# TOWARDS ENHANCED LOCATION AND SENSING SERVICES FOR THE INTERNET OF THINGS

by

YASER AL MTAWA

A thesis submitted to the School of Computing

In conformity with the requirements for

the degree of Doctor of Philosophy

Queen's University

Kingston, Ontario, Canada

April, 2017

# Abstract

Location discovery (i.e., Localization) and sensing coverage services in Wireless Sensor Networks (WSNs) have received significant attention from the Internet of Things (IoT) research community. The usage of WSNs within IoT mandates taking into account IoT characteristics when considering sensing coverage. These characteristics include heterogeneity, large scale, dynamicity, and multiple ownership. Anchors are typically used to enable localization in IoT settings. Anchor misplacement or errors in anchor location readings can cause significant disruption to location-based services in IoT. This thesis investigates the anchor misplacement problem, provides an analytical study of both localization, and sensing coverage under the presence of anchor misplacement.

We utilize two tools from computational geometry Voronoi Diagram (VD) and Delaunay Triangulation (DT) to partition the target region in order to make the problem solvable and easy to follow. We also borrow a graph-theoretic tool called Graph History to more closely understand the impact of anchor misplacement on sensing coverage. These tools allow us to locally study, analyze, and detect the impact of anchor misplacement in its vicinity. We analyze the problem of anchor misplacement, its impact on localization and sensing coverage, and we also identify new types of sensing coverage holes. We also present heuristics to mitigate the impact of anchor misplacement and improve the reliability and accuracy of WSN services. Our research approach and solution for the anchor misplacement problem can be utilized in a multiplicity of localization and sensing coverage applications regardless of the sensors or deployment types including IoT. Results show that our proposed algorithms are far more conducive to IoT context. They provide higher detection rates of misplaced anchors and sensing coverage holes, and more effective mitigation which result in higher enhancement of IoT services.

# Acknowledgements

All thanks go to Almighty Allah, the most gracious, the most merciful, for giving me determination, strength and patience to complete this work.

I would like to express my sincere gratitude to my advisors Dr. Hossam S. Hassanien and Dr. Nidal Nasser for their support, guidance, and patience. Their continuous guidance and motivation were reinforcing the knowledge of my PhD research. I appreciate the consistently excellent mentoring of my supervisors. They have exceptional skills in dealing with their students; I have ever had the pleasure and knowledge together before dealing with both of them. They guide their students to be methodical and scientists of the future. In four years of being their PhD student, I am confident that they are among the best representatives of outstanding supervisors at Queen's University.

Besides my advisors, I would like to thank the rest of my supervisory committee: Dr. Kai Salomaa, and Dr. Robin Dawes for their detailed comments and insightful feedback. Their questions expanded my knowledge from different perspectives.

I would like also to thank Basia Palmer, our research assistant at TRL, for her comments and endless support. You efforts to support me even prior my arrival to Canada are much appreciated.

I thank my wonderful lab mates for the brainstorming discussions, hard working together before deadlines, and for all jokes and fun we have had at TRL.

My sincere thanks goes to my family: my parents, brothers, and sisters for their continuous support and encouragement. You were and will always be a huge inspiration in my life.

Last but not least, I would like to thank my lovely wife, Sabah. Her love, encouragement, and dedication light the path throughout my PhD. Without the support and patience of my wife and my children, Ammar and Mona, I would never have completed my PhD. Although they had to endure my absence, they rarely complained. I hope the completion of my PhD will be a source of joy and happiness for them.

# Statement of Originality

I hereby certify that all of the work described within this thesis is the original work of the author. Any published (or unpublished) ideas and/or techniques from the work of others are fully acknowledged in accordance with the standard referencing practices.

Yaser Al Mtawa

April, 2017

# List of Acronyms

| | |
|---|---|
| ADC | Analogue to Digital Convertor |
| AoA | Angle of Arrival |
| APIT | Approximate Point in Triangulation |
| BS(s) | Base Station(s) |
| CHRAT | Coverage Hole: Ratio And Type |
| DAnD | Distributed Anchor Detection |
| DT | Delaunay Triangulation |
| DV | Distance Vector |
| DV-hop | Distance Vector based on hops |
| GPS | Global Positioning System |
| ITC | Intra-triangle Coverage |
| IoT | Internet of Things |
| IR | Infra-Red |
| IRAT | Identify Ratio And Type |
| LANDMARC | Location Identification based on Dynamic Active RFID Calibration |
| LoS | Line of Sight |
| LS-WSN(s) | Large-Scale Wireless Sensor Network(s) |
| LUB | Lower and Upper Bounds |
| MMSE | Minimum Mean Square Estimate |
| NLoS | Non-Line of Sight |
| PIaT | Point in a Triangle |
| RF | Radio Frequency |

| | |
|---|---|
| RFID | Radio Frequency Identification |
| RGG | Random Geometric Graph |
| RMSD | Root Mean Square Distance |
| RN(s) | Relay Node(s) |
| RSS | Received Signal Strength |
| RSSI | Received Signal Strength Indicator |
| SN(s) | Sensor Node(s) |
| SS | Signal Strength |
| TDoA | Time Difference of Arrival |
| ToA | Time of Arrival |
| TVSR | Testing Validity of the Sensing Report |
| UWB | Ultra-Wide Band |
| VD | Voronoi Diagram |
| WSN(s) | Wireless Sensor Network(s) |

# Co-Authorship

[1] Y. Al Mtawa, H. Hassanein, and N. Nasser, " Localization of IoT Sensors under Anchor Misplacement," *IEEE Sensors Journal* (Submitted)

[2] Y. Al Mtawa, H. Hassanein, and N. Nasser, "Sensing Coverage in IoT Deployment under the Presence of Anchor Misplacement," *IEEE Internet of Things Journal* (Submitted)

[3] Y. Al Mtawa, H. Hassanein, and N. Nasser, "Measuring the Validity of Sensing Coverage in the Presence of Anchor Misplacement," *IEEE International Conference on Communications* (*ICC*), May 2017. (Accepted)

[4] Y. Al Mtawa, H. Hassanein, and N. Nasser, "The Impact of Anchor Misplacement on Sensing Coverage," *IEEE Wireless Communications and Networking Conference* (*WCNC*), September 2016.

[5] Y. Al Mtawa, H. Hassanein, and N. Nasser, "Identifying Bounds on Sensing Coverage Holes in IoT Deployments," *IEEE Global Communications Conference* (*GLOBECOM*), December 2015.

[6] Y. Al Mtawa, N. Nasser, and H. Hassanein, "Mitigating Anchor Misplacement Errors in Wireless Sensor Networks," *IEEE International Wireless Communications and Mobile Computing Conference* (*IWCMC*), August 2015.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The Internet of Things (IoT) is a large-scale network of many physical objects that can be equipped with sensors, software, and connectivity to enable these "sensing objects"[1] which may belong to multiple operators/owners to communicate and exchange data with each other [1]. There are several components and enabling technologies of the IoT. Among which are Wireless Sensor Networks (WSNs), Infra-Red (IR), Radio Frequency Identification (RFID), Bluetooth and cellular networks. The services provided by a WSN such as sensing coverage and location discovery have received significant attention from the IoT research community [2]–[5]. The efficient utilization of these services under the umbrella of IoT mandates taking IoT characteristics into consideration. These characteristics include large scale, heterogeneity, dynamicity and multiple ownerships. Inaccurate location reporting resulting from localization errors affects the quality of services of WSNs. Localization errors also results in poor sensing coverage of objects. The gaps in sensing coverage between objects degrades the sensing quality. These two services are strongly affected by anchor misplacement. Anchor nodes are nodes in the network with known locations. They are usually equipped with Global Positioning System (GPS) [6] or placed in known position coordinates in the target field. Therefore, most of the studies assume accurate locations of anchor nodes, which is not always the case. For instance, environmental factors such as the wind, rain, water current, soil erosion, besides natural factors such as wildlife disturbing the terrain are all causes of anchor misplacement. Furthermore, there is always an inherent uncertainty in anchor node's location even with GPS-equipped anchor nodes due to erroneous measurements and

[1] In this thesis, we use the terms "sensor node," and "sensing object" interchangeably.

calculations. Therefore, both IoT location discovery and sensing coverage services are affected by anchor misplacement.

## 1.1 Motivations

The vision behind IoT is to construct a large-scale, coherent, unified framework from different building technology blocks. It is predicted that IoT will consist between 15-20 billion smart objects by 2020 [7].

WSN is one of the main enabling technologies for IoT. Therefore, enhancing the location and coverage services of WSNs has a direct impact on the realization of IoT. Anchor misplacement is a problem that needs to be investigated to have reliable WSNs services. For example, consider heterogeneous temperature sensors that are deployed in a region and belong to three different sensing providers as shown in Figure 1.1. These collective sensors can be viewed as shared resources, and their cooperation can provide a better quality of service. Usage of such shared resources can be further enhanced with participatory sensors (as in smartphones). The main challenge in such IoT setting is determining sensing coverage, and detecting coverage holes if any. Non-deterministic sensing node deployment often makes coverage holes inevitable even in a high-density network. Studying this problem with the above provided IoT settings (see Figure 1.1) and under anchor misplacement is needed for the research community to obtain more solid foundation about the essence of this new type of error. This allows us to enhance the results and make them more reliable.

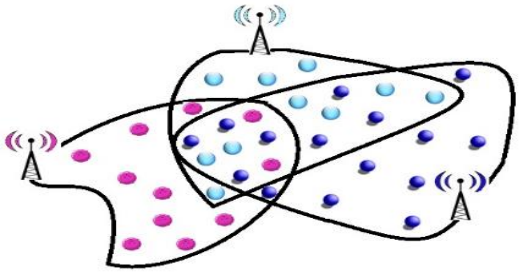


**Figure 1.1: Multiple IoT sensing providers.**

## 1.2 Thesis contribution

This thesis provides more insight about error theory in the field of IoT; it answers the following research questions:

1) Can we design a distributed scheme to discover sensing coverage holes in WSNs?
2) How does anchor misplacement affect localization in WSNs?
3) Does anchor misplacement affect sensing coverage in WSNs?
    - How to determine the actual coverage holes that have been undetected because of anchor misplacement?
    - How to determine the perceived coverage holes that have been generated by anchor misplacement?
4) How to measure the validity of sensing coverage reporting in the presence of anchor misplacement?

Our approach to answering the aforementioned research questions is as follows:

- Considering IoT characteristics (i.e., heterogeneity, random distribution, multiple owners), we identify each coverage hole and provide upper and lower bounds for its size. We utilize Delaunay Triangulation (DT) to partition the target sensing region into triangles. The vertices of these triangles are IoT objects. Since intra-triangle coverage holes are not uniform, our goal is to locally detect each hole and provide its bounds. The intra-triangle coverage (ITC) procedure is distributed and requires only the vertices of each triangle to involve in the calculation which makes ITC procedure scalable and efficient in terms of power consumption. We provide a theoretical analysis of IoT sensing coverage holes, and develop an efficient algorithm to detect coverage holes. We then provide the bounds based on the size of the identified coverage holes, and test the validity of the bounds empirically. The results show that the bounds become sharp as the sensing nodes increases. Furthermore, our findings are

significantly sound for many IoT coverage applications to either tolerate the coverage holes or call a healing procedure to cover the gaps.

- Calculate the effects of the anchor misplacement on localization accuracy. Then we propose a distributed scheme to detect the misplaced anchor objects. In light of the results, a novel localization framework is constructed to reflect the effect of anchor misplacement on localization. The performance evaluation of our proposed algorithm outperforms the other competitive algorithm in terms of successful detection ratio of misplaced anchor nodes, mistaken anchor ratio, and localization accuracy.
- Analyze the effects of anchor misplacement on sensing coverage. In this research, we address the sensing coverage problem and the different types of coverage holes. Our research approach focuses only in the locality of the affected sensing objects. The first type of coverage holes is actual coverage holes that have been falsely hidden, and unreported. The second is perceived coverage holes that have been falsely generated by anchor misplacement. Our results show that around 25% more perceived coverage holes will be generated on average as a result of misplacing 30% of anchor nodes which randomly deployed in the target field. This shows the significance of mitigating the impact of anchor misplacement on IoT sensing coverage.
- Calculate the true/false sensing reporting of sensing objects. This leads to measure whether or not a sensing object still reports from its original area even after anchor misplacement takes place. Our distributed scheme measures the validity of sensing coverage reporting. The scheme applies the validity criteria on each affected sensing objects and differentiates between their true positive and true negative reports. The findings of our study show that the randomness of anchor misplacement and displacement value mitigates the impact of anchor misplacement and gets higher true positive rate of the sensing report. The outcomes of our study have a wide range of applications that depend on reliability of sensing reports such as smart vehicles, leakage of pipelines, and smart buildings.

## 1.3 Document outline

This thesis is arranged as follows. Chapter 2 overviews the background and the material related to this research. It includes a detailed background of WSN and its two major services localization, and sensing coverage. Chapter 3 covers the bounds of sensing coverage holes. This includes identifying the upper and lower bounds of coverage holes in IoT deployment. The impact of anchor misplacement on localization accuracy and how to mitigate it will be detailed in Chapter 4. Chapter 5 extends our study in Chapter 4 to include studying the impact of anchor misplacement on sensing coverage. Chapter 6 presents a study to assess the validity of sensing coverage of sensor nodes under the presence of anchor misplacement. This includes providing a distributed scheme to measure whether the sensing report is “valid,” or “invalid”. Finally, we conclude with Chapter 7 in which the summary and future research problems are provided.

# Chapter 2

# Background and Literature Review

This chapter presents the background and existing research related to the work in this thesis. The chapter begins with an introduction about WSNs: characteristics and constraints. Also, it provides a detailed overview about localization and sensing coverage.

## 2.1 Wireless sensor networks

A WSN is composed of sensor nodes (SNs) which have sensing functionalities to monitor physical properties such as pressure, humidity, and temperature, as well as moving objects. Each sensor has a small processing unit, a battery as a power unit, memory, and a short-range wireless transceiver unit [8] [9]. The sensed information is normally propagated towards the Base Station (BS) possibly through intermediate nodes [10] [11]. Figure 2.1 shows the flow of sensed data starting from the SNs until reaching the end user. WSNs have distinguishing features that are different from the traditional multi-hop networks. These features are [12]:

- Sensors are densely deployed and cooperate to monitor/detect events.
- Sensors are prone to failure.
- Unlike traditional wireless networks which use peer-to-peer communication, WSNs usually use broadcast communication approach.
- Sensor nodes (SNs) are limited in resources such as power, processing capabilities, and memory.
- The topology of WSNs changes dramatically due to many reasons such as signal attenuation and sensor failure.
- WSNs are oriented to detect and/or estimate some events (not just provide communication). In this regard, data aggregation can be improved by using data fusion from multiple sensors.

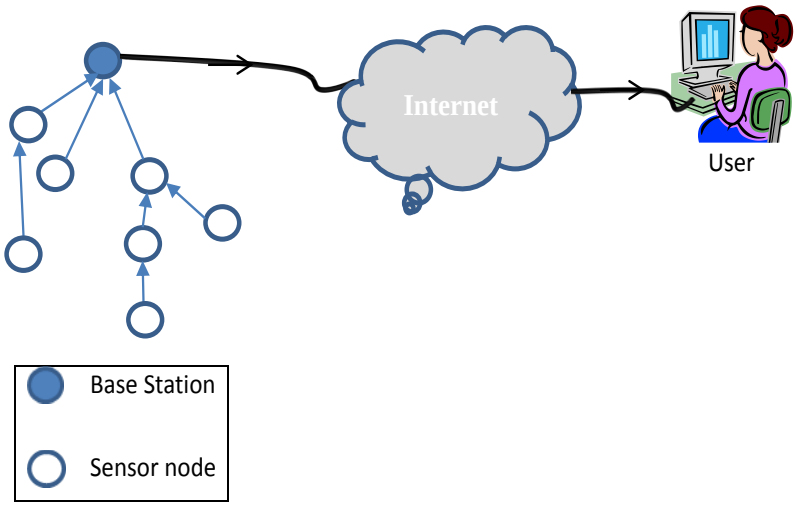



**Figure 2.1: Example of how a Typical WSN works.**

### 2.1.1 Communication in WSNs

There are two kinds of communication in WSNs single-hop or multi-hop [13] [14]. In the former, the network has a star shape as shown in Figure 2.2 (a), where the BS can communicate directly with any SN in the network. However, it is not always true that each SN has direct communication with BS (i.e., single-hop communication) especially in a non-deterministic deployment of thousands of sensors in a vast geographical region. Even in the deterministic scenario, having single-hop communication requires denser deployments for BSs due to the short communication range of SNs causing the cost to be very high. Multi-hop communication has a form of a mesh network, as shown in Figure 2.2 (b), and the communication between sensors and BSs located far away occurs via multiple intermediate hops. The SN is not only transmitting its own data, but it acts as a relay for other nodes, collaborating to propagate the data towards the BS. The existence of many paths to deliver the same data to one BS poses a routing problem to find the best possible path to propagate the data and eliminate the redundancy of transmitted data. It should be mentioned here that even multi-hop communication has limitations related to energy consumption.

The more a sensor has relaying data transmitted through it the more energy that sensor will consume.

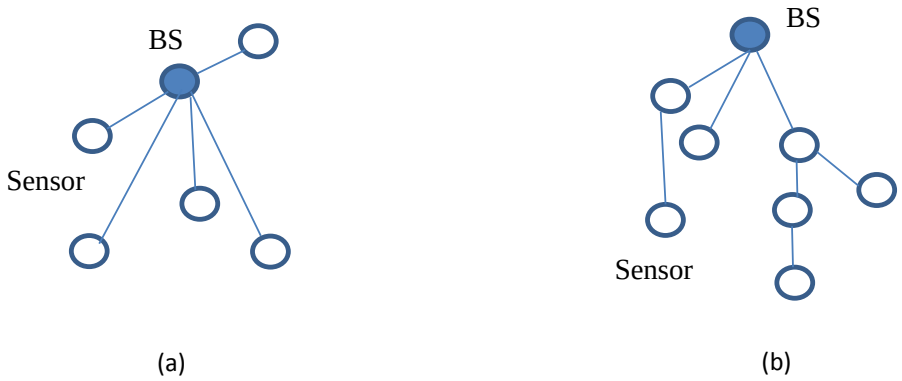



**Figure 2.2: (a) Single-hop (b) Multi-hop communication in WSNs.**

### 2.1.2 Constraints and Challenges

Technological advancements have resulted in the development of inexpensive, low-power, wireless micro-sensor networks. Figure 2.3 shows the components of the sensor node. Each sensor consists of four main components, the power unit which is usually a small battery, a sensing unit which made up of the sensor and the analogue to digital convertor (ADC), the processing unit which has two subunits: the processor and the memory, and a communication unit which is the antenna in a wireless sensor that keeps the sensor connected to the network. These units have severe resource limitations especially in their power supply, processing power, memory, and bandwidth [8]. Additional components can be added to the sensor's structure according to the application needs. For example, the localization system component (shown in the dashed box) can be added to meet the localization requirements of some applications.

WSNs usually use multi-hop communication to deliver data from sensors to BSs. This will impose a routing problem [15]. An efficient routing protocol for WSNs should consider the limited budget of resources in such networks. Energy can be saved if WSNs rely on distributed communication to arrange the processing power among all nodes not only on a specific node(s) as

in the cluster head, and coordinators in the traditional wireless networks [16]. The protocols used in WSNs should be light in power consumption and have low computational complexity; otherwise, the battery will be depleted quickly, and the network will start disconnecting [17] [18]. Security is also a major issue here in the sense that the network should be robust against security attacks and that data integrity should be preserved.

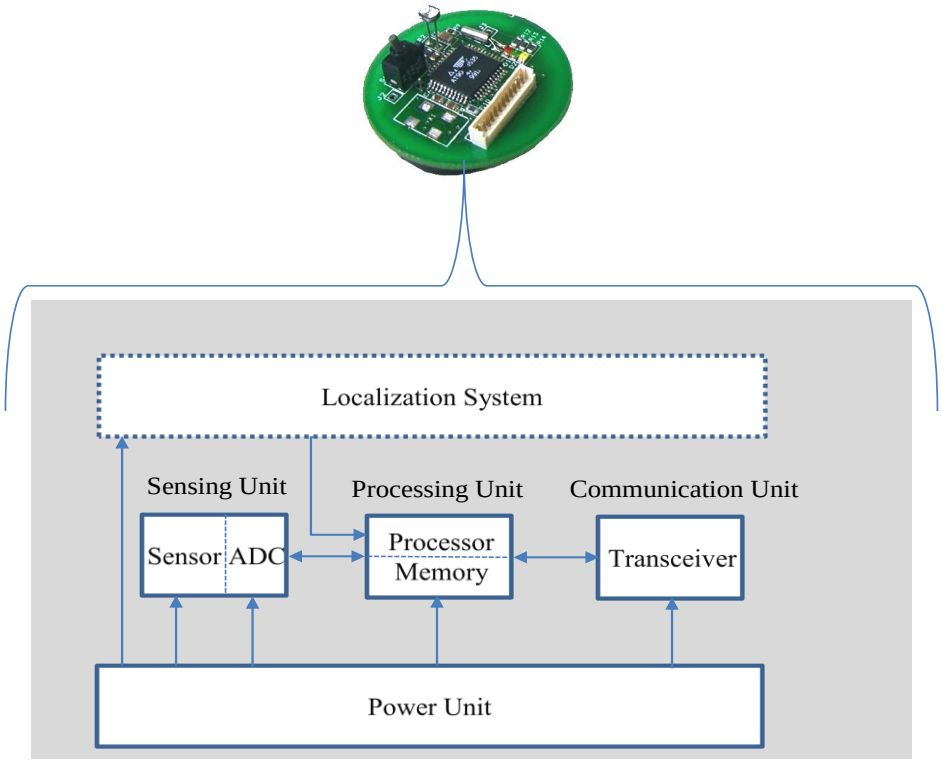



**Figure 2.3: Sensor node and its components.**

## 2.2 Localization in WSNs

This section covers a detailed overview of localization in WSNs, possible approaches to deal with localization in WSNs such as range-based and range-free. Furthermore, we also address the phases of range-based localization in single-hop WSNs. We deal specifically with the measuring phase and positioning phase. Measuring phase uses either distance-based or angle-based

techniques; while positioning phase derives the SN's location by using the measuring estimates generated from the first phase, and then applies methods such as lateration, multilateration, and angulation. Range-free localization which uses methods such as connectivity and fingerprint to estimate the locations of SNs in WSNs will also be covered.

### 2.2.1 Overview

Many of the aforementioned applications in WSNs require knowledge of the exact positions of sensing objects, and a node in a WSN has to be aware of its location in the physical world. Localization of sensors can be achieved by one of the following three ways [19] [10] :

1) Manually configuring a location into each node, which may not be practical for many uses such as a harsh environment where monitoring inherently depends on undeterministic deployment. Furthermore, it is impractical in the case of mobile sensing objects.
2) Equipping every node with a GPS receiver. This, however, increases the cost of the sensor. In fact, the current capabilities of processing and power of most sensors cannot fit a GPS receiver. Another deployment limitation is that the GPS does not work indoors properly [19] [20].
3) Designing algorithms to locate the sensing nodes [21].

### 2.2.2 Existing Localization Approaches

Localization techniques in the literature are classified in many ways depending on a set of features related to the deployment environment (indoor, or outdoor), how the scheme is executed (centralized, or distributed), mobility of anchors used (static, or mobile), the way of communication between nodes of the network (single-hop or multi-hop) [13].

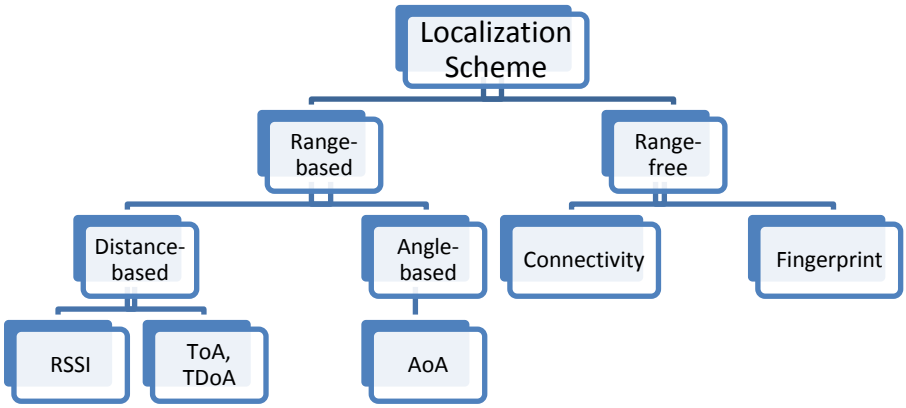



**Figure 2.4: Special classification of localization schemes**

Most localization schemes and systems depend on the communication between sensing nodes and anchor nodes. Communication in WSNs is either single-hop or multi-hop as illustrated in Section 2.1.1. Single-hop localization uses single-hop communication between the SNs and anchors; multi-hop localization uses multi-hop communication. Multi-hop localization suffers from error propagation where the error accumulates as the hopping is continuous [13] [22]. That is why range-based schemes and systems, that seek good accuracy, use single-hop localization. The range-free localization schemes can be either single-hop or multi-hop [23]. Connectivity-based systems usually use multi-hop localization such as in Distance Vector based on hops scheme (DV-hop) [24] [25], while other range-free fingerprint systems are inherently single-hop systems such as RADAR [26] or LANDMARC [27] as explained in Section 2.2.4. In this section, we provid a classification that depends on range-based versus range-free approaches as shown in Figure 2.4. Further explanation for these two approaches will be presented in the sections following.

### 2.2.3 Range-based Localization in WSNs

In range-based techniques, two main phases are usually involved to localize SNs in WSNs [13] [28] the measurement estimation phase, and the positioning derivation phase. We address the measuring phase and its related issues first.

#### 2.2.3.1 Measuring Phase

This phase is concerned with utilizing the exchanged data between the SNs and anchor nodes to estimate the distances or angles according to the technology used. For example, Time of Arrival (ToA) [29], Time Difference of Arrival (TDoA) [30] [30], and Received Signal Strength Indicator (RSSI) are used for distance estimates [31], where Angle of Arrival (AoA) is used to estimate the angle between sensing and anchor objects [32]. The next sub-section deals with the techniques that are usually used to estimate the distance measurements.

#### 2.2.3.2 Distance-based techniques

In the distance estimation phase, a node estimates its distance to other nodes in its vicinity. Distance estimation between two SNs (sender and receiver) is estimated by using measurements taken from some characteristics of the signals exchanged between these sensing nodes, including [33] [21] [34] signal speed, the elapsed time between sending and receiving the signal (time of flight), signal orientation, or signal strength. The distance estimation phase typically utilizes one or more of the following techniques:

1) Time of Arrival (ToA) [35] capitalizes on the relationship between signal speed, time of flight, and distance. This technique is widely used due to its simplicity since there is no need for additional hardware. However, it faces a difficulty in accurate calculation of the propagation time due to the high signal speed comparing to the distance[2]. Also, it requires highly synchronized clocks between the sender and the receiver.
2) Time Difference of Arrival (TDoA) [36]follows the same concept of ToA, however, it uses two different types of signals such as radio and acoustic. There is no need to synchronize the clocks of the two sensing nodes. TDoA requires additional hardware viz. microphones and speakers.

[2] The speed of radio signals, in a vacuum, is $3x10^8$ metres per second. e.g., 30 ns are required to travel a distance of 10m.

