra Wideband Systems and Technologies (ICUWB ’02), Baltimore, Maryland, USA, May 20-23, pp. 147-150.

[89] L. Lee, R. Scholtz, “Ranging in a dense multipath environment using an UWB radio link”. IEEE Journal on Selected Areas in Communications, vol. 20, no. 9, Dec. 2002, pp. 1677-1683.

[90] T. Karalar, and J. Rabaey, “An RF ToF based ranging implementation for sensor Networks”. IEEE International Conference on Communications (ICC ‘06), Istanbul, Turkey, Jun. 2006, pp. 3347-3352.

[91] M. Youssef, A. Youssef, C. Rieger, et al. “PinPoint: an asynchronous time-Based location determination system”. In Proceedings of the 4th international Conference on Mobile Systems, Applications and Services (MobiSys ‘06), Uppsala, Sweden, Jun. 19-22, 2006, pp. 165-176.

[92] M. Win, R. Scholtz, “Impulse radio: how it works”. IEEE Communications Letters, vol. 2, no. 2, Feb. 1998, pp. 36-38.

[93] E. Kaplan, “Understanding GPS: principles and applications”, Artech House, 1996.

[94] K. Pahlavan, X. Li, J. Makela, “Indoor geolocation science and technology”. IEEE Communications Magazine, vol. 40, no. 2, Feb. 2002, pp. 112-118.

[95] Y. Chraibi. Localization in wireless sensor networks. Master's thesis, Royal Institute of Technology, Sweden, November 2005.

[96] Elnahrawy, E., Li, X., Martin, R. P. “The limits of localization using signal strength: a comparative study”. In Proceedings of 1st Annual IEEE

Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 04), Santa Clara, CA, USA, Oct. 4-7, 2004, pp. 406-414.

[97] Whitehouse, K., Karlof, C., and Culler, D. “A practical evaluation of radio signal strength for ranging-based localization”. SIGMOBILE Mobile Computing and Communications Review, vol. 11, no. 1 Jan. 2007, pp. 41-52.

[98] V. Garg, “Wireless Communications & Networking”, Publisher: Morgan Kaufmann, USA, June 2007.

[99] P. Zheng and L. Ni. “Smart Phone and Next Generation Mobile Computing”. Publisher: Morgan Kaufmann, USA, 2005.

[100] Spread Spectrum Scene. “An introduction to indoor radio propagation,” http: //sssmag.com/indoor.html, June 2001.

[101] X. An, J. Wang, R.V. Prasad, and I. G. M. M. Niemegeers. “OPT: online person tracking system for context-awareness in wireless personal network”. In REALMAN '06: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality, pages 47-54, New York, USA, 2006. ACM.

[102] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, “Impact of radio irregularity on wireless sensor networks”. In Proceedings of the 2nd international Conference on Mobile Systems, Applications, and Services (MobiSys ’04), Boston, MA, USA, Jun. 06-09, 2004, pp. 125-138.

[103] D. Ganesan, B. Krishnamachari, A. Woo, et al. “Complex behavior at scale: An experimental study of low-power wireless sensor networks”. Technical Report UCLA/CSD-TR 02- 0013, UCLA Computer Science, 2002.

[104] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, “Understanding the causes of packet delivery success and failure in dense wireless sensor networks”. In Proceedings of the 4th international Conference on Embedded Networked Sensor Systems (SenSys ‘06), Boulder, Colorado, USA, Oct. 31 - Nov. 03, 2006, pp. 419-420.

[105] E. Miluzzo, X. Zheng, K. Fodor, and A. Campbell, “Radio characterization of 802.15.4 and its impact on the design of mobile sensor networks”. In Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN ‘08), Bologna, Italy, Jan. 30-Feb. 01, 2008, pp. 171-188.

[106] Kevin Fall et al. “Chapter 18: Radio propagation models”. The NS manual, May 2009. Available at http://www.isi.edu/nsnam/ns/doc/index.html.

[107] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, “The effects of ranging noise on multihop localization: an empirical study”. In Proceedings of the 4th international Symposium on information Processing in Sensor Networks (IPSN ‘05), Los Angeles, California,USA, Apr. 24-27, 2005, pp. 73-80

[108] J. A. Palazon, M. Sepulcre, J. Gozalvez, G. Prieto, "Experimental RSSI-based Localization System using Wireless Sensor Networks", Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'12), September 2012.