3) Received Signal Strength Indicator (RSSI) [37]: depending on the power of the transmission signal and the strength of the received signal, these values are compared to a specific model, such as the path loss model and then derives the estimated distance. This technique does not require additional overhead since it takes place anyway between the sender and the receiver. However, it suffers from multipath fading, and shadowing.

The following sub-Section addresses the technique to estimate the angle between the sender and the receiver nodes in WSNs.

#### 2.2.3.3 Angle-based technique

Angle of Arrival (AoA) [38]: it estimates the two angles between two anchors and the unknown sensing nodes, and to estimate the distance between the anchor nodes. This technique is impractical for LS-WSNs for the following reasons:

- It needs additional equipment such as an array of antennas, directional antennas or microphones which adds significantly to the size and the cost of the sensors.
- Accuracy is constrained by shadowing, multipath reflections. Therefore, each element of the antenna array should be calibrated, and stable to get reasonable accuracy since any small deviation in angle estimation results in a very large error in position estimation.
- This hardware consumes high power, making it energy inefficient.

#### 2.2.3.4 Summary of the Measuring Techniques

We provide the advantages and disadvantages of all techniques of distance and angle estimation in Table 2.1.

**Table 2.1: Advantages and disadvantages of the range-based localization techniques.**

| Localization Technique | Advantages | Disadvantages |
|---|---|---|
| **ToA** | • No need for additional hardware ➔ low cost | • Requires highly accurate synchronization of the sender and receiver clocks. ➔ Adding to the cost and complexity of a sensor network.<br>• Difficulty in accurately measuring the time of the flight. |
| **TDoA** | • No need for synchronization of the clocks of the sender and receiver.<br>• Can obtain very accurate measurements and, hence, accurate localization. | • Requires additional hardware like a microphone and speaker for the given example. |
| **RSSI** | • No additional hardware is necessary.<br>• Distance estimates can even be derived without additional overhead from communication that is taking place anyway. | • RSSI values are not constant but can heavily oscillate, even when sender and receiver do not move (fast fading, mobility of the environment, and presence of obstacles in combination with multipath fading). This affects the localization accuracy. |
| **AoA** | • No need for synchronization of the sender and receiver clocks. | • The accuracy of AoA measurements is limited by the directivity of the antenna, by shadowing and by multipath reflections.<br>• Additional hardware can obtain more accuracy, but add significantly to the size and cost of SNs. |

2.2.3.5 Positioning Phase

In the positioning phase, the distance or angle measuring estimates collected in phase one are respectively used by lateration or angulation methods, to compute the position of the blind sensing objects [39] [40] [41]. Next, we start by lateration method

2.2.3.5.1 Lateration Method

In general, the lateration method requires (*n*+1) distance measurements from the unknown node to the anchor node to estimate the blind node's location in (*n*) dimensions [42]. Trilateration depends on three distance measurements to be calculated, then the position (in 2D) of the unknown node is the intersection coordinates of the three circles centered in the anchors with distance measurements as radii [43] [44]. Trilateration is an essential geometric method which is involved in many localization systems such as GPS [45]. In the following example, $r_1$, $r_2$, and $r_3$ are three range measurements between the unknown node, *u*, and the three anchor nodes *A*, *B*, and *C* located at $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$, respectively. In ideal case where no errors are imposed to the localization, the estimated position $(x_u, y_u)$ for SN *u* is the intersection of the three circles as shown in Figure 2.5.

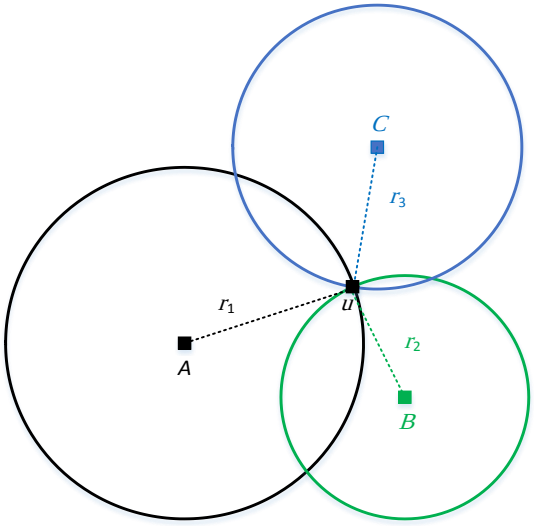



**Figure 2.5: Trilateration method in ideal case.**

The estimated position of $u\,(x_u, y_u)$ can be then calculated algebraically by solving the following non-homogeneous system.

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^{\,2} - r_3^{\,2}) - (x_1^{\,2} - x_3^{\,2}) - (y_1^{\,2} - y_3^{\,2}) \\ (r_2^{\,2} - r_3^{\,2}) - (x_2^{\,2} - x_3^{\,2}) - (y_2^{\,2} - y_3^{\,2}) \end{bmatrix}$$

This system of equations has the form $Ax = b$ where $\frac{1}{2}A$ is the leftmost matrix, $x$ is the unknown vertical vector $\begin{bmatrix} x_u \\ y_u \end{bmatrix}$, $b$ is the rightmost vector.

Next, we address multilateration method which is similar to trilateration, with one difference, that multilateration can use more than three anchor nodes to locate the unknown SN.

2.2.3.5.2 Multilateration Method

To avoid ambiguity and determine uniquely the location of a point in a plane using trilateration, the three positions of the anchor nodes should be non-collinear. Furthermore, the measurement techniques such as ToA, TDoA, RSSI, and AoA are biased estimators which means that there is a difference between the actual value of the distance (or angle) measurement and the estimated one. Thus, the measurements are erroneous which may result in the three corresponding circles (in trilateration method) not intersecting in a point; instead their intersection is an enclosed region as shown in Figure 2.6. The smaller this region is the less error affecting the localization resulting in better accuracy.

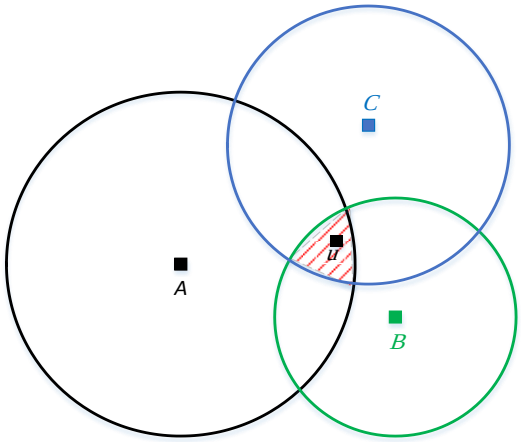



**Figure 2.6: Trilateration method in real case.**

Multilateration method [46] [17] is a generalization of trilateration method and requires more than three anchor nodes for localization. Multilateration, along with mean square error technique achieves the best estimation of the unknown vector $x$ such that $\|Ax - b\|_2$ is minimum. Note that

if the anchor is mobile, then more than three non-collinear positions for this anchor node are required for multilateration.

In next sub-Section, we deal with angulation method that utilizes the angle estimation to derive the position of the SN.

2.2.3.5.3 Angulation Method

Angulation utilizes the AoA measurements to apply the trigonometric fact that if two angles and the side between them are known then the position of the third point can be calculated as the intersection of the other remaining sides [47] [48]. For example, in Figure 2.7, A and B are two anchor nodes with known positions; while *u* is unknown sensing node. $\theta_1$ and $\theta_2$ are the measurements of AoA technique. The distance between A and B can be calculated; then the angulation method is applied to estimate the position of *u*.

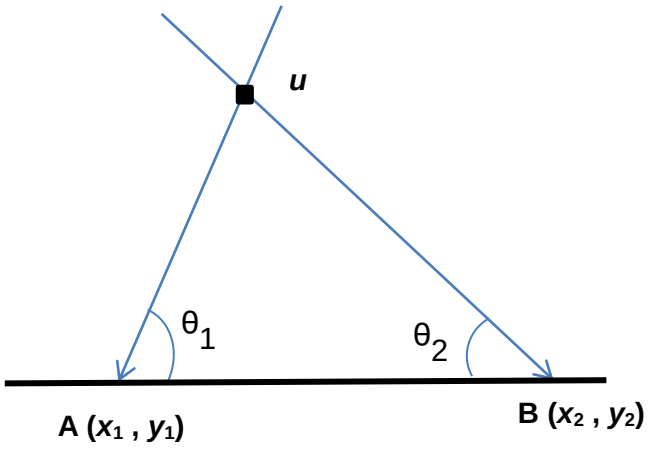



**Figure 2.7: AoA measurements.**

2.2.3.6 Summary of Localization in Range-based Systems

Figure 2.8 shows a procedure that summarizes the localization process in single-hop range-based systems.

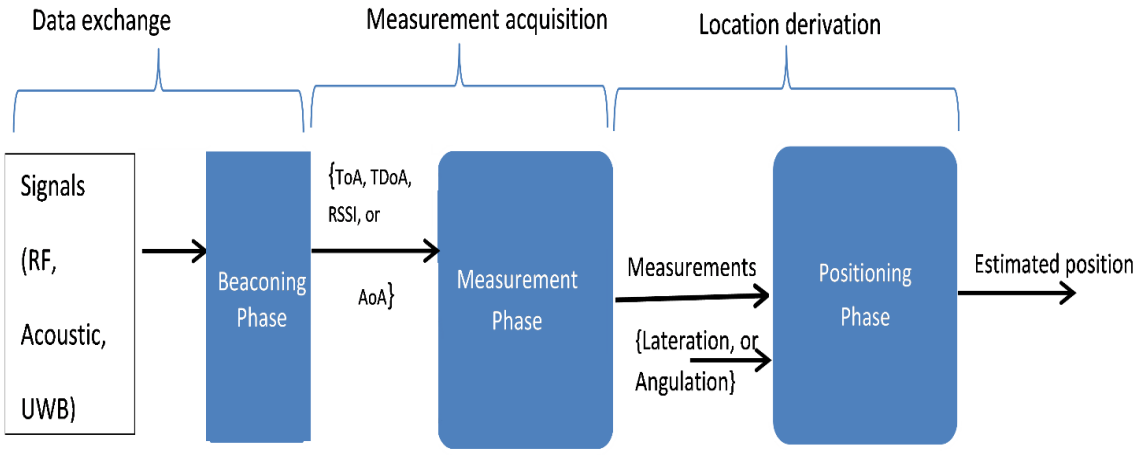



**Figure 2.8: Localization process in a single-hop range-based system.**

The flowchart above shows a three-phase process to localize sensing nodes. The first phase is the beaconing phase which is a default stage as it occurs by a spontaneous signaling and packet exchange between SNs and anchor nodes. The second phase is the measuring phase where the distance or angle measurements are estimated by using the measuring techniques such as ToA, TDoA, RSSI, or AoA. The output of the second phase (i.e., measurements) is entered as an input to the third phase (i.e., positioning phase) where the location is derived by using the positioning methods such as lateration or angulation.

### 2.2.4 Range-free Localization in WSNs

Range-free technique provides coarse-grained localization since it does not depend on calculating distances between the unknown sensing objects and the anchor objects; instead it estimates implicitly the ranges and then the location in a broad manner [49] [23] to overcome the drawbacks of range-based techniques (i.e., cost and energy consumption). Range-free schemes and systems can be classified to either connectivity-based or fingerprint-based. The former depends on the topology of the networks, where the latter depends on storing information of some locations (prints) for retrieving and utilizing at a later time. In both cases, the implicit estimation of the range and location is erroneous and does not fully reflect the actual distance and location. However,

range-free techniques provide a cost-effective alternative to the expensive range-based techniques and, hence, they are very prominent in IoT. This class of techniques is particularly oriented to the applications that do not require high accuracy in localization.

#### 2.2.4.1 Connectivity-based Approach

Connectivity-based schemes depend on graph topology of the network [50]–[52]. Some techniques such as DV-hop [24] utilize the minimum hop count (i.e., shortest path) between the sensing and anchor objects to estimate the distances first and then the location. Other connectivity-based techniques depend on polygons in which the vertices are anchor nodes. For example, the APIT scheme [23] utilizes the triangle of three anchors and decides whether the unknown SN is inside this triangle or not. Using this information, a SN's location can be estimated by intersecting all triangles containing this SN and then taking the centroid of this intersected region.

#### 2.2.4.2 Fingerprint-based Approach

Fingerprint or scene analysis depends on two phases. The first phase constructs the offline data base by recording RSSI at different locations with respect to different anchor nodes from which an RF map is constructed. The second phase (i.e., online phase) matches a set of observed RSSI values with the recorded RSSI values in the database created by the offline phase [53]–[56]. Clearly, this approach is time consuming and impractical for IoT applications. RADAR [26] and LANDMARC [27] are examples of such fingerprint systems**.**

#### 2.2.4.3 Summary of Localization in Range-free Systems

Figure 2.9 shows a procedure that provides the localization process in range-free systems.

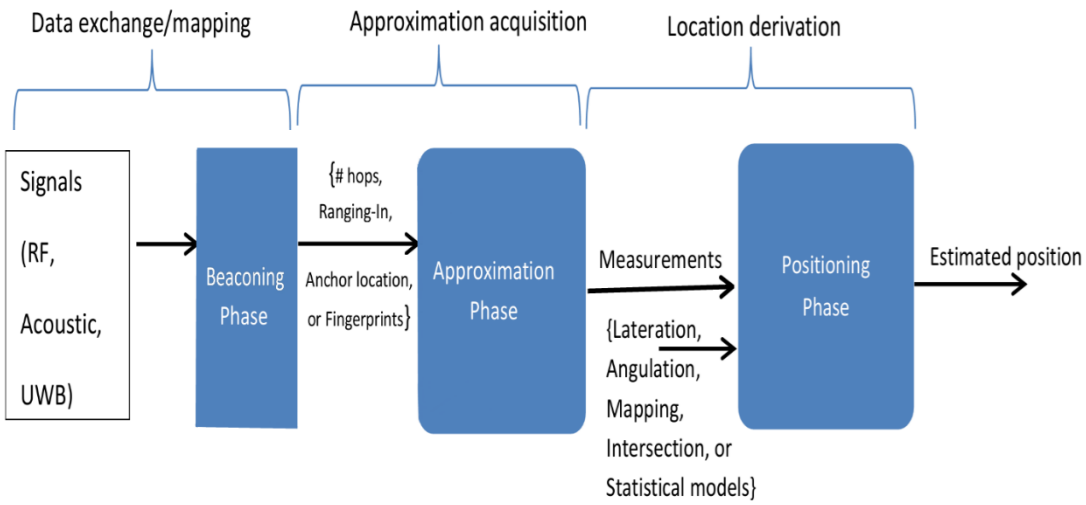



**Figure 2.9: Localization process in a range-free system.**

Like Figure 2.8, Figure 2.9 shows a three-phase process to localize sensing objects in WSNs. The first phase is the same in both figures with a slight change in Figure 2.9 where mapping can be used a priori in range-free fingerprint systems. The second phase is different since range-free system has no measurement estimates; instead it approximates the distance by other means such as number of hops, ranging-in that checks whether the SN is in range or not, anchor location, or fingerprint techniques. The third phase is the positioning phase. It takes the measurement approximation as input and applies a positioning method such as: lateration, angulation, mapping, intersection, or statistical models to derive the SN's location.

The advantages and disadvantages of the range-free schemes and systems are listed in Table 2.2.

**Table 2.2: Advantages and disadvantages of the range-free localization techniques.**

| Localization Technique | Advantages | Disadvantages |
|---|---|---|
| **Connectivity** | • No need for additional hardware ➔ low cost | • Provides coarse-grained localization ➔ not accurate. |
| **Fingerprint** | • No need for additional hardware ➔ low cost | • Provides coarse-grained localization ➔ not accurate.<br>• More effort and time are needed to build the offline database. → Not practical for large-scale WSNs.<br>• More suitable for indoor applications. |

The next section deals with sensing coverage which is another important WSN-based service in IoT.

## 2.3 Sensing Services using WSNs

Sensing services is a fundamental goal of WSNs and sensing coverage is its leverage to provide a reliable service. Sensing coverage measures to what extent the sensing reports reflect the true physical surroundings in the target sensing field. This means without a good coverage the sensing service would be unreliable or even obsolete.

### 2.3.1 Sensing Coverage Problem

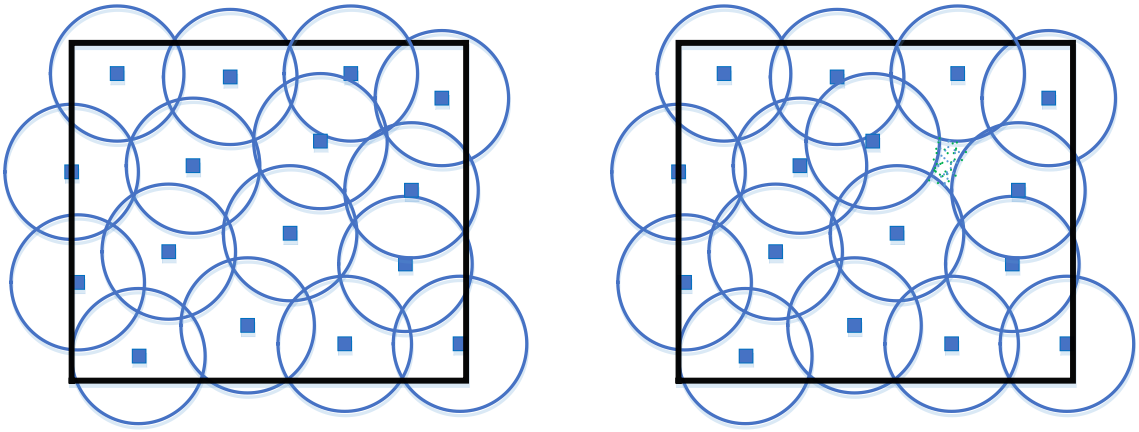
One of the main reasons that degrade the quality of sensing service is the presence of coverage holes in WSNs. Coverage hole exists if there are some points in the sensing field are not covered by any sensing object. However, there are some applications require at least k sensing objects to cover any point in S. This type of coverage is called k-coverage and it is used to allow more fault tolerance in some critical applications such as nuclear reactor's leakage [57]. The following is a definition of sensing coverage.

**Definition 2.1:** Let $S$ denotes the target sensing field. Let $N = \{s_i: s_i \text{ is a sensing object}; 1 \leq i \leq n, \text{where n is the number of sensing objects}\}$, be a set of sensing objects with unknown location $(x_i, y_i)$ in a plane. Each sensor $s_i$ has estimated its location $(x_i', y_i')$ and a sensing range $R_{si}$. Let $p$ be a point in $S$, then $p$ is covered if there is at least one $s_i$ such that $p$ is within distance of $R_{si}$ from $s_i$. In other words,$\{\exists\, s_i | d(p, s_i) \leq R_{si}, 1 \leq i \leq n\}$, where $d(a, b)$is the Euclidean distance between $a$ and $b$.

Applications vary in their sensing coverage requirements. Some of them require single-sensing coverage, which means any point in the target region should be monitored by at least one sensing node; while other applications require high coverage and, hence, require more than one to monitor each point in the target region.

### 2.3.2 Sensing Coverage Holes

Coverage holes exist when there are some points in the sensing field that are not covered by any sensing objects as shown in Figure 2.10.



(a) Full coverage (b) Coverage hole

**Figure 2.10: Sensing coverage.**

### 2.3.3 Sensing Models

Many sensing models can be constructed according to the application and surrounding environment. However, most of these models agree on that sensing fades out as distance increases. The following formula reflects this observation [58]:

$$S(s_i, p) = \frac{\lambda}{(d(s_i, p))^K} \quad \textbf{(2.1)}$$

Where $S$ denotes the sensibility between sensing nodes $s_i$ and point $p$, $d(s_i, p)$ is the Eucledean distance, and both $\lambda$ and $K$ are positive constants related to the sensor's technology.

There are different types of sensing models such as binary disc model and probabilistic sensing model.

#### 2.3.3.1 Binary Disc Model

In the binary disc model, a sensing node is assumed to do 360° monitoring. Therefore, a point in sensing field is covered if it is within the circular sensing range of at least one sensing node. Otherwise, it is not covered, as given in the following equation:

$$\mathrm{C}(p) = \begin{cases} 1 & if\ d(s_i, p) \leq R_{s_i} \\ 0 & otherwise \end{cases} \quad \textbf{(2.2)}$$

Where $R_{s_i}$ is the sensing range of $s_i$. Thus the binary disc model abstracts the sensing coverage of $s_i$ by a disc of radius $R_{s_i}$.

#### 2.3.3.2 Probabilistic Sensing Model

This model depends on uncertainty in sensor detection. Therefore, it utilizes the detection probability when the point of sensing field is at distance greater than the value of uncertainty, but within the sensing range [59].

$$\mathrm{C}(p) = \begin{cases} 1 & if\ d(s_i, p) \leq R_{si} - \epsilon \\ 0 & if\ d(s_i, p) \geq R_{si} + \epsilon \\ e^{-\alpha\beta^{\gamma}} & if R_{si} - \epsilon < d(s_i, p) < R_{si} + \epsilon \end{cases} \quad \textbf{(2.3)}$$

Where $\epsilon$ is the uncertainty value in sensor's detection, $\beta = d(s_i, p) - (R_{si} - \epsilon)$, and both $\alpha$ and $\gamma$ are parameters that measure probability of detection when $p$ is at distance greater than $\epsilon$ but still

within a range of $R_{si}$. Different values of these parameters reflect the characteristics of various types of sensor and, consequently, different detection probability.

### 2.3.4 Sensing Coverage Deployment Methods

The deployment strategies vary according to the application, the size of target area, the available information about the density and locations of sensors, and whether or not the target region is accessible.

#### 2.3.4.1 Deterministic Deployment

This type of deployment depends on predefined parameters such as the shape of the network, sensing node's location, distance between sensing nodes, and density. Deterministic deployment of sensors allows more control on constrained resources of sensors such as energy consumption. Most of the schemes dealing with deterministic deployment choose energy consumption as the most important metric to optimize [60]. Example of this type of deployment is grid-based deployment: hexagon, square, and equilateral triangle. Equilateral triangle grid-based deployment guarantees complete coverage and requires a minimum number of sensing nodes [61]. Art Gallery problem [62] is also a traditional problem of this type of deployment. In this problem, one seeks to place a minimum number of sensor cameras such that every point in the gallery is monitored by at least one sensor camera.

#### 2.3.4.2 Random Deployment

Unlike deterministic deployment, random deployment has no available information about the shape of the network and the location of the sensors. This type of deployment is ideal for a large scale network such as the IoT, harsh inaccessible areas such as forests, mountains, dangerous areas, war zones, and hazardous areas, such as chemical plant explosions, and nuclear plant accidents. However, random deployment results in accumulating sensing nodes in some parts of the target fields. Thus the coverage is not full. Keeping all sensing nodes active simultaneously results in

quicker energy depletion and, consequently, the network disconnects. This means that no more data gathering will be reported from certain areas. That is why many sensing coverage schemes in large-scale networks includes a sleep schedule that controls the active sensing nodes [63] to prolong the lifespan of the network.

# Chapter 3

# Identifying Bounds on Sensing Coverage Holes

WSN is a fundamental IoT enabling technology [64]. A successful integration of WSN in IoT requires merging WSN's resources in a larger IoT pool of resources [65]–[67]. For instance, sensing coverage is a main service of WSN. For more reliable sensing coverage, several owners of heterogeneous sensing networks can collaborate to provide better service for end users as Figure 1.1 illustrates. Collaborative wireless sensor network becomes a significant method to overcome the limited resources of each sensing node [68].

This chapter is organized as follows. Motivations and contributions are presented in Section 3.1. Section 3.2 presents the problem formulation, and assumptions of our research. Analysis using Voronoi Diagram (VD) [69] and Delaunay Triangulation (DT) [70] toward efficient coverage are given in Section 3.3. Section 3.4 is devoted to study in detail the Intra-triangle coverage, and the algorithm to detect and bound coverage holes. Section 3.5 presents experimental results to validate our proposed algorithm. Section 3.6 concludes the chapter.

## 3.1 Motivations and Contributions

Existing work on sensing coverage in WSN assume sensing nodes are homogeneous and belong to only one sensing service provider. Most of the research addresses deterministic sensor placement and deployment planning to achieve greater coverage and/or to extend the network lifetime [71] [72]. Our research, on the other hand, investigates IoT sensing coverage where sensing nodes are a) heterogeneous as they have different functionalities and capabilities, b) randomly deployed which is normal in IoT, and c) belong to different sensing service providers.

We identify the coverage holes and provide upper and lower bounds for these coverage holes. We utilize DT to partition the target sensing region into triangles. The vertices of these triangles are

sensing nodes. Since intra-triangle coverage holes are not uniform, our goal is to locally detect each hole and provide its bounds. intra-triangle coverage (ITC) procedure is distributed and requires only the vertices of each triangle to involve in the calculation which makes ITC procedure scalable and efficient in terms of power consumption. We provide theoretical analysis of IoT sensing coverage holes, and develop an efficient algorithm to detect coverage holes. We then provide upper and lower bounds of the identified coverage holes, and test the validity of these bounds empirically.

This research contributes towards the realization of a sensing cooperative IoT, in which the available sensing resources (from overlapped and overlaid sensors) are used to achieve a given coverage. As such, applications can determine whether to tolerate some coverage holes or whether to initiate a healing procedure to mitigate some coverage holes in the network. To the best of our knowledge, this is the only research that investigates IoT sensing coverage: identifying the coverage holes locally, and providing upper and lower bounds on each sensing coverage holes.

## 3.2 Problem Definition and System Model

Given a dynamic and random deployment of sensing nodes, we are interested in detecting the coverage holes and providing upper and lower bounds on coverage holes in a distributed manner. The analysis exploits powerful structures in computational geometry such as the VD and DT. Our approach to detect and bound coverage holes depends only on the locality of each convex polygon of the computational structure that represents the sensing field. We make the following assumptions:

1) Sensing nodes can send/receive packets to/from their neighbors. This assumption is important to exchange the sensing nodes' information locally through, most likely, multi-hop communication in order to build our computational structure in a distributed way.
2) Sensing nodes know their location.
3) No three Sensing nodes are collinear. This assumption enables constructing the Delaunay Triangulation.

4) The sensing target field is bounded. This is the case for most IoT applications.

Random deployment refers to the case where the placement of groups of sensing nodes (belonging to different providers) in the target field is independent from other groups. Let $N = \{s_i: s_i \text{ is an sensing node}; 1 \leq i \leq n\}$, be the collective set of IoT sensing nodes with location of sensor $s_i$ being $(x_i, y_i)$. We define $NH(s_i)$ to denote the neighborhood of $s_i$, that is, $NH(s_i) = \{s_j | d(s_i, s_j) \leq R_{t,s_i} \text{ and } d(s_i, s_j) \leq R_{t,s_j}, s_i \neq s_j, s_j \in N\}$, where $d(s_i, s_j)$is the Euclidean distance between $s_i$ and $s_j$, and $R_{t,s_k}$ is the transmission range of IoT $s_k$. Each sensing node that receives this information is able to estimate its distance from the emitting sensor.

In this research, we adopt the binary disc model. Figure 3.1 shows the disk model representation of the overlapped region in Figure 1.1.

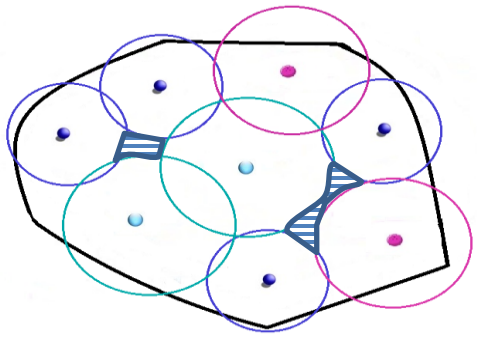


**Figure 3.1: Disc model of overlapped IoT sensing providers with coverage holes.**

## 3.3 Towards Efficient Sensing Coverage Detection

We now investigate the full coverage of target sensing field *S*. Initially, we address the randomness of sensing nodes, but with equal values of sensing range, denoted as $R_s$. We use the definition of the coverage problem presented in Section 2.3.1.

Let *p* be a point in *S*. We call $s_i$ a dominant sensor of point *p* if $s_i$ has the shortest distance to *p* among all other sensing nodes in *S*. That is $dom(p) = \{s_i | d(p, s_i) = \text{Min}(d(p, s_j), 1 \leq j \leq n)\}$, where $s_i, s_j \in S$. Let $\text{MaxMin}(S) =$

$\underset{p_j \in A}{\text{Max}} \left( d\left(p_j, dom(p_j)\right), \text{ j } \in [1, \infty) \right), p_j \in S$. Note that if MaxMin(S) $\leq R_s$ then $S$ is fully covered. However, it is not feasible to obtain MaxMin($S$) among infinite number of points $p_j$ in *S*. To overcome this problem, we utilize VD to cluster the sensing field *S* into adjacent convex polygons, called cells and denoted by Vor($s_1$), Vor($s_2$), ….., Vor($s_n$). Each cell Vor($s_i$) is associated with only one sensor $s_i$, $1 \leq i \leq n$ as shown in Figure 3.2. For two sensors $s_i$ and $s_j$, in the plane field, the perpendicular bisector of the line segment of $s_i$ and $s_j$ splits the plane into two half-planes. Let $h(s_i, s_j)$ denote the half plane that contains $s_i$, while $h(s_j, s_i)$ denote the half plane that contains $s_j$. Note that a point $p \in h(s_i, s_j)$ if and only if $d(p, s_i) < d(p, s_j)$. Thus Vor($s_i$) is the intersection of all half-planes generated by the perpendicular bisectors of the line segments of $s_i$ and each sensor in $NH(s_i)$. . Each bisector line segment is called an edge and the endpoints of this edge are called vertices. For any point $p$ in $Vor(s_i), 1 \leq i \leq n$, then $s_i$ the closest sensor to $p$. Note that if $p$ is on a common edge of two neighbouring polygons, then it is equidistant from the two sensors associated with these polygons [69]. The following lemma provides the necessary and sufficient conditions to have full coverage in VD.

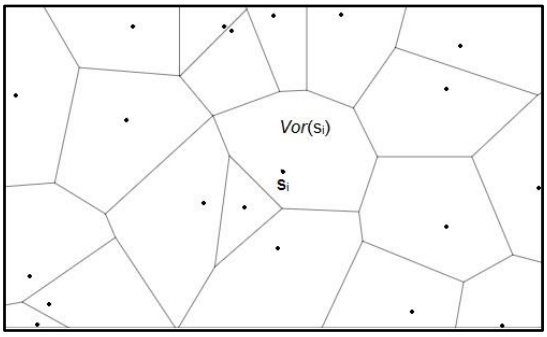



**Figure 3.2: VD partitions sensing field into convex cells.**

**Lemma 3.1:**

Sensing field *S* is fully covered if and only if all vertices in its corresponding Voronoi Diagram have a distance less or equal to $R_s$ to their associated sensors.

**Proof.**

Proving the "only if" part is straight forward. We focus on the "if" part. Voronoi Diagram partitions a sensing field into convex cells. The farthest point in any convex polygon from its associated sensor is one of its vertices. Let $u$ be a vertex in Vor($s$) with

$$d(u,s) = \max_{\substack{j \in [1,n] \\ k \in [1,|V(\mathrm{Vor}(s_j))|]}} d(v_k, s_j), v_k \in V(\mathrm{Vor}(s_j)) \quad \textbf{(3.1)}$$

and $V(\mathrm{Vor}(s_j))$[3] is the set of all vertices in $\mathrm{Vor}(s_j)$. This means $\mathrm{MaxMin(S)} = d(u,s)$. Therefore, for any point $x \in Vor(s_i)$, $d(x,s_i) \leq \mathrm{MaxMin(S)} \leq R_s$. □

The coverage problem of sensing field *S* is now converted, by Lemma 3.1, from checking an infinite set of points in *S* into testing a finite set[4] of points that represent the cell's vertices of VD. This lowers the computational cost, adding to the feasibility of the solution. If we could maintain $\mathrm{MaxMin(S)} \leq R_s$, we guarantee the full coverage of *S*. Therefore, VD is a powerful tool to show the existence of coverage holes rather than to quantify the coverage of WSN [73]. This is because Voronoi polygons have different convex shapes with various numbers of edges and have a non-unit-circular model. Therefore, VD does not provide much information about the location and the size of each coverage hole in the field. Thus, we need to have a more efficient structure to control and track the boundary of each coverage hole. So we triangulate the set of sensor points. The vertices of the generated triangles are the sensors. Two sensors $s_i$ and $s_j$ form a triangle edge if Vor($s_i$) and Vor($s_j$) have a common Voronoi edge $e$. This implies that the triangle edge $s_i s_j$ is a segment of the perpendicular bisector line of $e$. This triangulation is called DT which provides angle-optimal planar triangles such that the circle that circumscribes any triangle, with non-collinear sensors, is devoid of any other sensors. Note that the strong property of convexity in VD is still held in DT as any triangle is the basic convex polygon. Next, we provide a corollary that links the coverage problem to the edges of DT.

[3] The average number of vertices in any Voronoi cell is less than 6 [69].
[4] The size of this set is at most 2n-5 [69].

**Corollary 3.1:**

Let ℵ be a set of sensor nodes deployed in a bounded field *S*. Furthermore, let $R_t \geq 2R_s$. If the sensing field *S* is fully covered, then the length of every Delaunay triangulation edge is at most $2R_s$.

**Proof.**

Assume the sensing field *S* is fully covered, then every point in the field has a maximum distance of $R_s$ from at least one sensor. Let $x$ be a point on the bisector line of segment line joining two sensors $s_1$ and $s_2$. Point *x* is part of the common Voronoi edge between the Voronoi cells associated with $s_1$ and $s_2$. This means $\mathrm{d}(x, s_1) \geq \mathrm{d}(s_1, s_2)/2$ and $\mathrm{d}(x, s_2) \geq \mathrm{d}(s_1, s_2)/2$. But $\mathrm{d}(x, s_1) = \mathrm{d}(x, s_2) \leq R_s$ which implies that $\mathrm{d}(x, s_1) + \mathrm{d}(x, s_2) \leq 2R_s$. Thus, $\mathrm{d}(s_1, s_2) \leq 2R_s$. □

The heterogeneity in IoT means that sensors have different sensing ranges. Let $R_{s_i}$ denotes the sensing range of sensing node $s_i$ . Let $s_i$ and $s_j$ are two triangle vertices that have sensing ranges $R_{s_i}$ and $R_{s_j}$, respectively, then the following lemma holds.

**Lemma 3.2:**

If the triangle Δ is fully covered, then every pair $s_i$ and $s_j$ of its set of vertices satisfies the following condition:

$2r \leq R_{s_i} + R_{s_j}$, where $r$ is the radius of the circumcircle of Δ.

**Proof.** Follows from the proofs of Lemma 3.1 and Corollary 3.1. □

Note that if *L* is the longest side of Δ. Then its opposite angle $\theta$ is the largest among the three. It follows that $\frac{\pi}{3} \leq \theta \leq \frac{\pi}{2}$. By a well-known sin formula we now have

$$2rsin\frac{\pi}{3} \leq L = 2rsin\theta \leq 2rsin\frac{\pi}{2}$$

$$\sqrt{3}r \leq L \leq 2r$$

We next investigate how to detect and define the bounds of each uncovered area in DT.

### 3.4 Intra-triangle Coverage

The coverage problem is reduced by DT to studying the coverage of each individual triangle in DT. Lemma 3.2 shows that if we could find two vertices $s_i$ and $s_j$ in Δ with $R_{s_i} + R_{s_j}$ less than twice of the radius of the circumcircle (i.e., the *circumradius*), then there is an uncovered area in Δ. Alternatively, if $R_{s_i} + R_{s_j}$ is less than the length between $s_i$ and $s_j$, then there is a coverage hole in Δ. According to the largest angle $\theta$ in Δ, we differentiate three scenarios of the circumcenter: the circumcenter is inside Δ if $\theta < \frac{\pi}{2}$, outside Δ if $\theta > \frac{\pi}{2}$, or on the longest side opposite to $\theta = \frac{\pi}{2}$. Although DT provides the best possible optimal-angle planar triangles (angles around $\frac{\pi}{3}$), in random deployment it is possible to find largest angles greater than $\frac{\pi}{3}$. The following question then arises: What is the minimum density of sensors such that DT is well behaved?

Let $R_s$ be the minimum sensing range among all sensing nodes that participate in the coverage of the target field. In optimal cases, all angles of Δ are equal to $\frac{\pi}{3}$ (i.e., equilateral triangle) and the length of triangle's side is $d = \sqrt{3}R_s$ [74]. Thus the area of a triangle is $\frac{3\sqrt{3}}{4}{R_s}^2$. Furthermore, the number of triangles in any triangulation is *2N-2-k*, where *N* is the number of sensors and *k* is the number of which are on the convex hull of *N* [70]. Assume the sensing field *S* has a size *L x L*; the area size that should be covered by each triangle is $\frac{L^2}{2N-2-k}$. Therefore,

$$\frac{L^2}{2N-2-k} < \frac{3\sqrt{3}}{4}{R_s}^2 \quad \textbf{(3.2)}$$

This gives

$$\frac{2L^2}{3\sqrt{3}{R_s}^2} + \frac{2+k}{2} < N \quad \textbf{(3.3)}$$

We assume that the minimum density is achieved. Let $s_i$be an sensing node vertex in a triangle Δ. Then $s_i$contributes to the intra-triangle sensing coverage of Δ. The coverage contribution of $s_i$

is the size of the angular sector centered at $s_i$ with radius $R_{s_i}$. Calculating the contribution of $s_i$ in $\Delta$ requires the angle at $s_i$. Since the lengths of all edges of $\Delta$ are known, we use the cosine formula to extract the angle at each sensor. That is

$$\alpha = cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad \textbf{(3.4)}$$

Where $a$, $b$ and $c$ are the lengths of $\Delta$'s sides, and $\alpha$ is the angle opposite to side of length c. Therefore, the coverage contribution of $s_i$ is

$$CNT(s_i, \Delta) = \frac{\alpha}{2} {R_{s_i}}^2 \quad \textbf{(3.5)}$$

where $\alpha$ is the angle at $s_i$ in triangle $\Delta$. The following formula gives the $ITC$ of $\Delta$, denoted by $ITC(\Delta)$. That is

$$ITC(\Delta) = \sum_{s_i \in V(\Delta)} CNT(s_i, \Delta) - A_{mut}, \quad \textbf{(3.6)}$$

where $A_{mut} = \left(A_{1,2} + A_{1,3} + A_{2,3}\right) - A_{1,2,3}$, $V(\Delta)$ is the set of the three vertices of $\Delta$, and $A_{ij}$ is the common area size contributed by both angular sectors centered at vertices $s_i$ and $s_j$ and $A_{1,2,3}$ is the area covered by all three vertices.

### 3.4.1 Discovering Coverage Holes

Assume a random deployment of sensing nodes over a terrain *S*. Without loss of generality, we assume that sensing node network is stationary at the time instance of discovering the coverage holes. A low computational cost and distributed algorithm can be used to construct a DT of *S* such as the localized algorithm in [75]. Clearly, the uncovered area will be inside $\Delta$ if the largest angle $\theta < \frac{\pi}{2}$. Otherwise the uncovered area is extended to outside $\Delta$. The latter case will be considered by the intra-coverage analysis of the neighbouring triangle that contains the circumcenter of $\Delta$. However, it is still possible to find part of the uncovered area inside $\Delta$ in the case where $\mathrm{d}(\mathrm{s_i}, \mathrm{s_j}) >$

$R_{s_i} + R_{s_j}$ for any two vertices$s_i$ and $s_j$ in Δ. For the first case, the centroid of the coverage hole will be computed as well as the boundary of this hole which will be discussed next.

The intra-triangle uncovered areas have different shapes; however, we model the upper and lower bounds of each uncovered area in triangle Δ as circles. The lower bound circle is a circle centered in the centroid of the polygon that strictly contains the uncovered area in Δ; it is the largest circle that can be inscribed inside the uncovered area. On the other hand, the upper bound circle is the minimum circle that circumscribes the uncovered area of Δ. To compute lower and upper bounds for the uncovered area in Δ, we follow the following procedure: first, we find a set $U$ of intersection points, namely the angular sectors and the edges of Δ, and the intersection points of the angular sectors themselves. Let $s_i$ and $s_j$ are two vertices in Δ. If $R_{si} + R_{sj} > d(s_i, s_j)$ we exclude the intersection points between the circles centered in $s_i$and $s_j$ and the edge $s_i s_j$. Let $U'$ be the new set of intersection points. The points of $U'$' form a polygon *P*. Our goal is to find the minimum/ maximum circle that circumscribes/inscribed-in *P*. To do that, we need to determine the centroid $c$ of this polygon. The coordinates of the centroid is given by the following formula [76]:

$$c_x = \frac{1}{6A} \sum_{i=0}^{|U'|-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad \textbf{(3.7)}$$

$$c_y = \frac{1}{6A} \sum_{i=0}^{|U'|-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad \textbf{(3.8)}$$

where $A$ is given by $A = \frac{1}{2} \sum_{i=0}^{|U'|-1} (x_i y_{i+1} - x_{i+1} y_i)$, and $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$ are two consecutive points on *P*'s hull. Let $R_l = Min_{p_i \in U'}\ d(c, p_i))$ . The circle centred in c with radius $R_l$ represents a lower bound of the uncovered area in Δ. Likewise, let $R_u = Max_{p_i \in U'}\ d(c, p_i))$. Then $\pi {R_u}^2$ represents the size of the minimum circle that circumscribes *P* and, hence, considers as an upper bound of the uncovered area in Δ.Therefore, we have the following bounding formula:

$\pi {R_l}^2 < Uncovered\ Area < \pi {R_u}^2$ where both bounding circles are cantered at the centroid of a polygon *P* that contains the uncovered area. It should be noted here that the use of centroid *c* instead of circumcenter of Δ is more effective for the following reasons: 1) the circumcenter of Δ does not always belong to the uncovered area due to the variation of IoT sensing ranges. 2) The circumcenter could be outside Δ which makes the calculation of $ITC$ irrelevant. 3) The bounds using the centroid $c$ are tighter as it represents the uncovered area more fairly. Next, we present the algorithm that deals with $ITC$ to detect non-uniform uncovered areas and provide a uniform upper and lower bound for these areas.

### 3.4.2 Lower and Upper Bounds (LUB) Algorithm

The steps of analytical study to compute lower and upper bounds can be summarized in the following algorithm.

**Algorithm 3.1:** Lower and Upper Bounds (LUB)

**Input:** triangle Δ
**Output:** *c*, lowerBound, upperBound

```
1  if HasCoverageHole(Δ) then
2      P  = findPolygon(Δ);
3      c  = findCentroid(P);
4      Rl = findRadiusLowerBound(P, c);
5      Ru = findRadiusUpperBound(P, c);
6      return c, lowerBound, upperBound;
7  end if
```

LUB algorithm assumes that all sensors have been localized and their locations are known. While DT is being constructed, each sensor starts to know its neighbours for each triangle Δ in DT. LUB algorithm first check the existence of coverage hole by calling *HasCoverageHole(*Δ*)* function which simply checks Δ against the coverage criteria in Lemma 3.2. If a coverage hole is discovered, the function *findPolygon(*Δ*)* is invoked to find the polygon that strictly circumscribed the uncovered region as discussed in this section. *findCentroid(P)* will apply the equations (**3.7**) and (3.8) to find the centroid of *P*. The remaining is to call *findRaduisLowerBound(P, c)* to calculate the shortest

distance between *P*'s vertices and *c* which is the radius of the lower bound. Similarly, *findRaduisUpperBound(P, c)* returns the longest distance between *P*'s vertices and *c*.

## 3.5 Experimental Results

We conduct several experiments where random non-uniform sensing nodes are deployed in the target field. These experiments tend to show the validity of our algorithm and its numerical computation. We use Visual Studio C++ to implement the algorithm. In all experiments, we set the values of the parameters in the following way, unless otherwise stated: the terrain is 300X300 $m^2$, the number of sensors N=350, the variance of IoT sensing range is 5m. The results of all conducted experiments are the average of 10 runs. We use the implementation of a distributed algorithm in [77] to construct the DT that represents the target sensing field. We first study the effect of sensing range on coverage. We also investigate the impact of sensing node density on the coverage percentage. Figure 3.3 shows that increasing the average of sensing coverage will increase the coverage percentage. The results from Figure 3.3 demonstrate the consistency and the validity of our approach in a typical setting with well-understood sensing coverage parameters.

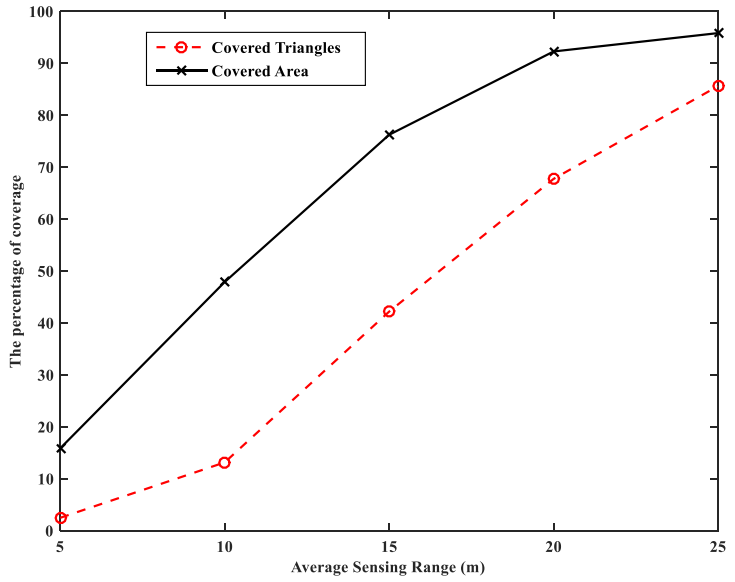



**Figure 3.3: Coverage percentage vs. average of sensing coverage.**

We next demonstrate the performance and consistency of the LUB algorithm. To assess scalability, we enlarge the target field to be 500x500 $m^2$ with sensing nodes $N$=600, and the value of average sensing range to be 20m (with variance of 5m), unless otherwise stated. Figure 3.4 shows that the upper and lower bounds are slightly tighter (closer to the actual size of the coverage hole) as the sensing node density increases. This is because adding more sensing nodes will fine tune large uncovered areas and, therefore, makes the bounds closer to the actual size of coverage holes. Increasing the density maintains the largest angel of any triangle in DT within the range of optimality (i.e., $\frac{\pi}{3}$), which makes DT well behaved. Note that the lower bound is on average twice as much closer to the actual size of coverage hole than the upper bound.

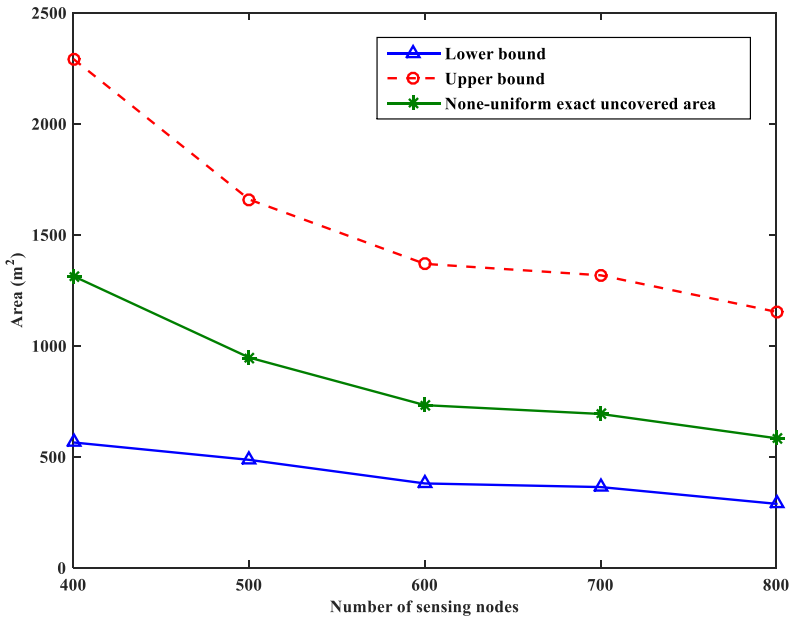



**Figure 3.4: Lower and upper bound of uncovered area with different sensing densities.**

Next, we show the impact of the sensing coverage on the derived bounds. We set the variance of sensing coverage of all sensing nodes to be 5m around the average sensing coverage. The results show that the average IoT sensing range enhances the behaviour of LUB algorithm and the values of the bounds becomes closer to the actual size of the coverage hole as shown in Figure 3.5. As the

average sensing coverage increases, both bounds converge to the actual size of coverage hole and this is expected as the uncovered areas becomes smaller and finer.

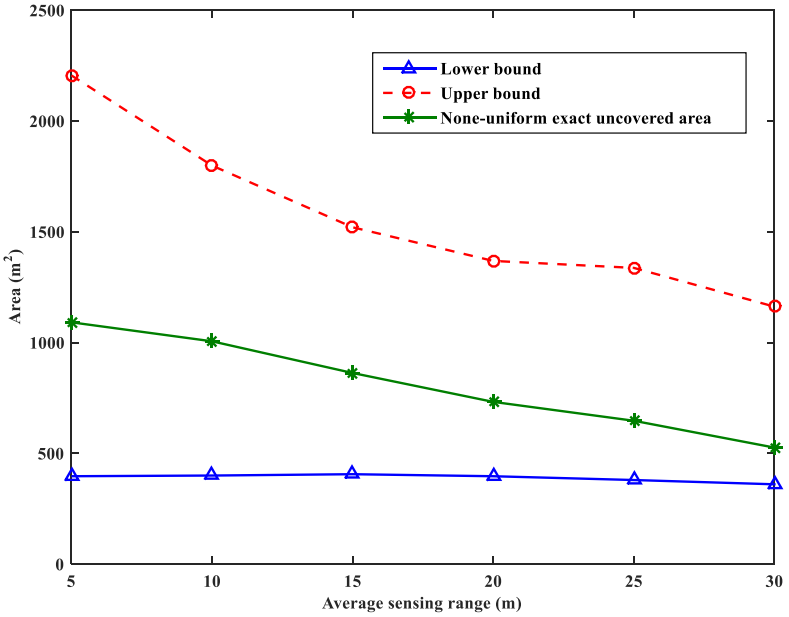



**Figure 3.5: The impact of average IoT sensing range on LUB.**

## 3.6 Summary

The continuing research in sensing coverage is essential for the realization of IoT. This chapter investigates the IoT sensing coverage problem where heterogeneous and randomly deployed sensing nodes are considered. Computational geometry provides a localized approach that enabled us to discover the problem in a distributed manner, by addressing the intra-triangle coverage, detecting the coverage holes, and providing lower and upper bounds for coverage holes. Our findings are significantly important for many IoT large-scale coverage applications to either tolerate or call a healing procedure to gap the coverage loss.

The results reinforce the importance of cooperative sensing coverage of multiple sensing providers. Collective IoT sensing nodes not only improve the percentage sensing coverage, but also enhance the identification of the bounds of coverage holes among these networks. This study also

shows the possibility that heterogeneous networks which provide cooperative sensing coverage can expand their lifespan by preserving energy while maintaining the average sensing range at a desired level.

# Chapter 4

## The Impact of Anchor Misplacement on Localization

Localization (i.e., location discovery) in Wireless Sensor Networks (WSNs) has recently attracted a great deal of interest in the research community [78]–[81]. Like sensing coverage that we addressed in the previous chapter, the interest of localization service is expected to grow since localization in WSNs is the cornerstone of many IoT applications, such as smart buildings, smart vehicles, smart homes, wildlife and environment monitoring, military, health care, and merchandise tracking [82], [83]. Large-scale and dense WSNs pose several challenges to localization systems, including robustness, scalability, accuracy, energy consumption, and interoperability. In this chapter, we consider two of these challenges, namely accuracy, and scalability. Accuracy cannot be achieved without creating an error model that truly reflects the essence of different types of errors. These localization errors can be divided into three classes as follows [84]: 1) Setup errors, which occur because of the misplacement of anchor nodes, this may be caused by many reasons such as: errors in manual configuration, network scale in terms of density and size, environmental factors such as the wind, rain, water current, and soil erosion and natural factors like wildlife disturbing the terrain, 2) Measurement errors, which are induced by environmental parameters such as the availability of Line-of-Sight (LoS), obstructions, humidity, and temperature. Furthermore, measurement errors can also be caused by the limitations of the sensing and communication technologies used such as the antenna in Received Signal Strength Indicator (RSSI) techniques which by itself, can be noisy causing error in distance measurement, and 3) Algorithmic errors, which are induced by the nature of the localization algorithm used. One example is, algorithms that seek to achieve good global localization by getting good local accuracy are most likely to have this kind of error. Another example is concerning the limited power in WSNs. This may lead to reducing the complexity of the algorithm as a trade-off with localization accuracy.

Small margins of error may result in trivial localization error; however, it can reach to significant levels as propagated error becomes larger through multi-hop localization. Therefore, having good accuracy depends on removing or at least mitigating the effects of the aforementioned three classes of errors. In this chapter, we delve into the analysis of setup error. We focus on the effect of anchor misplacement on localization of sensing objects in large scale network such as IoT.

In WSNs, the monetary and deployment costs are significant since such costs hinder the expansion of the network. That is why cooperative WSNs is a very important approach in IoT. In this approach, different sensing nodes which belong to multiple owners can cooperate to achieve reliable services with a reasonable level of accuracy. It is generally accepted that increasing the anchor density will trigger the granularity of the localization region to become finer thus reducing localization errors. However, this may not be viable and may even be inadequate in noisy environments. On the other hand, it has been shown that having the minimum number of anchors (i.e., anchors placement) is a NP-complete problem by using the dominating set problem as a base of transformation [85]. This motivates the research for heuristics and integer linear programming algorithms to find near-optimal solutions for the problem. Uniform placement of a minimal number of anchors typically yields high localization accuracy. However, anchor misplacement results in localization errors. Anchor misplacement refers to the problem where the anchor node B is in a specific position, but thinks it is in a different position. The main focus of this chapter is to investigate the effects of anchor misplacement on localization accuracy. We provide a distributed algorithm to mitigate these effects in WSNs.

The chapter is organized as follows. We provide the motivations and contributions in Section 4.1. Section 4.2 overviews the fundamental works related to our research. Section 4.3 presents the problem formulation. Section 4.4 is devoted to studying in detail the effects of anchor misplacement on localization accuracy. Our proposed detection and mitigation algorithm is explained in Section

4.5. Section 4.6 presents simulation results and performance of our proposed algorithm. Section 4.7 concludes the chapter and discusses future research directions.