[109] Widyawan, M. Klepal, and D. Pesch. “Influence of predicted and measured fingerprint on the accuracy of RSSI-based indoor location systems. In Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC 2007), Hannover, Germany, March 2007.

[110] R. Dalce, T. Val, A. Van den Bossche. "Comparison of Indoor Localization Systems based on Wireless Communications." Wireless Engineering and Technology, Scientific Research Publishing, Irvine - USA, vol. 2, no. 4, octobre 2011.

[111] F. Reichbach and D. Timmermann, "Indoor Localization with Low Complexity in Wireless Sensor Networks," IEEE International Conference on Industrial Informatics, Singapore, August 2006, pp. 1018-1023.

[112] L. Doherty, K.S.J. Pister, and L.E. Ghaoui. "Convex position estimation in wireless sensor networks", in: Proceedings of IEEE INFOCOM '01, Alaska, USA, Apr.2001, vol. 3, pp. 1655- 1663.

[113] Jang-Ping Sheu; Jian-Ming Li; Chih-Shun Hsu, "A Distributed Location Estimating Algorithm for Wireless Sensor Networks", IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), vol.1, pp.1-8, June 2006.

[114] J.P. Sheu, P.C. Chen, and C.S. Hsu. A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement, IEEE Transactions on Mobile Computing, 7(9) (2008), pp. 1110-1123.

[115] T. He, C. Huang, B. Blum, et al. "Range-free localization schemes for large scale sensor networks". In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, CA, USA, Sep. 14-19, 2003, pp. 81-95.

[116] L. Gui, A. Wei, T. Val, "A two-level range-free localization algorithm for wireless sensor networks," IEEE Conference on Wireless Communications Networking and Mobile Computing, pp. 1- 4, Chengdu, September 2010.

[117] L. Gui, T. Val, A. Wei, "A Novel Two-Class Localization Algorithm in Wireless Sensor Networks", Network Protocols and Algorithms, vol. 3, no. 3 (2011).

[118] D. Niculescu, and B. Nath, "DV based Positioning in ad hoc networks", Journal of Telecommunication Systems, 22(1/4) (2003), pp. 267-280.

[119] S. Hou, X. Zhou, and X. Liu, "A novel DV-hop localization algorithm for asymmetry distributed wireless sensor networks," 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 4, pp. 243-248, 2010.

[120] Z. Zhang, G. Xu, Y. Li, and S. Huang, "DV-hop based self-adaptive positioning in wireless sensor networks," IEEE Conference on Wireless Communications Networking and Mobile Computing, pp. 1-4, 2009.

[121] J. Lee, W. Chung, E. Kim, and I. Hong, "Robust DV-hop algorithm for localization in wireless sensor network", International Conference on Control Automation and Systems, pp. 2506- 2509, 2010.

[122] N. Mishra, H. Zhang, J. D. La_erty, and H. Ho_mann, "A probabilistic graphical model based approach for minimizing energy under performance constraints," *in Procs of the Twentieth Int Conf on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS 15, 2015, pp. 267-281.

[123] P. Alonso, M. F. Dolz, R. Mayo, and E. S. Quintana-Orti, "Modelling power and energy consumption of dense matrix factorizations on multicore processors," *Concurrency Computat.*, 2014.

[124] S. Roy, A. Rudra, and A. Verma, “An energy complexity model for algorithms," *in Procs of the 4th Conf on Innovations in Theoretical Computer Science,* ser. ITCS '13, 2013, pp. 283-304.

[125] V. A. Korthikanti and G. Agha, “Towards optimizing energy costs of algorithms for shared memory architectures," *in Procs of the Twenty-second Annual ACM Symp on Parallelism in Algorithms and Architectures*, ser. SPAA '10, 2010, pp. 157-165.

[126] A. Sanjeev, B. Boaz, “Computational Complexity: A Modern Approach”. Published by Cambridge, 2009.

Declaration

I hereby declare that the material contained in this dissertation has not been previously submitted for a degree in this or in any other university. I farther declare that this dissertation is sonly based on my own research.

I also declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I understand that my dissertation may be made electronically available to the general public for reference purpose.

21/03/2017

Samuel Asferaw Demilew