## 4.1 Motivations and Contributions

Let $s_i$ be a sensing node with an unknown location. Localizing $s_i$ using multilateration requires $K$ distance measurements, $\hat{d}_{ij}$, to anchor nodes, where $K \geq 3$. This forms $K$ circles, $C_k$, where $1 \leq k \leq K$ , with anchor nodes as centers and distance measurements as radii. In the ideal case where no presence of any type of error, the intersection point of these circles is the estimated location of $s_i$ which is also the actual location. In practice, there are always error components that affect the location discovery process. These components cause the multilateration circles not to intersect in one point. Instead, the intersection will be an area $A$ that is enclosed by these circles. In other words, $A = \bigcap_{k=1}^{K} C_k$ and in this case, the location of $s_i$ can be estimated by some methods such as least square error to minimize the localization error. Figure 4.1 shows with a misplacement of anchor $B_2$ to a position $B'_2$, yet it thinks that it is still in the declared position $B_2$.

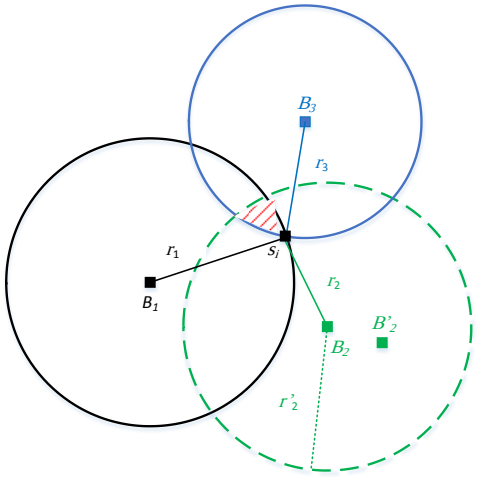



**Figure 4.1: Trilateration method with misplacement error.**

The size of area $A$ depends on the magnitude of anchor misplacement. So the larger magnitude of misplacement, the larger the area is and, consequently, the bigger localization error. In fact, localization may fail to find estimated position if anchor misplacement cause these circles not to

intersect at all. Figure 4.2 shows the impact of anchor misplacement where the localization error increases proportionally with respect to the displacement value and the number of misplaced anchor nodes. In this research, we analyze the problem and provide a distributed scheme that is practical in large-scale networks, and we consider also the drawbacks of other schemes.

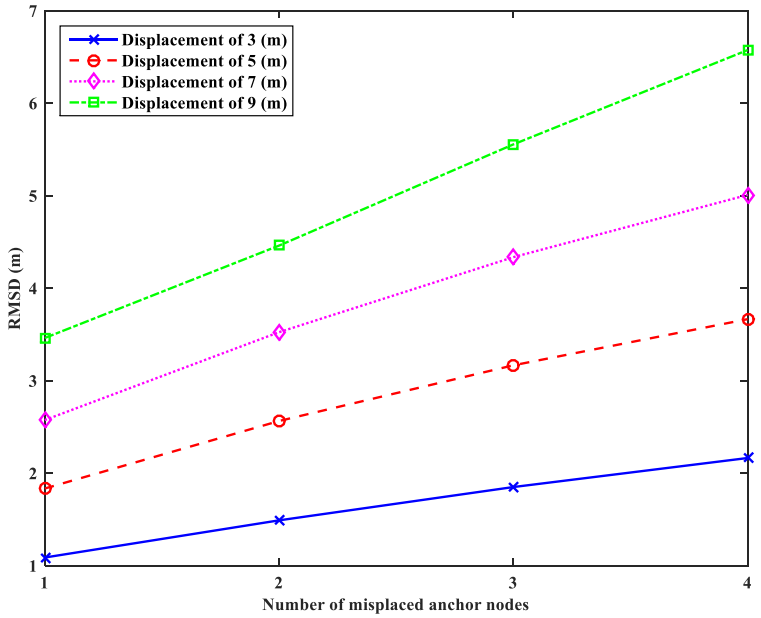



**Figure 4.2: Localization error vs displacement value.**

In this chapter, we study the effects of anchor misplacement on localization. Our main focus is on enhancing localization service by removing the impact of anchor misplacement. This chapter focuses on designing a robust scheme that is able to detect and then mitigate the impact of uncertainty in the positions of anchor nodes. This would highly improve localization service and make it more reliable. We make the following contributions: 1) We investigate the impact of anchor misplacement on location discovery processes, and 2) we analyze the problem and provide a distributed algorithm to detect misplaced anchor nodes. Subsequently, we conduct simulation experiments to evaluate the performance of our proposed algorithm.

## 4.2 Related Work

Most of the localization research in the field of WSNs has been devoted to studying the measurement errors. The measurements, as addressed in Chapter 2, can be obtained using one of the following techniques [86]: Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), or Received Signal Strength Indicator (RSSI). Thus, localization research varies according to the measurement method used. For instance, [87] focuses on RSSI measurement error while the research in [88], and [89] are dedicated to model ToA measurement error. The RSSI-based localization is appealing as it does not require any additional hardware and the distance can be derived from models such as path loss model. The uncertainty of anchor location has been overlooked. It is assumed that anchor nodes have correct positions which is not true. However, very few attempts addressed the localization error caused by anchor misplacement. For example, the authors in [90] assume that the anchor nodes are deployed with some uncertainty in their locations. They propose to localize the misplaced anchor nodes in order to reduce their effect on localization. The authors use Cramer-Rao bound to obtain a lower bound for localization error in multi-hop topologies under the presence of anchor misplacement. Fan et al. [91] propose an approach to detect misplaced anchor nodes and disregard their inputs in localization. They address two issues: ranging error resistance and anchor misplacement. They assume that misplaced anchor nodes broadcast their old positions. The authors propose an algorithm to classify anchor nodes into "misplaced" and "correct". They use the following threshold inequality:

$$\frac{|\hat{d}_{ij} - d_{ij}|}{\hat{d}_{ij}} < \omega \qquad \textbf{(4.1)}$$

Where $d_{ij}$ and $\hat{d}_{ij}$ are the declared and measured distances, respectively. For instance, assume two anchor nodes $B_i$ and $B_j$. $d_{ij}$ is based on their original position (where do they think they are), and $\hat{d}_{ij}$ can be estimated by RSSI based on their physical (i.e., current) positions. $B_i$ and $B_j$. are both "correct" (i.e., not misplaced) if inequality (4.1) holds. Otherwise, at least one of them is

"incorrect". The threshold $\omega$ is a pre-defined value which depends on several parameters such as channel noise and the average displacement value. This detection procedure is incremental. It connects two anchor nodes $B_i$ and $B_j$ if they are correct according to formula (4.1). It starts with connected pair of anchor nodes. New anchor node $B_t$ is added to this construct if it fulfils inequality (4.1) with all connected anchor nodes so far. This eventually will grow to construct the largest all-connected component [5]. This algorithm has advantages, but also has critical drawbacks. Most notable is that it requires that a new anchor node being added should be linked to all anchor nodes of the largest all-connected component that have been formed so far. This condition provides a highly accurate filtering of the misplaced anchor nodes; however it also increases mistaken anchor nodes as even the correct anchor nodes have differences between their declared distances and estimated distance due to the inherent inaccuracy in the measurement techniques such as RSSI. Anchor misplacement may cause disconnectivity in the network of anchor nodes. This leaves more than one large connected component. The algorithm is not feasible in large-scale network such as IoT because it is unrealistic in terms of one-hop communications among every pair of anchor nodes in the network.

In our proposed approach to detect misplaced anchor nodes, similar to Branch and Bound (BB) algorithm [92], we allow the all-connected component to branch and connect a new candidate anchor node, $B_t$ , only if $B_t$ fulfills formula (4.1) with all its neighbours which already part of all-connected component. Our proposed algorithm is a distributed as the branching occurs in the neighbourhood of each candidate anchor node. This should also reduce the energy consumption as the overhead communication among anchor nodes will be reduced as well. It decreases the mistaken anchor nodes and maintain a good detection of the misplaced anchor nodes.

[5] *All-connected component is a subset of the anchor nodes, and in which each node is connected to every other node using a single-hop communication.*

## 4.3 Problem Definition and System Model

Although the deterministic deployment of SNs, viz. grid deployment, seems to achieve minimum number of SNs and a high degree of coverage and connectivity [93] [61], random deployment remains most feasible in IoT context. Random deployment refers to the case where each sensor node is deployed uniformly over the terrain and independently from all other SNs. Let $N = \{s_i: s_i \text{ is an sensing node}; 1 \leq i \leq n\}$, be a set of SNs with unknown location $(x_i, y_i)$ in a plane. Assume we have another set $M = \{B_j: B_j \, is \, an \, anchor \, node; n + 1 \leq j \leq n + m\}$, of anchor nodes in a plane with known location $(x_i, y_i)$.The subscripts $i$ and $j$ in $s_j$ and $B_j$ are identifiers of each sensor and anchor in the terrain, respectively. Furthermore, assume that each anchor node $B_j$ transmits a signal, which contains its location, to the neighborhood. Let $a$ be a node (either sensor or anchor node) in the network. We define $NH(a)$ to denote the neighborhood of $a$, i.e., $NH(a) = \{b | d(a,b) \leq R_t, a \neq b, b \in N \cup M\}$, where $d(a,b)$is the Euclidean distance between $a$ and $b$, and $R_t$ is the transmission range of a node. Each node that receives this information is able to estimate its distance from the emitting node.

IoT involves a large density of sensing objects most of which are placed randomly. So we adopt the random geometric graph (RGG) to represent the random deployment and varying sensor densities [94]. Let $G(V, R)$ be a RGG, where $V$ is the set of sensor and anchor nodes (i.e., $V = N \cup M$). The common model for RGG is the disc model where two nodes $a$ and $b$ are connected if they are in the range of $R$ from each other (i.e., $a \in NH(b)$ and $b \in NH(a)$). Anchor misplacement can cause error in disc model to connect two nodes while they may actually be disconnected. This is described next.

## 4.4 The Effects of Anchor Misplacement

Suppose that anchor node $B_j$ is misplaced to $B'_j$ , and let $M' \subseteq M$ be the set of misplaced anchor nodes. The neighbourhood of anchor node $B_j$ is the set of all anchor nodes that are connected with

$B_j$. That is, $NH_B(B_j) = \{W | W = NH(B_j) \cap M\}$. We differentiate the following three classes of sensing nodes: 1) Set $S_1 = \{s_i | s_i \in NH(B_j) \backslash NH(B'_j), j \in M'\}$ in which the SNs belong to the vicinity of anchor node $B_j$ only before its misplacement. For example, in Figure 4.1, $S_1$ is not empty if $r'_2 < r_2$, 2) Set $S_2 = \{s_i | s_i \in NH(B_j) \cap NH(B'_j), B_j \in M'\}$, represents SNs in the vicinity of the anchor node $B_j$ before and after its misplacement, and 3) Set $S_3 = \{s_i | s_i \in NH(B'_j) \backslash NH(B_j), B_j \in M'\}$, represents the counterpart of set $S_1$. In this set, SNs belong to vicinity of the anchor node $B_j$ only after its misplacement. For example, in Figure 4.1, $S_3$ is not empty if $r'_2 > r_2$ .

The subsequent localization errors of anchor misplacement can be calculated by averaging the localization error of every affected sensor node. Let $\hat{p}_i$ and $\hat{p}'_i$ be the estimated positions of $s_i$ before and after anchor misplacement occurred, respectively. We study the impact of anchor misplacement on localization over the three classes $S_1$, $S_2$, and $S_3$. Sensing nodes that belong to class $S_1$ are misplacement-error-free because they are physically out of the communication range of $B_j$ which mean that $B_j$ has no impact on their localization. For both classes $S_2$ and $S_3$, let $p_i$ be the actual position of $s_i \in S_2 \cup S_3$. The localization error of $s_i$ before anchor misplacment takes place is $e_i = |\hat{p}_i - p_i|$ and the localization error after anchor misplacement occurs is $e'_i = |\hat{p}'_i - p_i|$. Therefore, the impact of anchor misplacement on localization is given by $Abs(e_i - e'_i)$ [6], where Abs($d$) denote the absolute value of $d$. To calculate the whole effect on localization accuracy, we average the localization error of all SNs in class $S_2 \cup S_3$as follows: $E = \frac{1}{|S_2 \cup S_3|}\sqrt{\sum_{i \in S_2 \cup S_3}(e_i - e'_i)^2}$, where $|S_2 \cup S_3|$ denotes the size of the set $S_2 \cup S_3$. Note that $B_j$ has no impact on localization of $s_i$ if it is misplaced to any point on the circumference of the circle centered in $s_i$ and has a radius equals to the distance between $s_i$ and $B_j$ before the misplacement.

[6] *This result assumes that the other error components such as environment and channel conditions remain unchanged.*

In this thesis we use multilateration for localization and, usually, uses Minimum Mean Square Estimate (MMSE) [40] for fine-tuning the estimated position.

Furthermore, we use the root mean square distance (RMSD) to measure the localization error in the network. RMSD is given by the following formula:

$$RMSD = \sqrt{\frac{\sum_{j=1}^{n}\left(\left(x_j - x'_j\right)^2 + \left(y_j - y'_j\right)^2\right)}{n}} \quad \textbf{(4.2)}$$

Where $(x_j - y_j)$ and $(x'_j, y'_j)$ respectively are the actual and estimated positions of sensor node *j* and *n* is the total number of sensor nodes. RMSD is widely used in the literature for the comparison of the estimation error in different localization algorithms, e.g., reference [55].

## 4.5 Mitigating the Impact of Anchor Misplacement

In this section, we discuss our approach to detect the misplaced anchor nodes. Then, we propose our algorithm to find these anchor nodes.

### 4.5.1 Detecting the Misplaced Anchor Nodes

Assume an anchor node $B_j$ is displaced from its original position, $B_{j,d}$ with coordinates$(x_j, y_j)$, to a new position $B_{j,d'}$ with coordinates$(x'_j, y'_j)$. However, $B_j$ it still broadcasts its original position. On the other hand, $B_j$ is still able to communicate with other nodes through, probably, multi-hop communication. Let $s_i$ be an IoT sensor node such that $s_i \in NH(B_j)$ . However, it may happen that $s_i \notin NH(B_j)$after the misplacement of $B_j$. In this scenario, the disc model will mistakenly consider $B_j$ as a candidate to localize $s_i$ which hinder the accuracy of localization.

The severe consequences of anchor misplacement necessitates detecting the misplaced anchor nodes and prevent them from sending wrong coordinates.

Next, we provide some auxiliary definitions, then propose our algorithm to detect misplaced anchor nodes. The set of anchor nodes along with their connectivity can be represented as RGG with $G_B = (V_B, R_t)$ , where $V_B = M$ is the set of anchor nodes (with their current positions) and

$R_t$ is the transmission range. A component $C$, of a graph $G_B$, is connected if there is a path (probably multi-hop) between any two anchor nodes in $C$ .. That is, let $C = \{S = (B_1, B_2, \dots, B_k)\}$, $C$ is connected if $\forall B_i$ and $B_f \in S, \exists\ pth\ of\ connected\ anchors\ B_j, B_{j+1}, B_{j+2}, \dots., B_{j+z}: B_i \in NH_B(B_j)\ and\ B_f \in NH_B(B_{j+z}),\ B_i \neq B_j, j+z < m$, where $m$ is the size of $M$.

**Definition 4.1:** The neighbourhood of component *C* is the union of the neighbourhood of all anchor nodes in this component. That is, $NH_C(C) = \{B_r\ | B_r \in \bigcup_{B_i \in C} NH_B(B_i), B_i \neq B_r, B_r \notin C\}$.

**Definition 4.2:** Two anchor nodes $B_i$ and $B_j$ are threshold-consistent if $\frac{|\hat{d}_{ij} - d_{ij}|}{\hat{d}_{ij}} < \omega$, where the actual distance $d_{ij}$, between $B_i$ and $B_j$, based on their original positions (where do they think they are), and the measured distance $\hat{d}_{ij}$ which can be estimated by RSSI method.

**Definition 4.3:** An anchor node $B_i$ is threshold-consistent with component $C_i$ if the following condition holds: $\forall B_i \in C_i$ , if $B_j \in NH_B(B_i)$ , then $B_i$ and $B_j$ are threshold-consistent.

We are ready now to present our proposed algorithm.

**Algorithm 4.1:** Distributed Anchor Detection (DAnD) Algorithm

**Input:** random geometric graph $G_B = (V_B, R_t)$ of anchor nodes
**Output:** largest component of $G_B$ that is threshold-consistent

```
1  for each anchor node B_i ∈ V_B
2      Initialize component C_i by adding B_i to it;
3  end for
4  for each element B_j ∈ NH_C(C_i)
5      if B_j is threshold-consistent with C_i, then
6          add B_j to C_i;
7      end if
8  end for
```

The computational complexity of DAnD algorithm is $O(mz)$, where $z$ is the average number of neighbours, and $m$ is the number of anchor nodes. The worst case occurs when all anchor nodes can communicate with one another (i.e., $z = m - 1$); this is unlikely to happen in dense networks since the transmission range of the nodes is limited. However, the number of comparisons in the above algorithm can be reduced by half if redundancy is avoided. Thus, if $B_i$ checks the detection condition with $B_j$, then we prevent $B_j$ to check the condition again with $B_i$. There is no extra

communication overhead involved in executing this algorithm as it utilizes only the location information of local anchor nodes.

### 4.5.2 Dealing with the Detected Misplaced Anchor Nodes

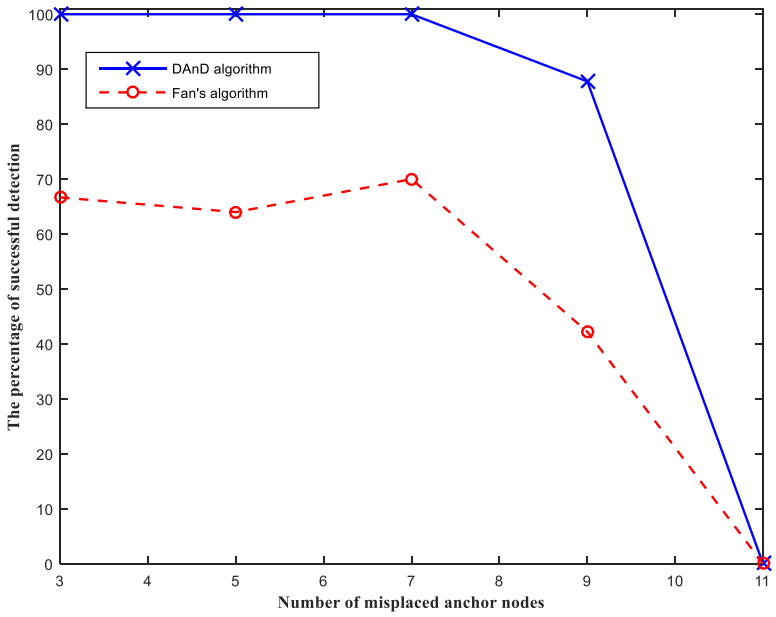
The overlooking of misplaced anchor nodes results in poor localization accuracy. Therefore, the errors triggered by these anchor nodes should be mitigated. The mitigation process first detects the misplaced anchors nodes, then applies one of the following solutions: 1) correct their positions by localizing them using correct anchor nodes, 2) discard them from participating in localization or, 3) correct some misplaced anchor nodes and discard the others. Choosing which option to follow depends on the size of the network, the density of anchor nodes, and the number of misplaced anchor nodes. For example, if the ratio of misplaced anchor nodes is high and the target field is harsh and inaccessible, then correcting the misplaced anchor nodes, by localizing them, would be a practical option in this case.

## 4.6 Performance Evaluation

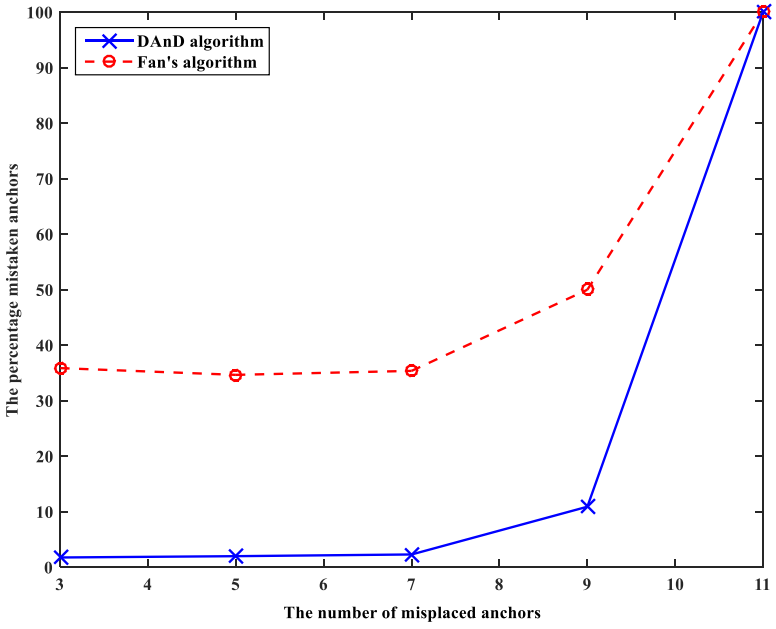
We use network simulator, NS-3, to study the impact of anchor misplacement on localization accuracy. The localization accuracy refers to the average difference between the true position and estimated position of the unknown SNs. We are interested in calculating the localization accuracy under the presence of anchor misplacement problem. For more realistic deployment, the terrain is divided into square grid cells. Each cell area is set to be 100X100m$^2$. *n* unknown SNs and *m* anchor nodes are randomly deployed in each cell. In the experiments, unless stated otherwise, we set the number of cells to 4 cells and the threshold $\omega = 0.01$. IoT sensing nodes and anchor nodes per cell to be 20 and 5, respectively. The transmission range is set to be 160m to guarantee covering all sensing nodes in the cell. Moreover, we conduct all experiments under a fixed value of anchor misplacement (i.e., displacement offset) of 7m on both *x* and *y* coordinates and the misplacements occurred in the same direction. Furthermore, misplaced anchor nodes are randomly selected.

In all experiments, we study the effect of the following parameters: 1) number of misplaced anchor nodes, 2) transmission range, and 3) error threshold. The performance metrics are: A) the successful detection rate of the misplaced anchors, B) the mistaken anchor nodes rate of the correct anchor nodes and, C) the localization error. The successful and mistaken detection ratios of the misplaced anchor nodes are calculated. Then, we calculate the localization accuracy for two cases: before and after applying our proposed algorithm, i.e., DAnD. To calculate the localization error, in each run, we compute the average location error. We use multi-lateration-based localization with MMSE to estimate sensing nodes' positions. The results are compared to the performance of the algorithm in [91]. For the sake of simplicity, let us name this algorithm Fan's algorithm according to the name of its first author. Fan's algorithm was chosen because it is one of the fewer algorithms that addressed the anchor misplacement in localization of WSNs. It considers also a WSN with Gaussian measurement error, which exactly matches our case. Figure 4.3(a) shows the effect of the number of misplaced anchor nodes on the performance of DAnD algorithm (i.e., our proposed algorithm) versus Fan's algorithm. The ratio of successful detection of both algorithms remains constant when the number of misplaced anchor nodes is less than 7, with almost 100% for DAnD and barely 70% for Fan's algorithm. These values reflect the high capability of DAnD algorithm, compared to Fan's algorithm, to reach all anchor nodes and test them against detection condition. As the number of misplaced anchor nodes becomes larger than 7, the successful detection ratio for both algorithms starts declining until they reach zero at 11 misplaced anchor nodes. The analysis of this is as follows: Both algorithms work on selecting the largest component of anchor nodes network that contains only threshold-consistent anchor nodes. If an anchor node belongs to this component, then it is considered to be correct. Otherwise, it is considered to be misplaced. Therefore, starting from 11 misplaced anchor nodes, which are more than half of the total number of anchor nodes, the largest component is the component that includes the 11 misplaced anchors. Thus, the successful detection rate is zero. The ratio of mistaken anchor nodes, in Figure 4.3(b),

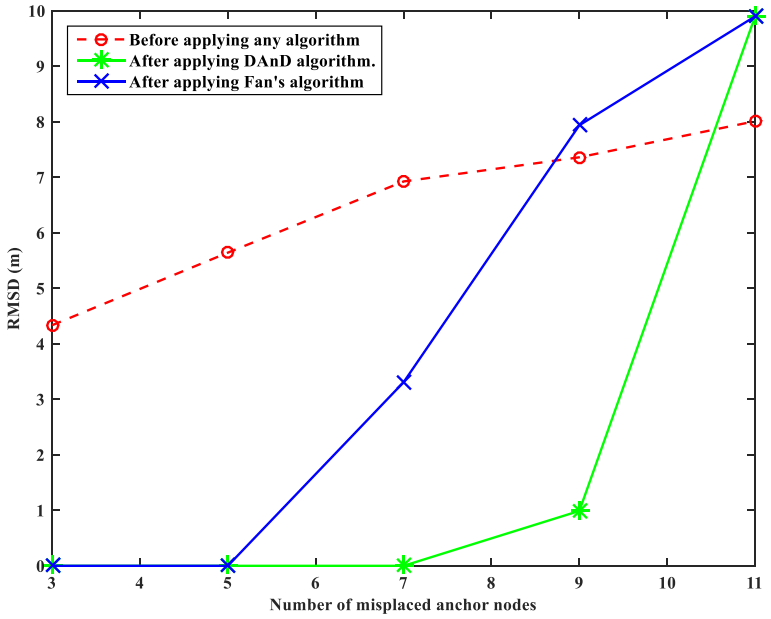
starts very close to zero percentage for DAnD and with approximately 30% on average for Fan's algorithm. The reason behind this result is that, unlike DAnD algorithm, Fan's algorithm fails to test some correct anchor nodes. Both algorithms continue with this rate until it reaches 7 misplaced anchor nodes. After this point, the ratio increases until it reaches 100% at 11 misplaced anchor nodes for both schemes. This percentage represents the complements of the successful detection percentage when more than half of the anchor nodes are misplaced. Figure 4.3(c) shows the effect of number of misplaced anchor nodes on RMSD of the whole network before and after applying both algorithms. The algorithms start off with zero value of localization error. However, the error increases after 5 and 7 misplaced anchor nodes for Fan's and DAnD algorithms, respectively. The RMSD increases sharply in Fan's algorithm as the successful detection of misplaced anchor nodes declines dramatically; however, both algorithms reach the same value of RMSD at 11 misplaced anchors. This is the peak value and is equal to $7\sqrt{2}$. This result is intuitive as in this case the localization of the sensor nodes is calculated only by misplaced anchor nodes, which have been missed during the detection. It is interesting to note that when the number of misplaced anchors passes a certain limit (8 anchor nodes for Fan's, and 10 nodes for DAnD), there is no benefit of running these algorithms since no further error mitigation can be achieved. Thus, when more than half of anchor nods are misplaced, it is better to deploy new anchor nodes and discard all the previous ones in order to achieve better accuracy. This is because the ratio of mistaken anchor nodes becomes larger than the ratio of correct detection.




**(a) The number of successful detection for DAnD algorithm vs Fan's algorithm**

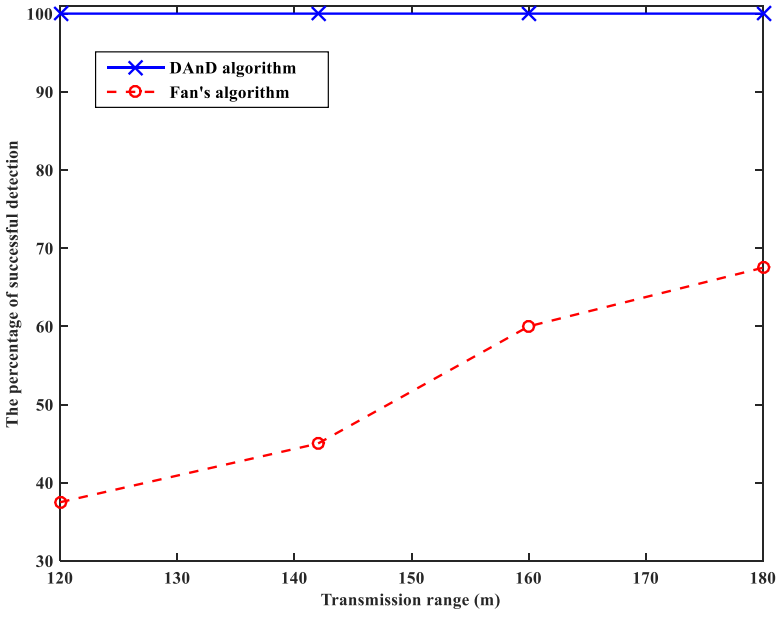



**(b) The number of mistaken detection for DAnD algorithm vs Fan's algorithm**

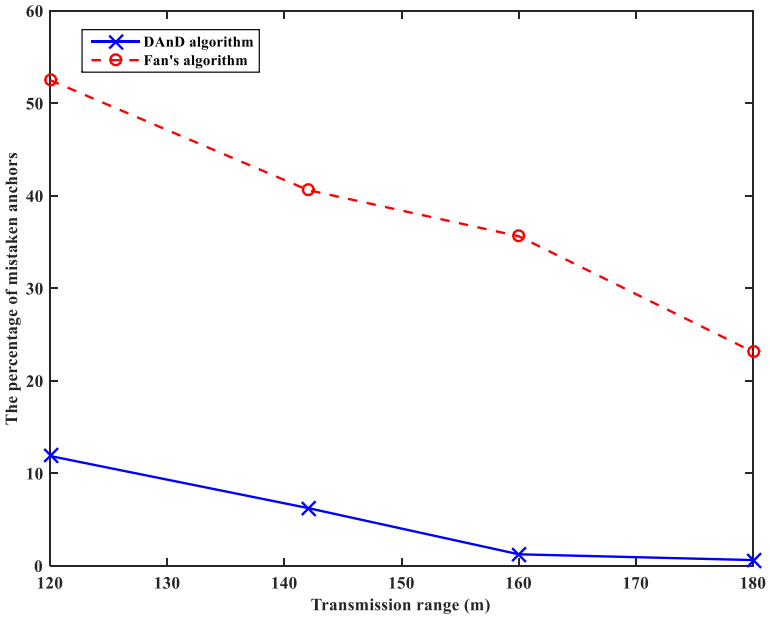



**(c) Localization error vs number of misplaced anchors**

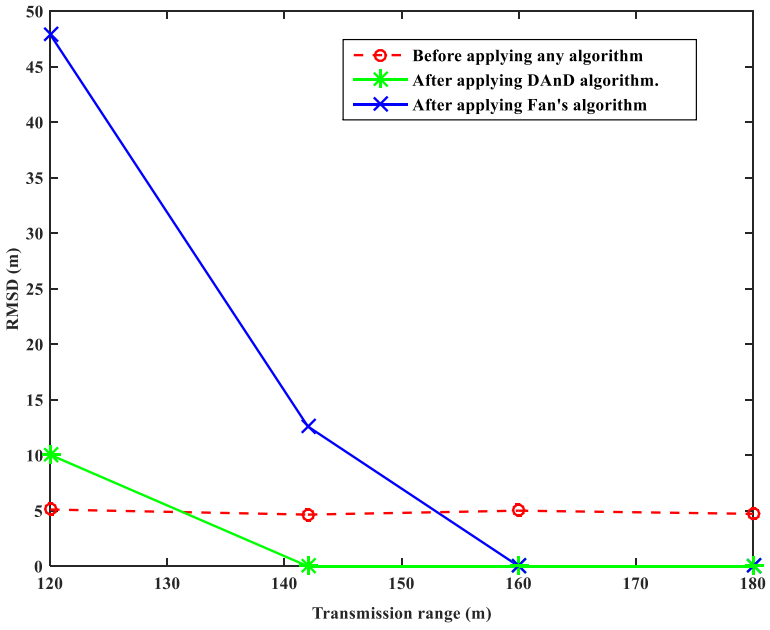
**Figure 4.3: The effect of number of misplaced anchor nodes.**




**(a) The number of successful detection for DAnD algorithm vs Fan's algorithm**



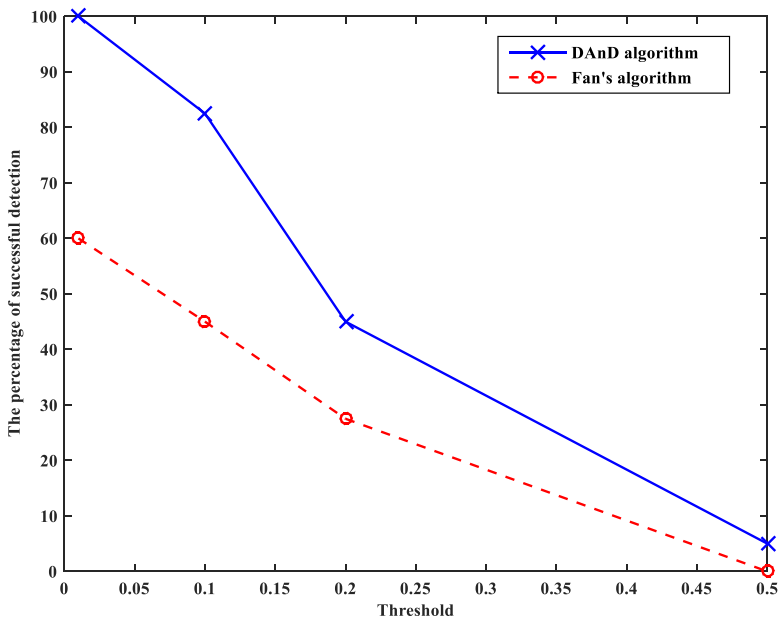

**(b) The number of mistaken detection for DAnD algorithm vs Fan's algorithm**



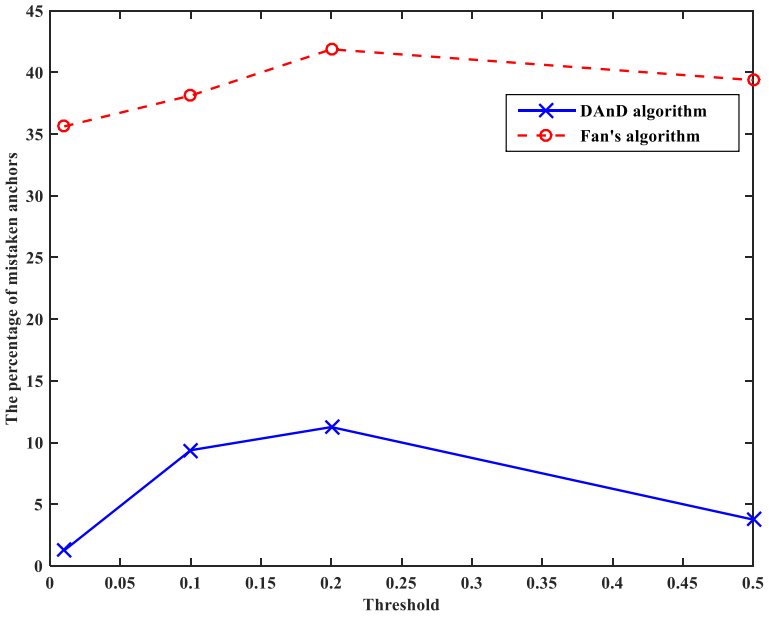

**(c) Localization error vs transmission range**

**Figure 4.4: The effects of the transmission range.**

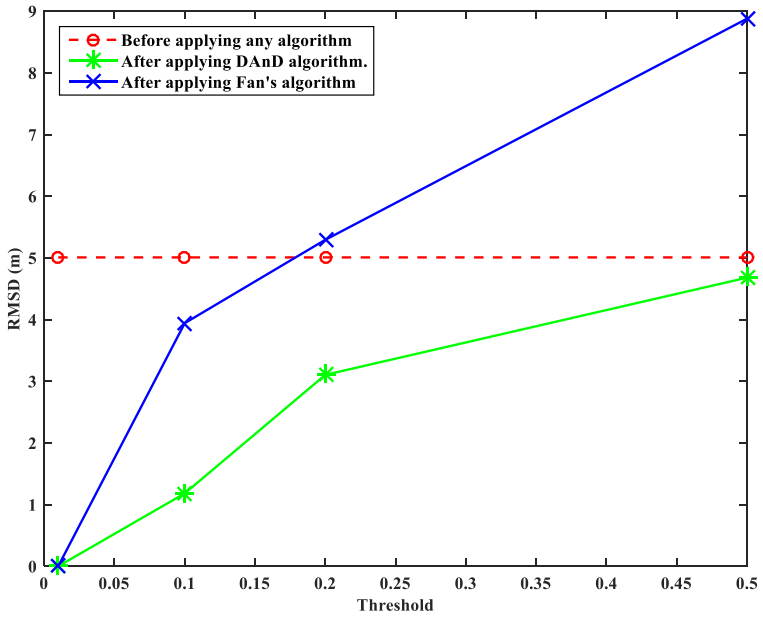
Figure 4.4 shows the effect of transmission range. Figure 4.4(a) shows that DAnD algorithm outperforms Fan's algorithm due to its ability to successfully detect the four misplaced anchor nodes for all presented transmission ranges. The rate of successful detection of Fan's algorithm increases as the transmission range increases but does not exceed 70% in the best case. As expected, the rate of mistaken anchor nodes, Figure 4.4(b), decreases as the transmission range increases. In worst case, our algorithm has around 10% mistaken anchor nodes; while Fan's algorithm, has more than 50% mistaken nodes. Intuitively, the RMSD of the network decreases as the transmission range increases as shown in Figure 4.4(c). It is interesting to see that our algorithm mitigates the error with shorter transmission range. In our next experiment we check the effect of different threshold values on the successful and mistaken detection rates and consequently on the RMSD.




**(a) The number of successful detection for DAnD algorithm vs Fan's algorithm**




**(b) The number of mistaken detection for DAnD algorithm vs Fan's algorithm**




**(c) Localization error vs threshold**

**Figure 4.5: The effects of the threshold.**

Two factors affect the selection of the threshold $\omega$ in equation (4.1): the average displacement value, and the channel quality. Better estimation of these factors results in better choice of threshold value and consequently better detection rate of misplaced anchor nodes. As can be seen in Figure 4.5: The effects of the threshold.(a) when $0.01 \leq \omega \leq 0.02$ , DAnD detects all misplaced anchor nodes making no mistaken anchor nodes. It is expected that when $0 < \omega \leq 0.01$ the algorithm can detect all the misplaced anchor nodes but will also make some mistaken anchor nodes as the threshold becomes smaller, the successful detection becomes more accurate and consequently results in fewer missed misplaced anchor nodes. However, the ratio of mistaken anchor nodes increases in this case. As shown in Figure 4.5: The effects of the threshold.(a) and (b), when $\omega > 0.02$, the successful detection of both algorithms decreases until it reaches zero at some value of threshold that is less than 1. This value depends on the average displacement magnitude of the anchor nodes; the larger the displacement magnitude, the larger the required value of threshold is. Figure 4.5: The effects of the threshold.(c) shows the RMSD of the network versus the threshold. The RMSD of the network without applying DAnD or Fan's algorithms is not dependent on the threshold value. After applying the algorithms, it can be seen that the lowest RMSD belongs to the suitable range $[\omega_{min}, \omega_{max}]$. In this case the suitable range for our experiment is $0.01 \leq \omega \leq 0.02$. Clearly, our DAnD algorithm outperforms Fan's algorithm as it mitigates the error for all presented thresholds. In general, the suitable range $[\omega_{min}, \omega_{max}]$ varies according to the aforementioned factors; however, the same behavior of detection ratio, mistaken ratio, and RMSD are expected for both algorithms.

## 4.7 Summary

In this chapter, we investigated the problem of anchor misplacement in WSNs. Mitigating the impact of anchor misplacement contributes towards accurate and reliable localization service. We address this problem with consideration of more realization of IoT. We propose a distributed algorithm to detect the misplaced anchor nodes. The performance evaluation of our proposed

algorithm outperforms the algorithm presented in [91] in terms of successful detection ratio, mistaken anchor ratio, and localization accuracy.

# Chapter 5

# The Impact of Anchor Misplacement on Sensing Coverage

Recent research considers the placement problem of homogeneous sensing objects to achieve longevity and high sensing coverage. However, recovery and detection of coverage hole(s) has attracted only a few works [61]. Similar to localization in WSN, sensing coverage in WSN lacks the global vision for IoT. For example, the current coverage schemes focus on homogeneous smart objects which belong usually to one owner or one service operator. This void in research shows the need to tackle these issues in large scale networks such as IoT.

In Chapter 3, we addressed the sensing coverage in IoT. We identified coverage holes and provided lower and upper bounds for each hole. In this chapter, we address the impact of anchor misplacement on sensing coverage in the context of IoT.

In this research we: 1) formulate the problem of actual versus perceived coverage, 2) utilize Delaunay Triangulation to provide theoretical analysis for the two types of coverage holes (i.e., actual unreported and false perceived coverage holes) that have been formed as a result of anchor misplacement, 3) develop an efficient algorithm to detect different types of coverage holes, 4) calculate the area ratio of each type of coverage holes to the total area, and 5) implement the algorithm and run experiments to show the correctness of our theoretical analysis. To the best of our knowledge, this is the only research that considers the impact of anchor misplacement on sensing coverage.

## 5.1 Related Work and Motivation

Consider an environmental experiment to measure the air quality in a region as shown in Figure 1.1. For this purpose, the experts use heterogeneous sensing nodes that are already deployed in that region and belong to three different sensing providers. For convenience, Figure 1.1 is

redrawn here as Figure 5.1. These collective shared resources can provide better results in such case and can improve the quality of sensing service.

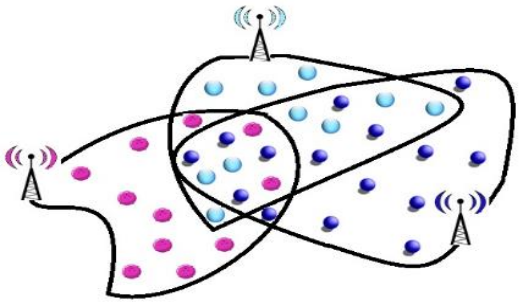


**Figure 5.1: Multiple sensing coverage providers.**

Applications have different sensing coverage requirements. For example, some applications require full sensing coverage such as applications of critical plants, viz a nuclear power plant. Other applications tolerate some coverage holes such as applications in agriculture and weather forecasting. Existing work on sensing coverage in WSNs assumes sensing nodes are homogeneous and only belong to one sensing service provider/owner. Most of the research addresses deterministic placement and deployment planning of sensing nodes to achieve greater coverage and/or to extend the network lifetime [72]. The sensing coverage problem is more pronounced in the IoT context due to the critical challenges of scalability, robustness, heterogeneity, and security [65]. These challenges are normal consequence of the explosive growth of a number of devices with different technologies being introduced globally.

Addressing WSNs in the context of IoT mandates sensing objects to be: 1) heterogeneous as they have different functionalities and capabilities, 2) randomly deployed which is common in IoT, and 3) belong to multiple sensing service providers. The challenge in IoT setting is determining sensing coverage especially under the presence of sensing coverage holes. Anchor misplacement leads to special new types of coverage holes due to inaccurate localization of some sensor nodes. The results in Chapter 4 show that anchor misplacement degrades localization accuracy. Sensing coverage quality is also affected due to: a) inaccurate data collection as the sensed data may contain

inaccurate locations if their corresponding sensing nodes were localized inaccurately using misplaced anchor nodes. This results in coverage holes and hence the full coverage is not preserved and, b) inefficient energy consumption because anchor misplacement leads routing protocols to depend on inaccurate locations. Hence, drains the energy of the sensing nodes. In Chapter 3, we investigate the coverage hole problem in IoT context; where we identified each coverage hole, found its location, and calculated the lower and upper bounds of its area size. Here we study the effect of anchor misplacement and the special types of coverage holes posed by this problem.

## 5.2 Preliminaries

In this section, we introduce some necessary definitions and assumptions. Let $S$ denote the target sensing field. We use the definition of sensing coverage presented in Chapter 2.

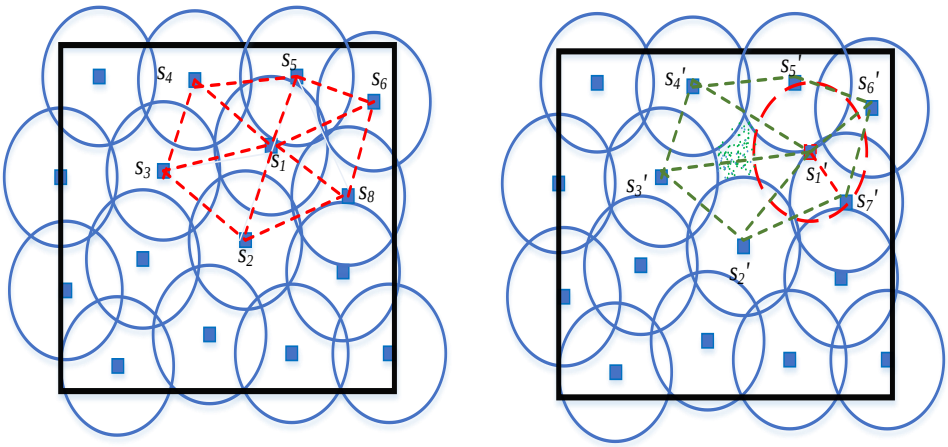
We address different types of coverage holes in the vicinity of the sensing nodes that are affected by anchor misplacement. Assume that sensing node $s_i$ is localized under anchor misplacement. We call $s_i$ in this case an affected sensing node. Let $N' \subseteq N$, be the set of anchor-misplacement-affected sensor nodes or affected sensor nodes, for simplicity. As in Chapter 4, we use the usual cardinality notation |.| to denote the size of the set. For example, $|N'|$ is the cardinality of $N'$. Further let $C_{s_i}$ denotes the *actual* sensing coverage area that is covered by sensing node $s_i$. Next, we introduce some important auxiliary definitions.

**Definition 5.1:** The *actual* collective sensing coverage ($C_{act}$) of all affected sensing nodes in WSN is defined as the union of their physical sensing coverage in the network. That is $C_{act} = \bigcup_{i=1}^{|N'|} C_{s_i}$. Let $s_i'$ be the erroneous estimated location of $s_i$. $s_i$ will report sensed data from inaccurate location which creates a *perceived* coverage around $s_i'$. Further, let $C'_{s_i}$ denotes the *perceived* sensing coverage area that is covered by affected sensor node $s_i$ as if $s_i$ in its estimated position.

**Definition 5.2:** The *perceived* collective sensing coverage ($C_{per}$) of all affected sensing node s in WSN is defined as the union of their *perceived* sensing coverage in the network. That is $C_{per} = \bigcup_{i=1}^{|N'|} C'_{s_i}$.

Clearly the larger the intersection between $C_{act}$ and $C_{per}$, the less perceived coverage exists and, hence, the less impact of anchor misplacement on sensing coverage. The comparison between $C_{act}$ and $C_{per}$ shows new types of coverage holes:

1) Perceived coverage hole where an area is covered actually by $C_{s_i}$ as in Figure 5.2(a), but not covered by $C'_{s_i}$ as in Figure 5.2 (b) where the hole exists in two triangles $s_1's_2's_3'$ and $s_1's_3's_4'$. In this case, $C_{act} > C_{per}$.




(a) (b)

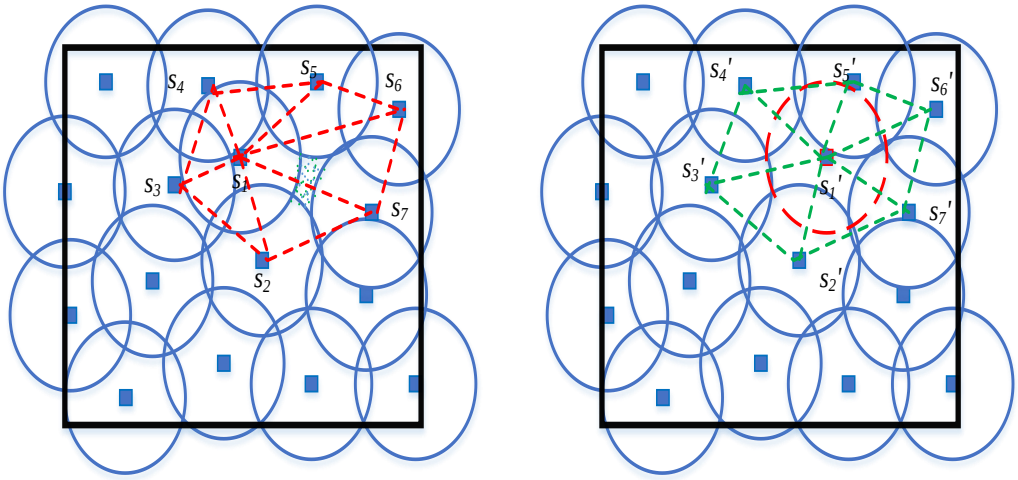
**(a) An ideal case where there is no coverage holes.**

**(b) Perceived coverage hole due to anchor misplacement.**

**Figure 5.2: Perceived hole can be identified by triangulation in the vicinity of the affected sensing node $s_1$.**

2) Unreported actual coverage hole where an area is not covered by $C_{s_i}$ as in Figure 5.3(a) where the hole exists in triangles $s_1s_6s_7$ and $s_1s_7s_2$, but covered by $C'_{s_i}$ as in Figure 5.3(b). In this case, $C_{act} < C_{per}$.

In practise, in order to do a comparison between $C_{s_i}$ and $C'_{s_i}$ and to characterize which scenarios lead to each type of sensing coverage holes, we need a computational structure that enable us to do the required calculations efficiently. For this purpose, we use Delaunay Triangulation to study this problem in the locality of each affected sensor node.




(a) (b)

**(b) Original deployment with actual coverage hole.**

**(c) Actual coverage hole is masked. Thus unreported.**

**Figure 5.3: Actual unreported coverage hole can be identified by investigating the triangles in the vicinity of the affected sensor $s_1$.**

## 5.3 Problem Definition and System Model

In WSNs, sensing node density must be above a specific threshold to maintain coverage; otherwise, coverage holes exist. Anchor misplacement may lead to special types of coverage holes due to inaccurate localization of the affected sensor nodes. The results in Chapter 4 show that anchor misplacement impacts the accuracy of localization. Similarly, coverage quality would be affected due to the inaccurate data collection of the affected sensing nodes since they report sensed data from erroneous estimated locations. This results in two special types of coverage holes: Actual-

unreported, and false perceived coverage holes. In this chapter, we address the impact of anchor misplacement on sensing coverage. Given a random deployment of sensor nodes and a localization error posed by some anchor misplacement on some sensor nodes, we are interested mainly to investigate the new types of coverage holes, and find the size ratio of each type of coverage hole to the total area.

The main assumptions in this chapter are the following:

1) WSN is connected and deployed in 2-D plane. This means that every object is able to receive and send packets to and from any other object. This assumption is important to exchange the information locally through multi-hop in order to build our computational structure in a distributed manner.
2) Localization of sensing objects is multi-lateration-based with MMSE for fine-tuning the estimated positions.
3) In order to show its impact on coverage, we assume that anchor misplacement is the most critical source of error that leads to localization uncertainty (i.e. disregarding other sources of error).
4) The sensing target field is bounded. This helps constructing DT in a simpler manner.

We adopt the same network and sensing models as in Chapter 3. That is, the random deployment for network model and binary disc as a sensing model. Random deployment refers to the case where each IoT object is deployed uniformly over the target field and independently from all other objects. On the other hand, binary disc model assumes that a point in a sensing field $S$ is covered if it is within the sensing range of at least one sensor. Otherwise, it is not covered.

## 5.4 The Effect of Anchor Misplacement on Sensing Coverage

In Chapter 3, we exploit DT to successfully reduce the problem of sensing coverage of a field from testing infinite number of points to discrete ones. Particularly, it is enough to study the problem locally using intra-triangle coverage presented in Chapter 3.

Let $s_i$ be a sensor node vertex in a triangle Δ. Intra-triangle coverage of Δ is given by the following equation: $ITC(\Delta) = \sum_{s_i \in V(\Delta)} CNT(s_i, \Delta)$, where $V(\Delta)$ is the set of three vertices of Δ. $CNT(s_i, \Delta) = \frac{\alpha}{2} {R_s}^2$ denote the coverage contribution of sensor $s_i$ where $\alpha$ is the angle at $s_i$ in triangle Δ and is calculated by the following formula: , $\alpha = cos^{-1}\left(\frac{a^2+b^2-c^2}{2ab}\right)$. Assume that some anchor nodes are randomly misplaced in sensing field *S*. These misplaced anchor nodes pose a localization error on some sensor nodes. We use the implementation of a distributed algorithm in [77] to construct the DT that represents the target sensing field *S*. Lemma 3.2 shows that if $(R_{si} + R_{sj})$ is less than $2r$ of one triangle Δ in DT, then there is a hole coverage in Δ. The uncovered area inside Δ can be calculated by subtracting $ITC(\Delta)$ from the full area size of Δ. That is $A_\Delta - ITC(\Delta)$. The area size of Δ can be calculated by the following formula: $A_\Delta = \sqrt{d(d-a)(d-b)(d-c)}$, where $d = \frac{a+b+c}{2}$ and *a*, *b* and *c* are the length of the sides of Δ. Assume that one sensor node say $s_2$ in Figure 5.4 was localized by some misplaced anchor nodes. Let ${s_2}'(x', y')$ be the erroneous location before any correction. Further, let $\vec{v} = (\Delta x, \Delta y)$ be the localization error vector. Then the coordinates of the corrected position for $s_2$ is $(x' - \Delta x, y' - \Delta y)$.

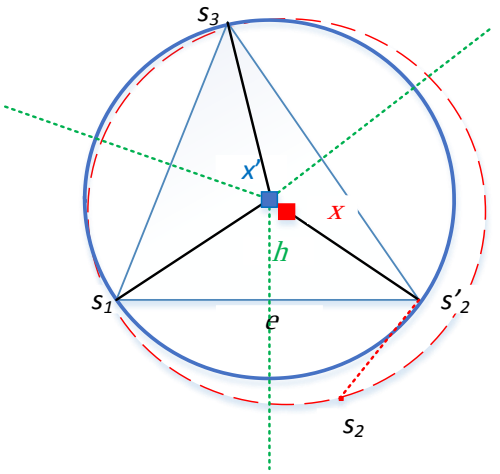



**Figure 5.4: An example of structural change on DT due to correcting the location of ${s_2}'$ to $s_2$.**

In order to measure the sensing coverage holes posed by anchor misplacement, we need to calculate the sensing coverage in two cases: with and without the existence of anchor misplacement. That is, for each affected sensing node $s_i$, we measure the coverage hole by comparing the sensing coverage of $s_i$ and its neighbours from one hand, and $s_i'$ and its neighbours on the other hand. This means we are interested to compare $C_{s_i}$ and $C'_{s_i}$ in their vicinities. In Figure 5.2 and Figure 5.3, the vicinity of affected sensor $s_1$ is $s_2s_3s_4s_5s_6s_7$ and the vicinity of $s_1'$ is $s_2's_3's_4's_5's_6's_7'$. The triangulation of these vicinities enable us to study the coverage holes in each triangle.

Next, we utilize the concept of history in graph theory to demonstrate the above description of calculating the sensing coverage for each affected sensor node with and without the existence of anchor misplacement.

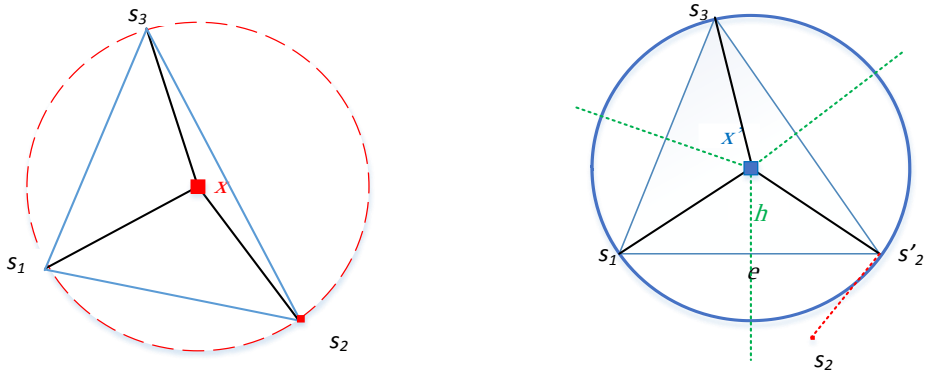
### 5.4.1 Anchor Misplacement as a Graph Operator

Let $D_T$ be a Delaunay Triangulation of IoT objects in the target terrain. Some of these objects have known locations (i.e., anchor nodes) and the rest have initially unknown locations. The anchor misplacement of anchor nodes triggers a change in $D_T$ as they impact the localization accuracy of the sensing objects. The change in $D_T$ could be in the distance metric of edges or in the structure as some objects become connected or disconnected. Denote the new triangulation $D_T'$. Thus anchor misplacement works as an operator which maps a given graph (i.e., $D_T$) into a new graph $D_T'$. Furthermore, let $NH(a)$ denotes the set of sensing objects in which each object has a common triangle edge with sensing object *a*. We utilize the concept of history to approach the Delaunay triangulation structure before and after anchor misplacement. The triangulation of $s_1$ and its vicinity (i.e., $NH(s_1) = \{s_2, s_3, s_4, s_5, s_6, s_7\}$ in Figure 5.2 and Figure 5.3 represents the history of the triangulation of $s_1'$ and its vicinity. The following is the formal definition of the history of a vertex in $D_T$.

**Definition 5.3:** Let $T(s_i)$ denotes a triangulation in $D_T'$ that is induced by both $s_i'$ and $NH(s_i')$. That is, the Delaunay triangles that have $s_i'$ as a common vertex. Similarly, we call $T^{-1}(s_i)$ the history of

a vertex $s_i'$ and it denotes a triangulation in $D_T$ that is induced by both $s_i$ and $NH(s_i)$, where $s_i$ is the correct position of$s_i'$.

Figure 5.5 shows one triangle of T($s_2$) and its history. The concept of history is not new in graph theory and has been used to study the asymptotic characteristics of iterated graphs such as line and path graphs [95] [96]. We note that the locations of all vertices in T($s_i$) are localized with the presence of anchor misplacement. However, the locations in $T^{-1}(s_i)$ are corrected as if there is no anchor misplacement or the misplacement has been reversed. Clearly the subgraphs T($s_i$) and $T^{-1}(s_i)$ may not be the same as some vertices can be in T($s_i$) but not in $T^{-1}(s_i)$ or vice versa. The location of each object in the terrain is a key point in our study as both subgraphs T($s_i$) and $T^{-1}(s_i)$ maybe isomorphic[7] but yet different in terms of edge lengths.



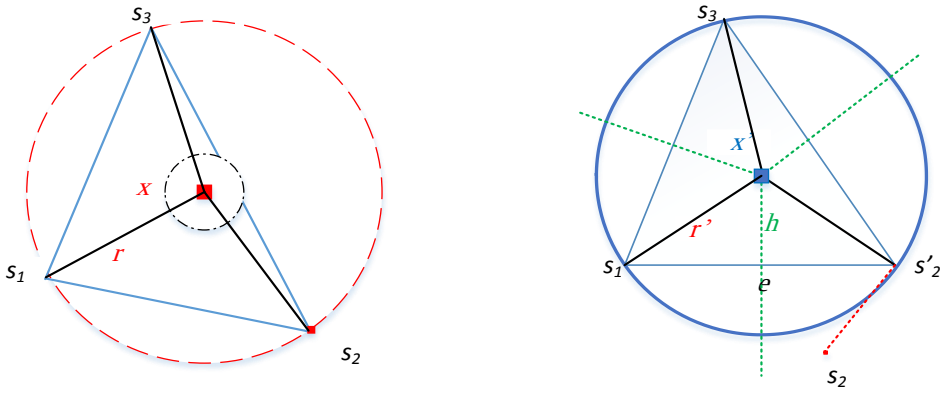

**(a) One triangle of $T^{-1}(s_2)$** **(b) The triangle in T($s_2$)**

**Figure 5.5: A partial snapshot of T($s_2$) and its history.**

We can construct $D_T$ from $D_T'$ in the following way: Identify the misplaced anchor objects by using the algorithm in Chapter 4. Then remove the affected sensing objects $s_i$ with their linked edges, and insert objects $s_i$ again in their correct positions. Lastly, construct the triangulation in their locality. Let $\deg(s_i, G)$ denotes the number of direct neighbouring objects of object $s_i$ in graph $G$. That is, $\deg(s_i, G) = |NH(s_i)|$. The average $\deg(s_i, D_T)$ is at most 6 and, therefore, the average

[7] Two graphs are isomorphic if they contain the same objects (i.e., vertices) linked in the same way.

number of triangles in both subgraphs $T(s_i)$ and $T^{-1}(s_i)$ will not exceed 6 [70]. This shows a low computational cost of our approach.




(a) **Coverage hole in $T^{-1}(s_2)$** **(b) The hole is masked in $T(s_2)$**

**Figure 5.6: Unreported coverage hole with center x.**

### 5.4.2 Coverage Holes with Anchor Misplacement

We study the impact of anchor misplacement on sensing coverage and detect the false coverage and actual coverage holes. To achieve this, we are interested in the common triangles of both subgraphs, that is the triangles in $T(s_i) \cap T^{-1}(s_i)$. The empty intersection indicates that $T^{-1}(s_i)$ is totally new structure and none of $NH(s_i)$ in $T(s_i)$ is triangulated with $s_i'$. This happens when the error posed by anchor misplacement on localizing $s_i$ is extremely high such that the estimated location $s_i'$ is out of the vicinity of $s_i$. Any common triangle $\Delta$ in $T(s_i)$ and $T^{-1}(s_i)$ falls in one of the following categories:

1) The local full coverage of $\Delta$ is maintained in both $T(s_i)$ and $T^{-1}(s_i)$.
2) The local full coverage of $\Delta$ exists in $T^{-1}(s_i)$, but not in $T(s_i)$ (i.e., perceived coverage hole).
3) There is no local coverage in both $T(s_i)$ and $T^{-1}(s_i)$.
4) The local full coverage of $\Delta$ exists in $T(s_i)$, but not in $T^{-1}(s_i)$ (i.e., actual unreported coverage hole as in Figure 5.6).

Categories 1 and 3 deal with extreme cases where triangle Δ is either covered in both T($s_i$) and its history or not. Category 2 shows the case of perceived unreal coverage hole as Δ is not covered in T($s_i$), but is covered in its history. In contrary of category 2, category 4 demonstrates the actual unreported coverage hole.

The two types of coverage holes (i.e., false perceived, and actual unreported coverage holes) can be identified by applying the same strategy followed in Chapter 3. Lemma 3.2 is applied on each triangle Δ in $\mathrm{T}(s_i) \cap \mathrm{T}^{-1}(s_i)$. Once for T($s_i$) and another time for $\mathrm{T}^{-1}(s_i)$. Thus, the coverage hole in Δ is identified according to the above categories. Finding the actual sensing coverage, $C_{s_i}$, and the perceived counterpart, $C'_{s_i}$, in their vicinities requires the calculation of the size of coverage hole for each Δ in $\mathrm{T}(s_i) \cap \mathrm{T}^{-1}(s_i)$. For more readability of this analysis, let $\Delta'$ and Δ denote the same triangle (i.e., have same vertices) in $\mathrm{T}(s_i)$ and $\mathrm{T}^{-1}(s_i)$, respectively.

The coverage ratio in category 1, RC1, can be written as a ratio of the area of $\Delta'$ to the area of Δ. That is, $\mathrm{RC1} = \frac{A_{\Delta'}}{A_{\Delta}}$. On the other hand, the coverage hole ratio in category 3, RC3, can be stated as the size of coverage hole in of $\Delta'$ to its corresponding in Δ. For example, the uncovered area of Δ, denoted by $\mathrm{UNC}(\Delta, \mathrm{T}(s_i))$, is equal to $A_{\Delta} - ITC(\Delta)$. Thus the ratio is, $\mathrm{RC3} = \frac{\mathrm{UNC}(\Delta', \mathrm{T}(s_i))}{\mathrm{UNC}(\Delta, \mathrm{T}^{-1}(s_i))}$. Note that $\Delta'$ in T($s_i$) and Δ in $\mathrm{T}^{-1}(s_i)$ have the same vertices, yet may not be similar in terms of edge length. In both categories 1 and 3, if the ratio is not equal to 1, then there is clearly inaccurate reporting of the sensing coverage of the surroundings of the vertices of Δ (or alternatively$\Delta'$). In this case, the actual coverage in the history is either underestimated or overestimated.

The ratio of perceived coverage hole in category 2 is $RC2 = \frac{\mathrm{UNC}(\Delta', \mathrm{T}(s_i))}{A_{\Delta'}}$. Similarly, the ratio of actual unreported coverage hole in category 4 is calculated by $RC4 = \frac{\mathrm{UNC}(\Delta, \mathrm{T}^{-1}(s_i))}{A_{\Delta}}$.

The aforementioned analysis takes in consideration the case where more than one affected objects are neighbours to each other. Assume $s_i'$ and $s_j'$ are two neighbours and affected objects in

$T(s_i)$ (or alternatively in $T(s_j)$). Then $T^{-1}(s_i)$ (or alternatively in $T^{-1}(s_j)$) contains both $s_i$ and $s_j$. Therefore, the effect of anchor misplacement on both $s_i$ and $s_j$ is reflected in all $ITC$ that contains $s_i$ or $s_j$, or both. However, there will be a redundant calculation of the $ITC$ of the triangle that have both contribution of $s_i$ and $s_j$ as vertices. This redundancy should be considered when calculating the whole $ITC$ in $T(s_i)$ and its history.

## 5.5 Coverage Hole: Ratio and Type Algorithm

The following algorithms identify the different types of coverage holes posed by anchor misplacement and calculate the ratio of each one. The Coverage Hole: Ratio And Type (CHRAT) algorithm assumes that all sensing nodes have been localized, with the existence of anchor misplacement, and their locations are known. CHRAT uses the DAnD algorithm in Chapter 4 to identify the misplaced anchors. Then finds the set of affected sensing nodes that have been localized by at least one misplaced anchor. For each sensing node $s_i$, CHRAT finds the set of collective triangles that have $s_i$ as a common vertex. It finds the set of triangles for each affected sensing node $s_i$ for both $T(s_i)$ and its history $T^{-1}(s_i)$ as shown in line 5 and 6. The intersection set of triangles between $T(s_i)$ and its history $T^{-1}(s_i)$ is calculated in line 7. CHRAT iterates over all triangles in the intersection and invoke Identify Ratio And Type (IRAT) algorithm to identify the coverage hole and determine its ratio.

**Algorithm 5.1:** Coverage Hole Ratio and Type (CHRAT)

**Input:** *M, N;* //the sets of anchors and sensors, respectively.
**Output:** Type_of_coverage_hole, ratio

```
1   G = Construct_Graph(M);// construct graph G //from
    anchor set only.
2   M′ = detect_Misplaced_Anchors(G);//set of misplaced
    anchors
3   N′ = find_Affected_Sensors(M');
4   for each sensor s_i ∈ N′
5   |   T(s_i)   = findTriangles(s_i);
6   |   T^-1(s_i)   = findTrianglesInHistory(s_i);
7   |   int_set(s_i) = T(s_i) ∩ T^-1(s_i);
8   end for
9   for each sensor Δ′ ∈ int_set(s_i)
10  |   Δ   = historyOfTriangle(Δ′ );
11  |   invoke IRAT(Δ′,Δ,s_i);
12  end for
```

IRAT takes a triangle, its history, and its corresponding sensing node as inputs. In lines 2 and 6 *HasCoverageHole(*Δ*)* function checks $\Delta' \in \mathrm{T}(s_i)$ and its history $\Delta \in \mathrm{T}^{-1}(s_i)$ against the coverage criteria in Lemma 3.2. This gives four combinations each which corresponds to the four categories discussed in the previous Section. *current_covered* and *history_covered* are two Boolean variables which indicate whether that the triangles $\Delta'$ and $\Delta$ , respectively, are covered or not. Once the type of coverage hole, if any, is identified, IRAT calculates the ratio of the coverage hole in $\mathrm{T}(s_i)$ to the area of coverage hole in $\mathrm{T}^{-1}(s_i)$.

**Algorithm 5.2:** Identify Ratio and Type (IRAT)

**Input:** triangle $\Delta$, triangle $\Delta'$, affected_sensor $s_i$

**Output:** Type_of_coverage_hole, ratio

*current_covered = history_covered = true;*
**if** *HasCoverageHole*($\Delta', \mathrm{T}(s_i)$) **then**
    *unc_*$\Delta'$ = $\mathrm{UNC}(\Delta', \mathrm{T}(s_i))$;
    *current_covered = false;*
**end if**
**if** *HasCoverageHole*($\Delta, \mathrm{T}^{-1}(s_i)$) **then**
    *unc_*$\Delta$ = $\mathrm{UNC}(\Delta, \mathrm{T}^{-1}(s_i))$;
    *history_covered = false;*
**end if**
**if** (*current_covered && history_covered*) **then**
    *Type_of_coverage_hole* = Category1;
    $A_{\Delta'}$ = *findTriangleArea*($\Delta'$);
    $A_{\Delta}$ = *findTriangleArea* ($\Delta$);
    *ratio* = $\frac{A_{\Delta'}}{A_{\Delta}}$ ;
    *break;*
**end if**
**if** (!*current_covered && history_covered*) **then**
    *Type_of_coverage_hole* = Category2;
    //false perceived coverage hole
    $A_{\Delta'}$ = *findArea*($\Delta'$);
    *ratio* = $\frac{unc\_\Delta'}{A_{\Delta'}}$;
    *break;*
**end if**
**if** (!*current_covered && !history_covered*) **then**
    *Type_of_coverage_hole* = Category3;
    *ratio* = $\frac{unc\_\Delta'}{unc\_\Delta}$;
    *break;*
**end if**
**if** (*current_covered && !history_covered*) **then**
    *Type_of_coverage_hole* = Category4;
    //unreported actual coverage hole
    $A_{\Delta}$ = *findArea*($\Delta$);
    *ratio* = $\frac{unc\_\Delta}{A_{\Delta}}$;
    *break;*
**end if**
**return** *Type_of_coverage_hole, ratio;*

## 5.6 Numerical Results and Discussion

We use NS-3 to simulate different scenarios of the conducted experiments. The outputs of the simulation step are used as input for a Visual Studio C++ program which includes our implementation of the proposed algorithm. The experiments show the effect of the following parameters: 1) number of misplaced anchors, 2) sensing ranges, and 3) the localization error *e* posed by anchor misplacement. The parameters of all experiments are set as follows, unless otherwise stated. The terrain is divided into four square cells. Each of which is 100 X 100 $m^2$. The number of sensor and anchor nodes per cell is 20 and 5, respectively. The transmission range is set to be 142m. All experiments are conducted under a displacement value of 7m for each misplaced anchor. Furthermore, each misplaced anchor nodes are randomly selected. The average IoT sensing range *r*=5m (with variance of 2 m). The results of all conducted experiments are the average of 10 runs. To show the importance of this research, we conduct an experiment that simulates the following a real-life scenario: given 40 objects in the terrain such as gas pipes. These pipes are fully covered by sensing nodes to monitor gas leakage. Given that an anchor misplacement incurred, we are interested in calculating the proportion of the miss-reported objects. In other words, we calculate the percentage of the objects that are no longer reported by their original sensing nodes. The results are shown in Figure 5.7. In the literature, it is widely understood that having more sensing nodes with short sensing ranges provides the best sensing coverage of a given terrain. However, our results show that this is not the case when anchor misplacement occurs. The result show that using fewer sensing nodes with larger sensing ranges provides better monitoring for the terrain as the percentage of miss-reported objects decreases by increasing the sensing range. For a sensing range of 5m, 40 sensing nodes are required to cover the objects, while for larger sensing ranges fewer sensing nodes are needed.

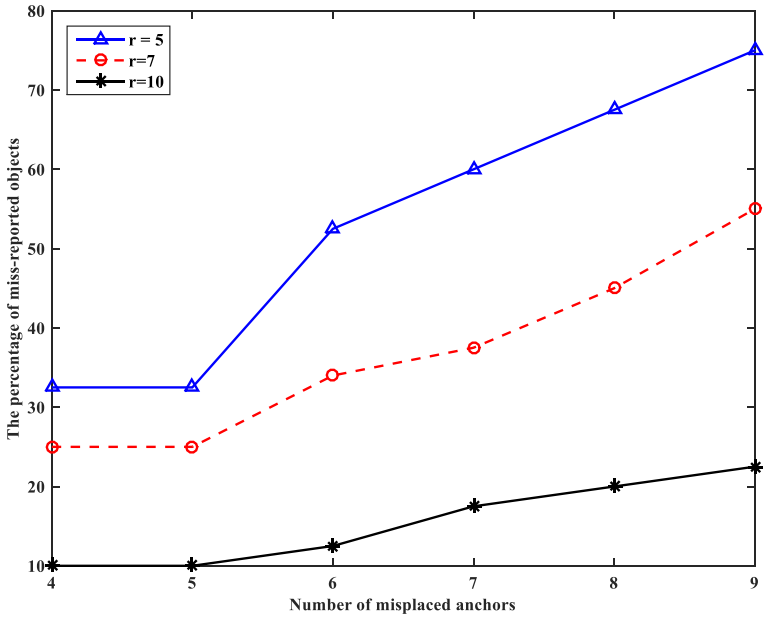



**Figure 5.7: Number of misplaced anchors vs. percentage of miss-reported objects.**

In the second experiment we set the sensing range to 5m (with variance of 2 m) and keep all other parameter settings unchanged. This experiment intends to show the relationship between root mean square distance (RMSD) of the estimated locations of the sensing nodes, percentage of perceived coverage, and the number of anchor nodes. Perceived coverage of an affected sensing node *s* is the size of the area that is mistakenly reported by *s*. In other words, it is the part of the sensing disc around s' (i.e., estimated location) that does not intersect with the actual sensing disc of *s*. We sum up the perceived coverage for all affected sensing nodes and then calculate its percentage to the summation of the actual sensing discs. The result, as shown in Figure 5.8, indicates that the percentage of perceived coverage increases proportionally as localization error increases. The results also show that when half of the anchor nodes are misplaced, an estimated 80% of the reported data is inaccurate.

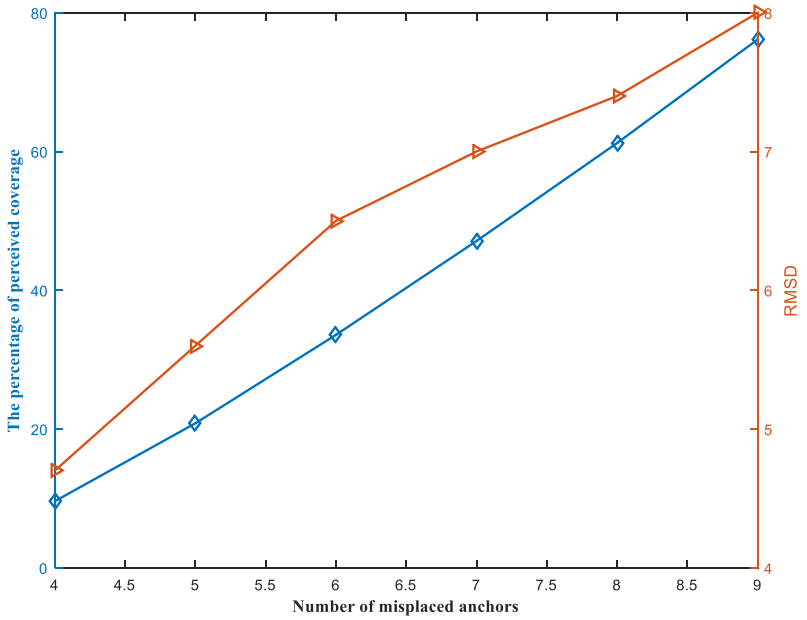



**Figure 5.8: Number of misplaced anchors vs. perceived coverage and RMSD.**

An interesting evaluation metric is percentage of the area of perceived coverage hole to the area of the covered terrain. We are also interested to check on the number of perceived coverage holes posed by anchor misplacement. To evaluate these metrics, we use 100 X 100 $m^2$ grid-based deployment to ensure the full sensing coverage with sensing radius set to 9m. All other parameters are unchanged. The results are shown in Figure 5.9. As the number of misplaced anchor nodes increases, the number of coverage holes and their area percentage increase as well. This is because more misplaced anchor nodes generate more localization errors and, hence, more perceived coverage holes.

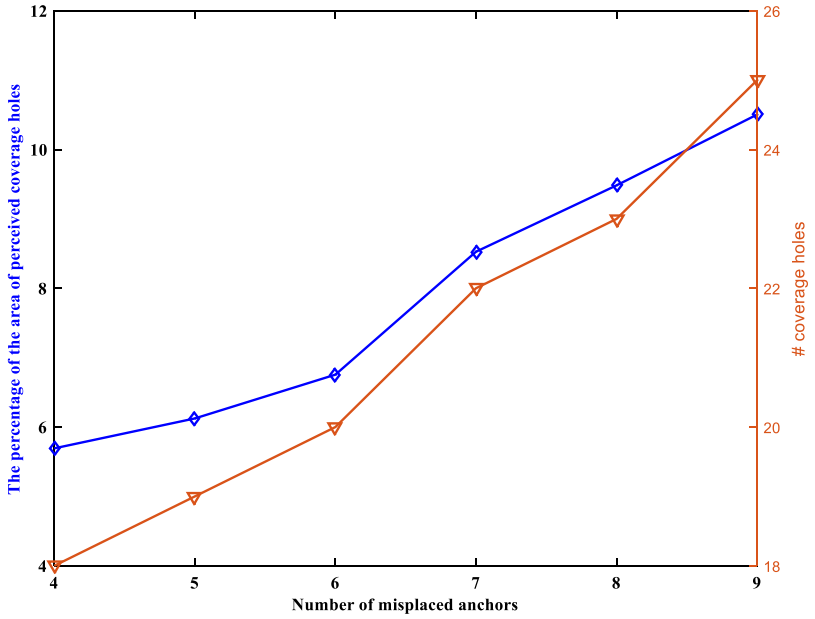



**Figure 5.9: Number of misplaced anchors vs. the percentage of the area of sensing coverage holes and the number of holes.**

These results demonstrate the consistency and the validity of our approach in a typical setting with well-understood sensing coverage parameters.

## 5.7 Summary

The realization of IoT requires investigating sensing coverage again under the characteristics of IoT itself and according to dynamicity of this environment. This research investigates the IoT sensing coverage problem with anchor misplacement. Anchor misplacement leads to new types of coverage holes which degrades the quality of sensing coverage. We consider heterogeneous and non-deterministic deployment of IoT sensing nodes. We exploit a Delaunay Triangulation tool from computational geometry to provide a localized approach to identify the type of coverage hole, and determine its ratio to the total area.

The results show the importance to overcome, rather than overlook, anchor misplacement. The perceived coverage is a serious degradation to the quality of sensing coverage. False sensing reports

posed by affected sensing objects may lead to life loss in cases such as wildfire and chemical and gas leakage. While collective IoT sensing nodes improves the percentage of sensing coverage, anchor misplacement increases the perceived coverage and generates new types of coverage holes. This study also shows that, unlike common belief, having sensing nodes that have a short sensing range can degrade the sensing coverage quality when their locations are inaccurate. Our findings suggest that a larger sensing range with fewer sensing nodes makes the impact of anchor misplacement less severe. This is also more economic in a very large context such as IoT. These findings can be utilized to tune the sensing range to keep the impact of anchor misplacement under control. Heterogeneous networks provide cooperative sensing coverage and can expand their lifespan by preserving energy while maintaining the average sensing range at a desired level.

# Chapter 6

# Measuring the Validity of Sensing Coverage Reporting in the Presence of Anchor Misplacement

The coverage problem is considered an important measurement of the quality of sensor network. It measures to what extent the sensor network can monitor the surrounding physical space. Sensing coverage in WSNs has attracted much research. The requirements of sensing coverage vary according to the application. Some applications require only single-sensing coverage, also referred to as 1-coverage, while other applications require k-coverage, where $k > 1$. The majority of sensing coverage research assumes that the anchor nodes are in correct positions and, therefore, do not pose any error on localization of sensing nodes. For example, in [97] the authors propose an approach for anchor placement to achieve optimal localization with a minimum number of deployed anchor nodes. They assume that the anchor nodes will be placed precisely in the correct position. Taking into consideration anchor misplacement, the analytical results of their research will definitely be different as the results of [61] conclude. The authors of [74] provide a closed formula for equilateral triangle grid-based deployment that achieves full coverage with a minimum number of sensing nodes and tolerates the misplacement of these nodes. Chapter 5 shows the severe impact of anchor misplacement sensing coverage. In Chapter 4, we address the mitigation of the impact of anchor misplacement on localization accuracy. Our findings show that the average localization error increases as the average of anchor displacement value increases.

Misplaced anchor nodes pose different error magnitudes on affected sensing nodes according to displacement value, and the distance between the misplaced anchor node and the sensing node. If some anchor nodes were inaccurately positioned, many of the sensing reports will be invalid. In this research, we investigate the sensing validity in the presence of anchor misplacement with and

without the existence of measurement errors. Given a set of anchor nodes, if some of them are misplaced, can we measure the validity of sensing reports of each sensing node? One interesting aspect of this research is that it addresses the validity of sensing reports in the presence of anchor misplacement in a non-uniform sensing area which represents either a convex or concave set.

The chapter is organized as follows. We provide the motivations and contributions in Section 6.1. Section 6.2 formulates the problem definition. Section 6.3 presents the model of sensing area and the impact of error components on sensing validity. Section 6.4 is devoted to testing the sensing validity. We design an algorithm to classify the sensing reports to either true positive or true negative. Section 6.5 presents simulation results. Section 6.6 concludes the chapter.

## 6.1 Motivations and Contributions

Assume an experiment is being conducted to measure the air temperature and humidity levels in a warehouse. The warehouse is divided into small non-uniform areas in which sensing nodes are intended to monitor these individual areas as shown in Figure 6.1. For this purpose, heterogeneous sensing nodes that are already deployed in all areas will be used. Each sensing node was placed in its "residence" area within the warehouse. So the residence area of a sensing node $s_i$ is the area of the warehouse where $s_i$ supposed to monitor. Several anchor nodes were misplaced or had inaccurate locations, the goal is to measure whether or not the sensing nodes are still valid and convey accurate reporting. This checks the estimated location of each sensing node whether it is within its residence area or not.

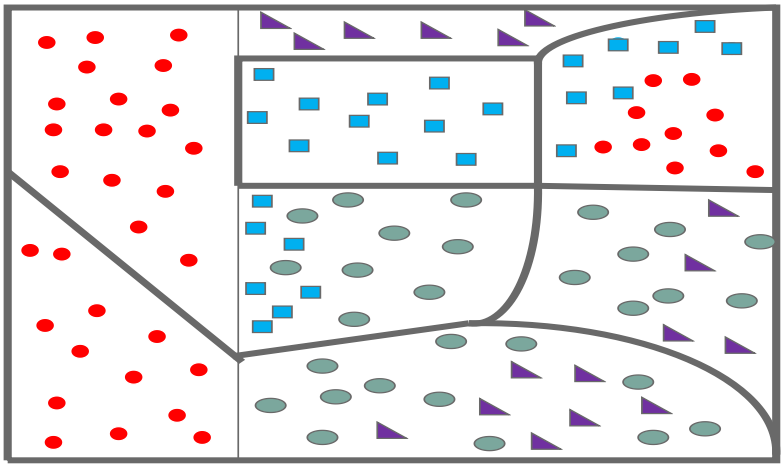


**Figure 6.1: Non-uniform sensing region with multiple sensing providers.**

Applications have different sensing coverage requirements. For example, a nuclear power plant requires full sensing coverage. Other applications tolerate some coverage holes such as applications in weather forecasting. Most of the existing work on sensing coverage in WSNs consider homogeneous sensing nodes, a single service operator, and grid-based deployment for simplicity, and to achieve better deterministic coverage, and to prolong the network lifetime. The sensing coverage problem becomes challenging in IoT context due to scalability, robustness, heterogeneity, and security. Dealing with WSNs in the context of IoT mandates considering the aforementioned challenges. Anchor misplacement affects the sensing coverage quality. The sensed data may contain erroneous locations if their corresponding sensing nodes have been affected by anchor misplacement. In Chapter 5, we investigate, in more depth, the impact of anchor misplacement on sensing coverage in terms of new types of coverage holes. In this chapter, the validity of sensing coverage in the presence of anchor misplacement is addressed. Furthermore, we address how to measure this validity, if it exists.

There are many components of an error that will affect the localization and result in validity issues in sensing coverage. Measurement error and set up error are examples of such components. In this research we: 1) formulate the problem of validity of reporting sensing coverage; 2) utilize triangulation tool to provide theoretical analysis for sensing coverage problem that have been

formed as a result of anchor misplacement; 3) develop an efficient algorithm to test the validity of sensing reporting; and 4) implement the algorithm and run various experiments to show the correctness of our theoretical analysis. To the best of our knowledge, this is the first attempt in the literature to measure the sensing validity in the presence of anchor misplacement.

## 6.2 Problem Definition

We first introduce the necessary definitions and assumptions.

**Definition 6.1:** Let $s_i$ be a sensing object with an unknown location. Let $s_i$ belong to a residence sensing area $A_k$. Then sensing report of $s_i$ is valid (or true positive (TP)) if the estimated location of $s_i$ is still within $A_k$. Otherwise, it is invalid (or true negative (TN)) as the estimated location of $s_i$ is outside of $A_k$.

Given a random deployment of sensing nodes in a non-uniform sensing field, and a localization error posed by anchor misplacement and measurement error on some sensing nodes, we are interested in investigating and modeling the problem of measuring the validity of sensing reporting which depends on the estimated location of the affected sensing nodes. In particular, we are interested in classifying the set of sensing nodes into true positive (TP), i.e., "valid", or true negative (TN) i.e., "invalid", sensing reports.

The main assumptions are the following:

1) Localization of sensing objects is based on multi-lateration with minimum mean squared error (MMSE) for fine-tuning the estimated positions.
2) Anchor misplacement and measurement error are the most critical sources of error that lead to localization uncertainty (i.e., disregarding other sources of error).
3) The sensing target field is a bounded 2-D plane. This ease the construction of triangulations.

We adopt the same network and sensing models in Chapter 3, with random sensor deployment and binary disc sensing model.

## 6.3 Model of Sensing Area

Grid-based deployment of sensing objects provides straight forward analysis of the problem of sensing coverage. For example, [74] provides a closed form of the number of sensing nodes and the spaces between them under equilateral-triangle grid deployment. This is not the case in random deployments where the sensing region has non-uniform areas.

Our approach depends on creating a polygon to represent each sensing area as shown in Figure 6.2, where the different shapes denote the estimated positions of heterogeneous sensing nodes. The region bounded by the grey line represents the sensing area. In order to test the location of each sensing node, we triangulate the representative inner polygon. Then we apply a cross product technique to assess the presence of a point inside a triangle. Consequently, the sensing node $s_i$ is either a true positive (TP) if it is inside any triangle, or a true negative (TN) otherwise. Next, we discuss how to create the inner polygon for each sensing area and we address the level of granularity to test the validity of sensing reporting.

### 6.3.1 Modeling a Non-Uniform Sensing Area

The deployment of sensing nodes usually creates a non-uniform sensing area. Measuring sensing coverage, in this case, is not a straightforward generalization as in its uniform counterpart. We assume that the border of each area that has a sensing node is known. This means we know the points on this border. Our goal is to detect whether or not the location of the sensing nodes affected by anchor misplacement are still within the residence sensing area. We model each sensing area as a simple polygon. We select the polygon to be strictly inscribed inside the sensing area (i.e., inner polygon). The question arises here is how do we compute such a polygon? Before we answer this question we should first illustrate the granularity related to sensing report. It is intuitive that the granularity will be finer if more points are selected from the border of the sensing area to be vertices in the computed polygon. Finer granularity of the sensing area provides more accurate decisions regarding the validity of sensing coverage.

We propose the following strategy to determine the level of granularity based in the number of vertices in the polygon. We differentiate between three levels of granularity according to the number of representative points of each edge in the sensing area. Thus, the granularity is:

- *Low*, if at most one vertex, in the generated polygon, represents each edge in the sensing area. No two consecutive edges without representation..
- *Medium*, if exactly one vertex, in the generated polygon, represents each edge in the sensing area.
- *High*, if more than one vertex, in the generated polygon, represents each edge in the sensing area.

Without loss of generality we assume that the sensing area is a convex set, hence its associated polygon is convex as well. Once the level of granularity is determined, we randomly choose the vertices that represent the edges of each sensing area. Then we connect these vertices clockwise to form an associated polygon.

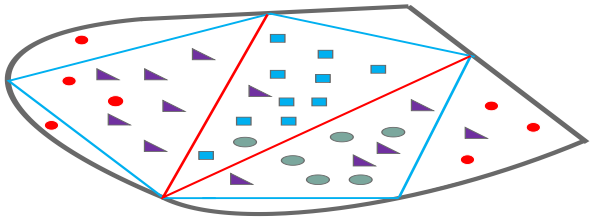


**Figure 6.2: A Possible inner polygon with a triangulation as a model of a non-uniform sensing area.**

### 6.3.2 The Impact of Error Components on Sensing Validity

In this Section, we study the error components that impact the localization accuracy and, consequently, affect the ratio of valid sensing reports. It is logical to address error components such as anchor misplacement and measurement error which inherently affects the estimated positions of sensing nodes. Unlike measurement error, which impacts the localization accuracy of all sensing nodes, anchor misplacement affects only the sensing nodes in the vicinity of misplaced anchor

nodes. Therefore, we need to differentiate between the impact of both error components. To achieve this, we need to know to what extent anchor misplacement contributed to the status of the sensing report of each affected sensing node. We first provide an overview of measurement error which depends on the physical properties of radio signal and channel quality. The general assumption is that the measurement error follows two-mode Normal distribution with a probability density function:

$$f_e(e) = \theta\, N(0, \sigma_{LoS}) + (1-\theta) N(\mu_{NLoS}, \sigma_{NLoS}) \quad \textbf{(6.1)}$$

Where $\sigma_{LoS}$ and $\sigma_{NLoS}$ are the standard deviation in line-of-sight (LoS) and none-line-of-sight (NLoS) scenarios. $\mu_{NLoS}$ denotes the mean in NLoS scenario. The random variable $e$ follows the LoS with probability $\theta$ and NLoS with probability $1-\theta$. We assume there is a measurement error model which is not changed through time and can be established prior to network deployment. In this research, we consider two error-force vectors (EFVs), namely measurement and misplacement vectors. Let $EFV_{meas}(s_i)$ , $EFV_{misp}(s_i)$ and $EFV_{result}(s_i)$ denote the error-force of mesurement, misplacement, and resultant vectors, respectively. $EFV_{result}(s_i)$ is the vector sum of $EFV_{meas}(s_i)$ and $EFV_{misp}(s_i)$ exerted during the localization of $s_i$ as shown in Figure 6.3. Disregarding other less important error components, we have the following formula:

$$\left|EFV_{meas}(s_i) + EFV_{misp}(s_i)\right| = \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad \textbf{(6.2)}$$

where |.| denotes the magnitude of a vector, $(x, y)$ and $(x_i, y_i)$ are the actual and estimated location of sensing node $s_i$, respectively. Furthermore, let $C_{meas}(s_i)$ , $C_{misp}(s_i)$ denote the magnitudes of contributed components of measurement and misplacment error, respectively, on the resultant error vector of sensing node $s_i$. Assume that each sensing node stores the values of $C_{meas}(s_i)$ and $C_{misp}(s_i)$. Note that if the two force vectors are in different directions (i.e., the angle between them is greater than $\frac{\pi}{2}$), then the smaller out of $C_{meas}(s_i)$ and $C_{misp}(s_i)$ should have a negative sign.

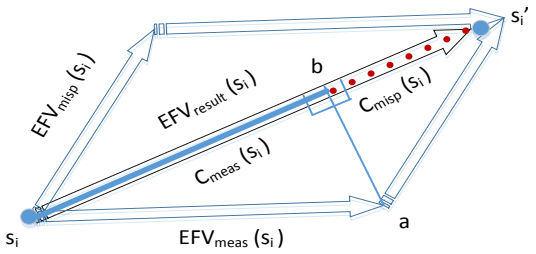



**Figure 6.3: Contributed errors of measurement and misplacement components in total resultant error.**

We next conduct an experiment to show the impact of $C_{meas}(s_i)$ and $C_{misp}(s_i)$ on Root Mean Square Distance (RMSD). Anchor misplacement could be random or ordered. Likewise, anchor displacement value could be random or fixed. Therefore, there are four different combinations of anchor misplacement and displacement values: 1) ordered anchor misplacement with fixed displacement value (OM-FD), 2) random anchor misplacement with fixed displacement value (RM-FD), 3) ordered anchor misplacement with random displacement (OM- RD), and 4) random anchor misplacement with random displacement value (RM-RD). In Figure 6.4, it is interesting to see that the deployment setting of RM-RD provides the least value of average RMSD among all other deployment settings. This interesting result supports the results in [98] where the authors found that random measurement contributes to high accuracy. Our results provide an extra finding that even with no measurement error, RM-RD provides higher localization accuracy. It is intuitive to see that RMSD values gets smaller as Signal to Noise Ratio (SNR) values gets bigger because the impact of measurement error gets smaller as well. Furthermore, the figure show that the impact of $C_{misp}(s_i)$ on accuracy becomes less effective as measurement error gets higher. We can conclude that RMSD is not so sensitive to anchor misplacement in high measurement-error environment because $C_{misp}(s_i)$ gets smaller compared to $C_{meas}(s_i)$ as $EFV_{meas}(s_i)$ cancels the effect of $EFV_{misp}(s_i)$.

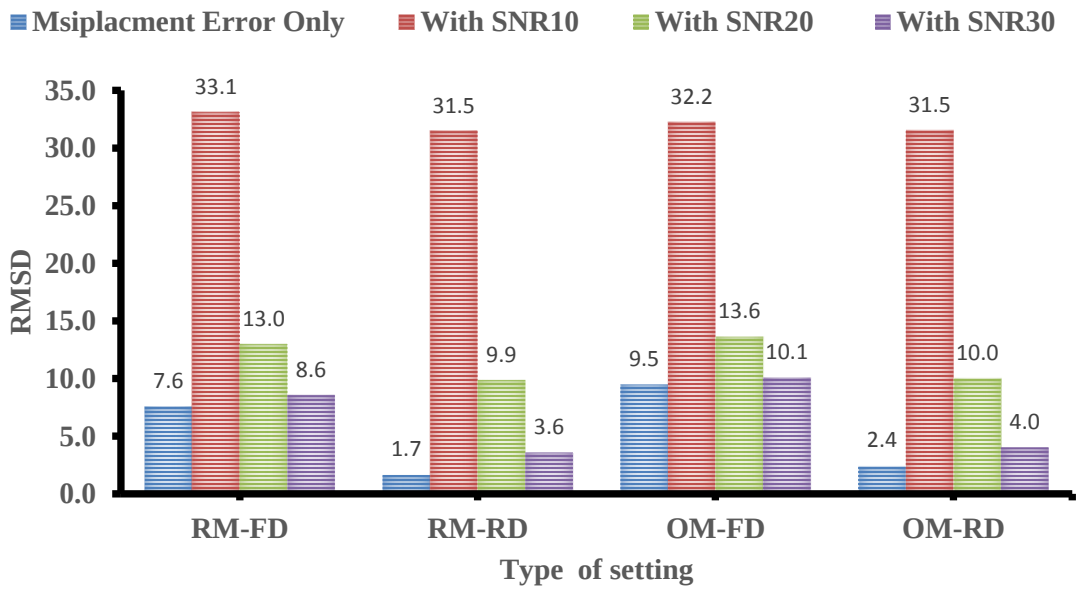



**Figure 6.4: The impact of the four different settings on RMSD (fixed displacement is set to 10m, random displacement follows N(0,10), RMSD is averaged over 14 misplaced anchor nodes).**

Next, we apply triangulation on the representative inner polygon or representative polygon for short.

### 6.3.3 Intra-Triangle Boundary Testing

By using the computational geometry tool of triangulation, we triangulate the representative polygon. This can be done by adding diagonal from one vertex to all other vertices in the case of convex polygon. If the polygon is a non-convex, the polygon should be first partitioned into convex pieces and then triangulate them. Another easier and efficient option is to decompose the non-convex polygon into so-called monotone pieces. A polygon $P$ is called monotone with respect to a line $\ell$, if every line perpendicular to $\ell$ intersects $P$ at most twice. Partition a simple polygon into $\ell$-

monotone polygons takes $O(n \log n)$. However, the $\ell$-monotone polygons can be triangulated in linear time [70]. Suppose there is a set $\mathcal{A} = \{\mathrm{A}_1, \mathrm{A}_2, \ldots., \mathrm{A}_k\}$ of sensing areas. The representative polygon of each of these sensing areas will be divided into triangles as shown in Figure 6.2. Then we check whether or not the position of a sensing node resides within any of these triangles. This is referred to as intra-triangle testing. As a result, there are two possible residence places of an estimated position. The first possible place (R1), the estimated position is inside one of the polygon's triangles which means TP sensing node. The second place (R2), the estimated position is outside all the polygon's triangles which means TN sensing node. Note that, when an estimated position of a sensing node $s_i$ resides in R2, this means that the magnitude of $EFV_{result}(s_i)$ becomes large enough and, hence, localize $s_i$ outside of its residence sensing area. Consequently, the accuracy of the sensing validity becomes lower. On the other hand, if the sensing nodes reside in R1, this means that the magnitude of $EFV_{result}(s_i)$ has no tangible impact on sensing quality. The following definition formalizes our discussion about intra-triangle testing.

**Definition 6.2:** Let area $A_1 \in \mathcal{A}$ be a sensing area and $P_{A_1}$ be its representative polygon. Furthermore, let $S_{A_1} = \{\mathrm{s}_1, \mathrm{s}_2, \ldots., \mathrm{s}_k\}$ be a set of sensing nodes deployed in area $A_1$, where *k* is the number of sensing nodes in $A_1$. Moreover, let $T_{A_1} = \{\mathrm{T}_1, \mathrm{T}_2, \ldots., \mathrm{T}_l\}$ be the set of triangles of $P_{A_1}$. We denote intra-triangle testing to the test that evaluates whether or not an estimated location of sensing node $s_i \in S_{A_1}$ resides inside any triangle of $T_{A_1}$.

To check the inclusion of a point inside a triangle, we follow the cross product method. Figure 6.5 shows a triangle ABC and a point $s_i'$ inside it. Let $\overrightarrow{AC}$ denote the vector that starts at point *A* and is directed towards point C. The idea behind cross product method is that the point is inside ABC only if $s_i'$ above vector BC, left to vector AC, and right to vector AB. If any one of these conditions fails, the point is outside the triangle. The direction of cross product of AC and A$s_i'$ should be in the same direction of the cross product of AC and AB as in Figure 6.5. The remaining combinations of vectors can be tested in similar way (see Algorithm 6.1).

**Algorithm 6.1:** Point in a Triangle (PIaT)

**Input:** $s_i'$, $T_i$; // point $s_i'$ and a triangle $T_i$
**Output:** Boolean value; //return true of a point $s_i'$ resides in a triangle $T_i$. Otherwise, return false.

```
1   {A,B, C} = get_Vertices(T_i);// return T_i's vertices
2   if( has_similar_dir((AC × As_i'), (AC × AB)) &&
        has_similar_dir((BC × Bs_i'), (BC × BA)) &&
        has_similar_dir((AB × As_i'), (AB × AC))) then
3       return true;
4   end if
5   else return false;
```

The details of the functions in PIaT are as follows: Function *get_Vertices*($T_i$) takes a triangle as an input and return a set that contains the vertices of $T_i$. Function $has\_similar\_dir(\vec{v}, \vec{u})$ tests whether or not vectors $\vec{v}$ and $\vec{u}$ have the same direction. $has\_similar\_dir$ returns true if the dot (i.e., inner) product of $\vec{v}$ and $\vec{u}$ is nonnegative. Otherwise, it returns false.

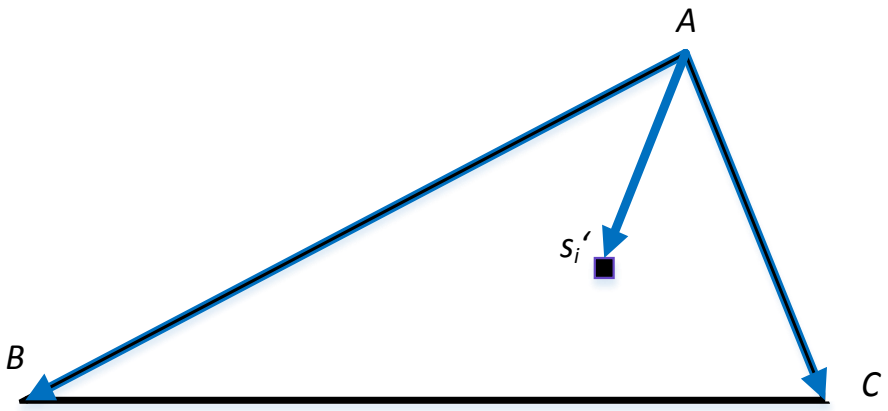



**Figure 6.5: Test a point in a triangle by cross-product method.**

Next, we design an algorithm that considers the method above and test the validity of the sensing reports.

## 6.4 Testing the Validity of the Sensing Report

The following algorithm tests the validity of sensing reports of the affected sensing node.

**Algorithm 6.2:** Testing Validity of the Sensing Report(TVSR)

**Input:** $T_{A_1}, S_{A_1}$ /*Set of triangles of a representative polygon of area $A_1$ and a set of sensing nodes in $A_1$, respectively*/

**Output:** Classified sensing nodes as TP, or TN.

```
1   for each triangle T_i ∈ T_A1
2     for each sensing node s_i ∈ S_A1
3       if (PIaT(s_i', T_i)) then
          // s_i' is the estimated position of s_i
4           set_Intra-Triangle_sensing(s_i', T_i) = 1;
5       end if
6     end for
7   end for
8   for each sensing node s_i ∈ S_A1
9     if (Intra-Triangle_sensing(s_i') == 1) then
10          setValidSensing = setValidSensing ∪ {s_i};
11    else
12          setInvalidSensing = setInvalidSensing ∪ {s_i};
13    end else
14  end for
```

TVSR algorithm takes two sets: triangulation set ($T_{A_1}$) of a representative polygon of area $A_1$ and a set $S_{A_1}$ of sensing nodes in $A_1$. The output is two classes of sensing nodes: TP or TN. In the first two loops, the algorithm tests the inclusion of each sensor's location point in every triangle of $T_{A_1}$. If such a triangle is found, the sensing node is added to TP class. The next loop marks as TN all the remaining sensing nodes that are not TP. The description of the functions is as follows. $PIaT(s_i', T_i)$ calls PIaT algorithm with two arguments, namely a point and a triangle. It returns true if $s_i'$ resides inside $T_i$. Otherwise, it returns false. $set\_Intra\text{-}Triangle\_sensing(s_i', T_i) = 1$ marks point $s_i'$ as TP. The rest of the sensing nodes are marked as TN.

## 6.5 Experimental Results

We use NS-3 to simulate different scenarios of the conducted experiments. The outputs of the simulation step will be inputs for a Visual Studio C++ program which includes our implementation of the proposed algorithm. We also utilize the implementation of a distributed algorithm in [77] to

construct a triangulation. The experiments show the effect of the following parameters on the percentage of valid and invalid sensing reports: the number of misplaced anchor nodes, and the measurement error component (i.e., $EFV_{meas}(s_i)$). We assume that $EFV_{meas}(s_i)$ follows a normal distribution with mean zero and variance ${\sigma_{i,j}}^2$, where $i$ and $j$ are the identifications of the sensing and anchor nodes, respectively. In order to better estimate the distance between the sensing and anchor nodes, we follow the formulation in [99] [100] which had adopted the following equation for the variance: ${\sigma_{i,j}}^2 = \frac{{d_{i,j}}^2}{SNR}$, where SNR is signal-to-noise ratio. The parameters of all experiments are set as follows, unless otherwise stated. The terrain is a square of 200 X 200 $m^2$. The terrain is a warehouse which is divided into six non-uniform sensing areas marked $A_1$-$A_6$, see Figure 6.6. However, for the sake of simplicity and to avoid repetition, we only focus on sensing area $A_1$. The number of sensing and anchor nodes per sensing area are 30 and 4, respectively. The sensing nodes are deployed randomly in each sensing area while the anchor nodes are placed on the corners of each sensing area. They are numbered 1-14. For full communication coverage, the transmission range is set to be 142m which is equivalent to half of the diameter of the terrain. As we illustrated in Section 6.3.2, there are two options for anchor misplacement ordered or random. Under ordered misplacement, the anchor nodes begin to be misplaced in order starting from anchor node number 1, then anchor node number 2, and so on until the required number of misplaced anchor nodes are reached. Anchor misplacement follows a Uniform random distribution. Similarly, the displacement value of misplaced anchor nodes can be either fixed or random. The fixed displacement value is set to 10m on both x- and y-coordinates. In the case of random displacement, the displacement value follows a normal distribution on both x- and y-coordinates with mean zero and variance of 10m. The results of all conducted experiments are calculated based on the average of 10 runs.

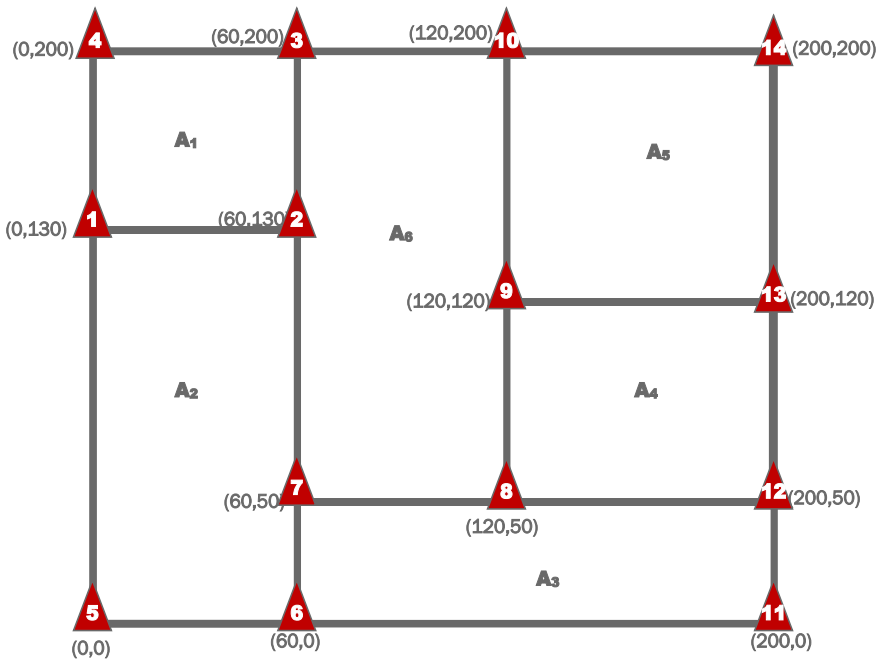



**Figure 6.6: Warehouse model with six non-uniform sensing areas with 14 numbered anchor nodes placed in the corners.**

We first study the impact of measurement error on sensing validity. The number of misplaced anchor nodes in this experiment is set to 5. The SNR values are 10, 20, and 30db. Furthermore, ordered anchor misplacement with fixed displacement value (OM-FD) is adopted in this scenario. The percentage of TP and TN of the sensing nodes will be calculated in two cases: with, and without the existence of measurement error (i.e., SNR-Free). The results are shown in Figure 6.7. The results show that as the measurement error becomes smaller, the percentage of TP increases while the percentage of TN decreases. On the other hand, TP-SNR-Free and TN-SNR-Free refer to the other case where no measurement error is applied. We note that the percentage of TP is at least as three times as the percentage of TN sensing nodes. Furthermore, Figure 6.7 shows that TP-SNR-Free and TN-SNR-Free tend to be convergence limits for TP, and TN, respectively. This is because, as the SNR value gets higher, the impact of $C_{meas}$ gets smaller which makes the resultant error component more driven by $C_{misp}$ .

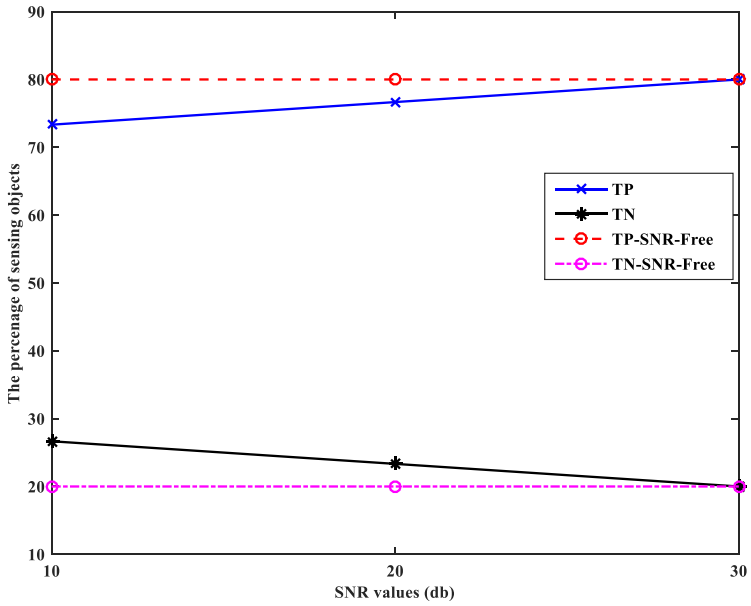
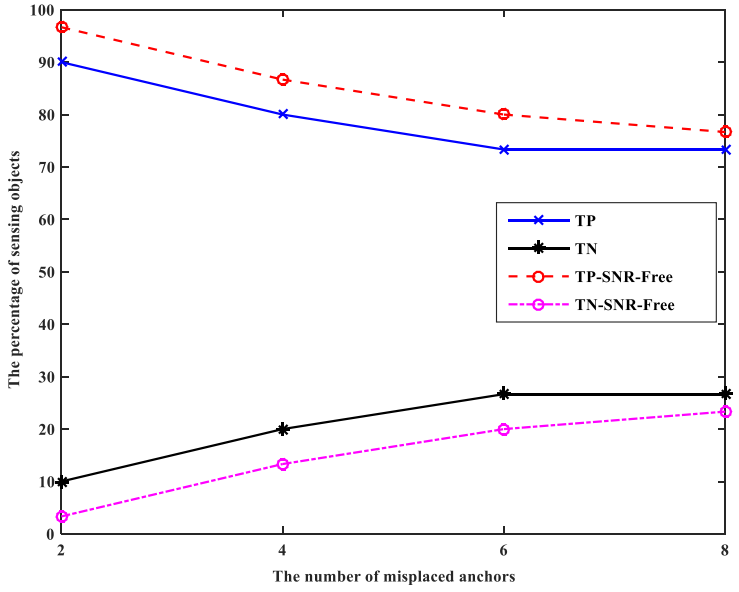
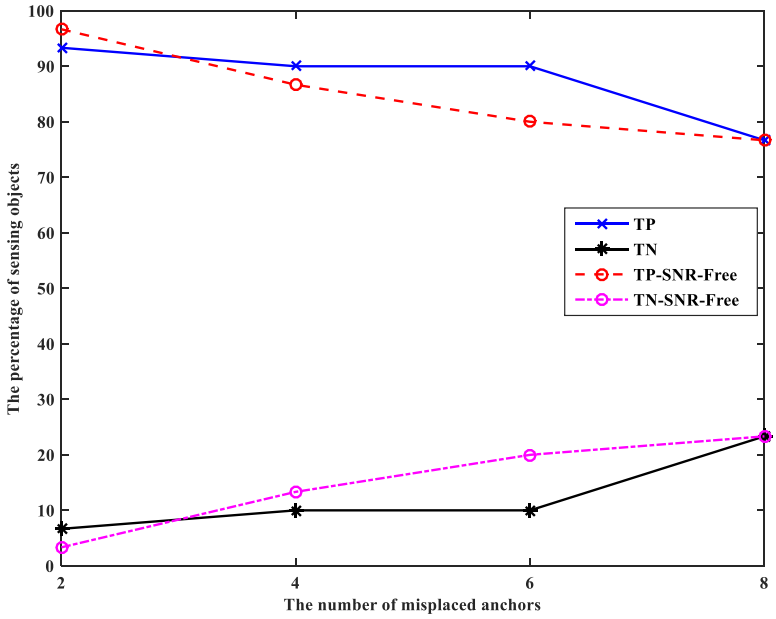



**Figure 6.7: The impact of measurement error on the sensing validity.**

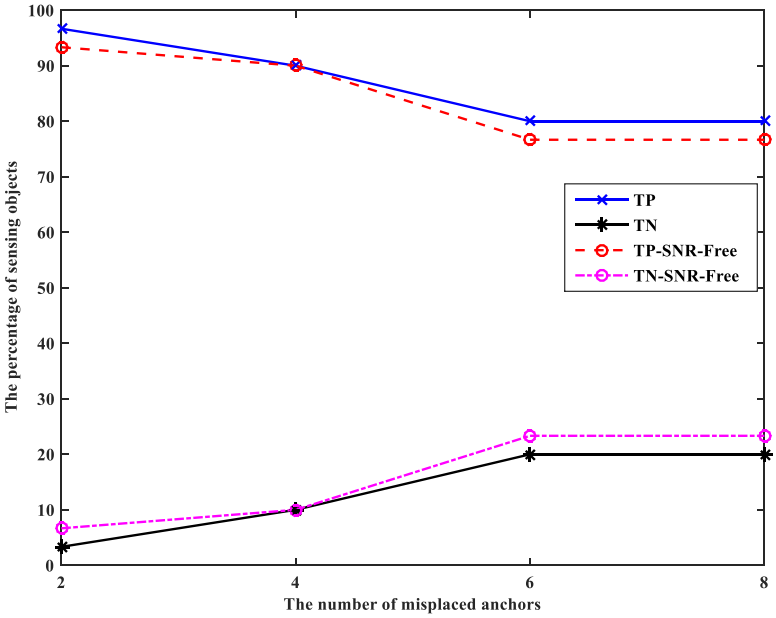
Next we study the impact of anchor misplacement on sensing validity under OM-FD setting. We conduct this experiment under various values of SNR, namely, 10, 20, and 30 db. The results are shown in Figure 6.8.




**(a) SNR = 10db**




**(b) SNR = 20db**




**(c) SNR = 30db**

**Figure 6.8: The impact of anchor misplacement on sensing validity under OM-FD with different values of SNR.**

Figure 6.8 shows that as the number of misplaced anchor nodes increases, the percentage of TP decreases, while the percentage of TN increases for all SNR values. Figure 6.8(a) has the lowest percentage of TP sensing nodes with average 70%, compared to 77% and 80% in Figure 6.8(b) and (c), respectively. It is intuitive to see that while the SNR value increases, TP converges to TP-SNR-Free, and TN converges to TN-SNR-Free. This is because $C_{meas}$ becomes smaller and eventually will have trivial values compared to $C_{misp}$.

The third experiment is similar to the previous one. However, it is conducted under RM-RD setting. We only include the result for SNR value of 10.

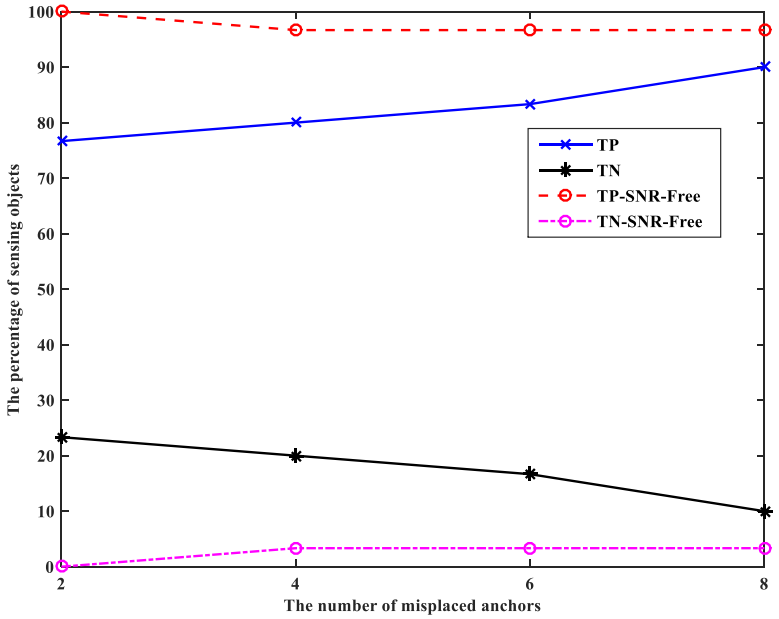



**Figure 6.9: The impact of anchor misplacement on sensing validity under SNR 10db.**

Figure 6.9 shows interesting results where the number of misplaced anchors has no negative impact of sensing validity. In contrast, as the number of misplaced anchor nodes increases, the percentage of TP sensing nodes increases while the percentage of TN sensing nodes decreases. This means that measurement error cancels out the impact of anchor misplacement. This result is consistent with our results described in Section 6.3.2 where randomness contributes to high accuracy and, hence, a high percentage of sensing validity.

Lastly, we compare the impact of the four combinations that we listed in Section 6.3.2, namely, OM-FD, RM-FD, OM-RD, and RM-RD. In this experiment, the TP, TN, TP-SNR-Free and TN-SNR-Free values are averaged over 14 misplaced anchor nodes.

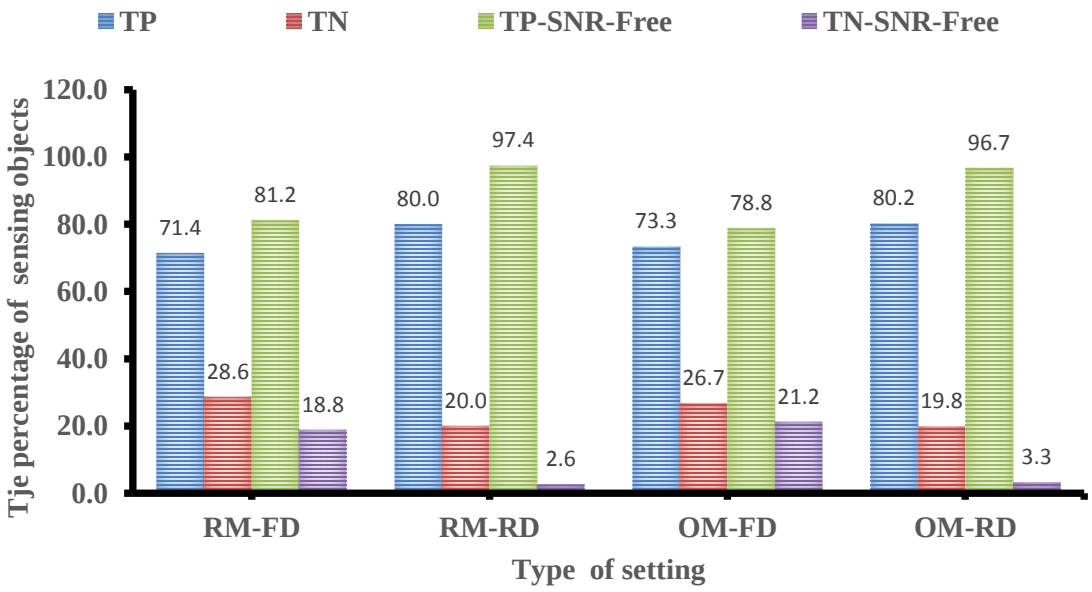



**Figure 6.10: The impact of different settings on sensing validity.**

The results show that the best percentage values of valid sensing are obtained when the anchor displacement values are selected randomly, see Figure 6.10. The differences between TP, TN and their SNR-Free counterparts are expected to get smaller when the values of SNR increases. This is because the measurement error gets smaller and, hence, converge to SNR-Free case.

## 6.6 Summary

In this chapter, we investigate the problem of sensing validity under the presence of anchor misplacement with and without the existence of measurement error. We present an algorithm that tests the sensing validity and classifies the sensing nodes as true positive (TP) or true negative (TN).

The results of our research show the importance of validating the sensing reports especially under the presence of error components such as anchor misplacement and measurement error. Results also show the need to evaluate the impact of anchor location uncertainty on the performance of the existing coverage protocols. Anchor misplacement poses serious degradation in the quality of sensing coverage. Consequently, false sensing reports are generated by affected sensing objects. While collective sensing nodes improves the percentage of sensing coverage, anchor misplacement increases the TN sensing nodes. The findings of our study also show that the randomness of anchor misplacement and displacement value mitigates the impact of anchor misplacement and gets higher valid sensing rate. Furthermore, our findings suggest that low SNR values undermines the impact of anchor misplacement. Under high SNR values, extending the sensing range of the affected sensing nodes reduces the impact of anchor misplacement and provide a great number of valid sensing report.

# Chapter 7

# Summary and Conclusions

Localization and sensing coverage are two major services in WSNs. WSN is one of several enabling technologies of IoT. Operating under the umbrella of IoT necessitates considering the characteristics of IoT such as heterogeneity, scalability, dynamicity, randomness, and multiple ownership.

Anchor misplacement affects the accuracy of localization. Furthermore, it creates perceived coverage which is a false coverage. This leads to a serious degradation to the quality of sensing coverage. False sensing reports posed by affected sensing objects may lead to fatalities in cases such as wildfire, chemical and gas leakage.

Next, the summary of this thesis is presented in Section 7.1. In Section 7.2, we conclude the thesis, while the future work directions is presented in Section 7.3.

## 7.1 Summary

In Chapter 3, we investigate the IoT sensing coverage problem where heterogeneous and randomly deployed sensing nodes are considered. Computational geometry has been utilized to establish a localized approach that enabled us to discover the problem in a distributed manner. Intra-triangle coverage approach is introduced to detect the coverage holes, and providing lower and upper bounds for coverage holes. The results benefit many large-scale coverage applications to detect the sensing coverage problem and either to tolerate the sensing holes or to heal them by adding more sensors to the affected area.

In Chapter 4, we investigate the problem of anchor misplacement in WSNs which has been generally overlooked by the WSN research community. Mitigating the impact of anchor misplacement should be considered in order to realize accurate and reliable localization service in

IoT context. A distributed algorithm to detect the misplaced anchor nodes has been proposed. The performance evaluation of our proposed algorithm outperforms the other competitive algorithm in [91] in terms of successful detection ratio, mistaken anchor ratio, and localization accuracy.

In Chapter 5, we extend our research in Chapter 3 to cover the sending coverage under the presence of anchor misplacement. Characteristics of IoT such as heterogeneous and non-deterministic deployment of IoT sensing nodes have been considered. We show new types of coverage holes which have been posed by anchor misplacement. We exploit a Delaunay Triangulation to zoom in on the problem. This enables a close detection of coverage holes in the vicinity of affected sensing nodes.

In Chapter 6, we address the problem of sensing validity with the possible existence of anchor misplacement and measurement error. We propose an algorithm that tests the sensing validity and classifies the sensing nodes as true positive (TP) or true negative (TN). The results of our research show the importance of validating the sensing reports especially under the presence of error components such as anchor misplacement and measurement error. Results also show the need to evaluate the impact of anchor location uncertainty on the performance of the existing coverage protocols.

## 7.2 Conclusion

In IoT context, collective networks run by multiple operators would enhance the services in terms of accuracy and reliability. For instance, collective IoT sensing nodes not only improve the percentage of sensing coverage, but also enhance the identification of the bounds of coverage holes among these networks. However, the research in WSNs has overlooked the problem of anchor uncertainty. The assumption that there is an absolute confidence of anchor's position is not always true. In fact, the probability of anchor misplacement increases in dynamic contexts such as IoT.

We found that a larger sensing range with fewer sensing nodes mitigates the impact of anchor misplacement on sensing coverage. This provides economic benefits in terms of fewer sensing

nodes being needed. It can also expand the lifespan of collective IoT networks by preserving energy while maintaining the average sensing range at a desired level.

Our findings show that the randomness of anchor misplacement and displacement value mitigates the impact of anchor misplacement and gets higher valid sensing rate. Furthermore, low SNR values undermines the impact of anchor misplacement. Under high SNR values, extending the sensing range of the affected sensing nodes reduces the impact of anchor misplacement and provides a greater number of valid sensing reports.

Thus, it is important to overcome the impact of anchor misplacement on major IoT services such as sensing coverage and localization. The error posed by anchor misplacement should be included in any future error model of IoT. Otherwise, the error model will be deficient.

## 7.3 Future Work

Communication coverage is of direct consequence of sensing coverage [101], [102]. Therefore, anchor misplacement has an impact on communication coverage as well. Future work includes to study the routing holes resulting from anchor misplacement and find out to what extent anchor misplacement impacts the lifespan (i.e., connectivity) of the network. It would be interesting to know what the minimum average node degree is such that the network remains connected after discarding the incorrect anchor nodes.

Another direction would be to design of a stochastic model for the anchor misplacement error. This requires an analytical study to formulate and determine the best fitting probability distribution for this type of error.

Future work may also include the relationship between sensor density and mitigating the effect of anchor misplacement on coverage hole. For instance, what would be the sensor density that would eliminate the coverage hole? Another possibility is looking into sufficient sensing coverage in overlapped sensing nodes. Given a set of deployed sensing nodes of multiple sensing

providers, what is the subset of these nodes that provides the best coverage and/or monetary cost savings?

# Bibliography


[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] M. Thoma, S. Meyer, K. Sperner, S. Meissner, and T. Braun, "On IoT-services: Survey, Classification and Enterprise Integration," in *2012 IEEE International Conference on Green Computing and Communications*, 2012, pp. 257–260.

[3] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 287–292.

[4] S. K. Sowe, T. Kimata, M. Dong, and K. Zettsu, "Managing Heterogeneous Sensor Data on a Big Data Platform: IoT Services for Data-Intensive Science," in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, 2014, pp. 295–300.

[5] T. Yashiro, S. Kobayashi, N. Koshizuka, and K. Sakamura, "An Internet of Things (IoT) architecture for embedded appliances," in *2013 IEEE Region 10 Humanitarian Technology Conference*, 2013, pp. 314–319.

[6] D. B. Hofmann-Wellenhof, D. H. Lichtenegger, and D. J. Collins, "Introduction to Global Positioning System," in *Global Positioning System*, Springer Vienna, 1994, pp. 1–11.

[7] H. Zhang, "Planet of the things," *Computer Fraud & Security*, vol. 2016, no. 3, pp. 16–17, Mar. 2016.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[9] D. Zhang, G. Li, K. Zheng, X. Ming, and Z. H. Pan, "An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 766–773, Feb. 2014.

[10] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, 2007.

[11] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, Apr. 2014.

[12] M. Ilyas and I. Mahgoub, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, 2004.

[13] U. Nazir, N. Shahid, M. A. Arshad, and S. H. Raza, “Classification of localization algorithms for wireless sensor network: A survey,” in *2012 International Conference on Open Source Systems and Technologies*, 2012, pp. 1–5.

[14] E. Fadel *et al.*, “A survey on wireless sensor networks for smart grid,” *Computer Communications*, vol. 71, pp. 22–33, Nov. 2015.

[15] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, May 2005.

[16] M. Hammoudeh and R. Newman, “Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance,” *Information Fusion*, vol. 22, pp. 3–15, Mar. 2015.

[17] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2007.

[18] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, “Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 551–591, Second 2013.

[19] Y. Pei, “A Survey on Localization Algorithms for Wireless Ad Hoc Networks,” in *Communications and Information Processing*, M. Zhao and J. Sha, Eds. Springer Berlin Heidelberg, 2012, pp. 512–519.

[20] Z. Farid, R. Nordin, and M. Ismail, “Recent Advances in Wireless Indoor Localization Techniques and System,” *Journal of Computer Networks and Communications*, vol. 2013, p. e185138, Sep. 2013.

[21] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons, 2010.

[22] M. R. Brust, M. İ. Akbaş, and D. Turgut, “Multi-hop localization system for environmental monitoring in wireless sensor and actor networks,” *Concurrency and Computation: Practice and Experience*, vol. 25, no. 5, pp. 701–717, Apr. 2013.

[23] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, “Range-free Localization and Its Impact on Large Scale Sensor Networks,” *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 4, pp. 877–906, Nov. 2005.

[24] D. Niculescu and B. Nath, “DV Based Positioning in Ad Hoc Networks,” *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, Jan. 2003.

[25] S. Kumar and D. K. Lobiyal, “An Advanced DV-Hop Localization Algorithm for Wireless Sensor Networks,” *Wireless Personal Communications*, vol. 71, no. 2, pp. 1365–1385, Jul. 2013.

[26] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2000, vol. 2, pp. 775–784 vol.2.

[27] 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, Nov. 2004.

[28] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[29] J. Kuriakose, S. Joshi, R. Vikram Raju, and A. Kilaru, “A Review on Localization in Wireless Sensor Networks,” in *Advances in Signal Processing and Intelligent Recognition Systems*, S. M. Thampi, A. Gelbukh, and J. Mukhopadhyay, Eds. Cham: Springer International Publishing, 2014, pp. 599–610.

[30] P. Wan, Z. Li, and B. Hao, “Time delay estimation of co-frequency signals in TDOA localization based on WSN,” in *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2016, pp. 1–5.

[31] F. Yaghoubi, A. A. Abbasfar, and B. Maham, “Energy-Efficient RSSI-Based Localization for Wireless Sensor Networks,” *IEEE Communications Letters*, vol. 18, no. 6, pp. 973–976, Jun. 2014.

[32] H. J. Shao, X. P. Zhang, and Z. Wang, “Efficient Closed-Form Algorithms for AOA Based Self-Localization of Sensor Nodes Using Auxiliary Variables,” *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2580–2594, May 2014.

[33] J. Wang, R. K. Ghosh, and S. K. Das, “A survey on sensor localization,” *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, Jan. 2010.

[34] G. Han, J. Jiang, L. Shu, Y. Xu, and F. Wang, “Localization Algorithms of Underwater Wireless Sensor Networks: A Survey,” *Sensors*, vol. 12, no. 2, pp. 2026–2061, Feb. 2012.

[35] I. Guvenc and C. C. Chong, “A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques,” *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 107–124, rd 2009.

[36] X. Qu and L. Xie, “A comparison study on TDOA based localization algorithms for sensor networks,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 4490–4495.

[37] K. Whitehouse, C. Karlof, and D. Culler, “A Practical Evaluation of Radio Signal Strength for Ranging-based Localization,” *SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, Jan. 2007.

[38] R. P and M. L. Sichitiu, “Angle of Arrival Localization for Wireless Sensor Networks,” in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2006, vol. 1, pp. 374–382.

[39] G. Deak, K. Curran, and J. Condell, “A survey of active and passive indoor localisation systems,” *Computer Communication*, vol. 35, no. 16, pp. 1939–1954, Sep. 2012.

[40] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2001, pp. 166–179.

[41] G. S. Kuruoglu, M. Erol, and S. Oktug, “Localization in Wireless Sensor Networks with Range Measurement Errors,” in *2009 Fifth Advanced International Conference on Telecommunications*, 2009, pp. 261–266.

[42] H.-P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14–15, pp. 1663–1676, Oct. 2011.

[43] T.-Y. Wu, G.-H. Liaw, S.-W. Huang, W.-T. Lee, and C.-C. Wu, "A GA-based mobile RFID localization scheme for internet of things," *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 245–258, Mar. 2012.

[44] F. Seco, A. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization," *Intelligent Signal Processing*, pp. 9–14, 2009.

[45] I. A. Getting, "Perspective/navigation-The Global Positioning System," *IEEE Spectrum*, vol. 30, no. 12, pp. 36–38, Dec. 1993.

[46] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[47] I. Amundson and X. D. Koutsoukos, "A Survey on Localization for Mobile Wireless Sensor Networks," in *Mobile Entity Localization and Tracking in GPS-less Environnments*, Springer, Berlin, Heidelberg, 2009, pp. 235–254.

[48] Z. Chen, F. Xia, T. Huang, F. Bu, and H. Wang, "A localization method for the Internet of Things," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 657–674, Mar. 2013.

[49] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: self-configuring localization systems," in *International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK*, 2001.

[50] G. Tan, H. Jiang, S. Zhang, Z. Yin, and A.-M. Kermarrec, "Connectivity-based and Anchor-free Localization in Large-scale 2D/3D Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, p. 6:1–6:21, Dec. 2013.

[51] X. Shen, Z. Wang, P. Jiang, R. Lin, and Y. Sun, "Connectivity and RSSI Based Localization Scheme for Wireless Sensor Networks," in *Advances in Intelligent Computing*, 2005, pp. 578–587.

[52] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-Free Localization Using Expected Hop Progress in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1540–1552, Oct. 2009.

[53] D. Milioris, G. Tzagkarakis, A. Papakonstantinou, M. Papadopouli, and P. Tsakalides, "Low-dimensional signal-strength fingerprint-based positioning in wireless LANs," *Ad Hoc Networks*, vol. 12, pp. 100–114, Jan. 2014.

[54] Y. Jin, W. S. Soh, and W. C. Wong, "Indoor localization with channel impulse response based fingerprint and nonparametric regression," *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 1120–1127, Mar. 2010.

[55] T. Bian, R. Venkatesan, and C. Li, "An Improved Localization Method Using Error Probability Distribution for Underwater Sensor Networks," in *2010 IEEE International Conference on Communications*, 2010, pp. 1–6.

[56] J. J. Robles, M. Deicke, and R. Lehnert, "3D fingerprint-based localization for wireless sensor networks," in *Navigation and Communication 2010 7th Workshop on Positioning*, 2010, pp. 77–85.

[57] J. Liang, M. Liu, and X. Kui, "A survey of coverage problems in wireless sensor networks," *Sensors & Transducers 1726-5479*, vol. 163, no. 1, 2014.

[58] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619–632, Mar. 2012.

[59] J. Chen, J. Li, S. He, Y. Sun, and H. H. Chen, "Energy-Efficient Coverage Based on Probabilistic Sensing Model in Wireless Sensor Networks," *IEEE Communications Letters*, vol. 14, no. 9, pp. 833–835, Sep. 2010.

[60] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[61] K. Xu, G. Takahara, and H. Hassanein, "On the Robustness of Grid-based Deployment in Wireless Sensor Networks," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, New York, NY, USA, 2006, pp. 1183–1188.

[62] J. O'rourke, *Art gallery theorems and algorithms*, vol. 57. Oxford University Press Oxford, 1987.

[63] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, Jun. 2008.

[64] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey," in *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, 2011, pp. 1–6.

[65] S. M. A. Oteafy and H. S. Hassanein, "Resource Re-use in Wireless Sensor Networks: Realizing a Synergetic Internet of Things," *Journal of Communications*, vol. 7, no. 7, Jul. 2012.

[66] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IOT Gateway: BridgingWireless Sensor Networks into Internet of Things," in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010, pp. 347–352.

[67] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, Mar. 2011.

[68] W. Li, J. Bao, and W. Shen, "Collaborative wireless sensor networks: A survey," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011, pp. 2614–2619.

[69] D. M. de Berg, D. M. van Kreveld, P. D. M. Overmars, and D. O. C. Schwarzkopf, "Voronoi Diagrams," in *Computational Geometry*, Springer Berlin Heidelberg, 2000, pp. 147–171.

[70] D. M. de Berg, D. M. van Kreveld, P. D. M. Overmars, and D. O. C. Schwarzkopf, "Delaunay Triangulations," in *Computational Geometry*, Springer Berlin Heidelberg, 2000, pp. 191–218.

[71] D. Tian and N. D. Georganas, "A Coverage-preserving Node Scheduling Scheme for Large Wireless Sensor Networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, New York, NY, USA, 2002, pp. 32–41.

[72] N. Ahmed, S. S. Kanhere, and S. Jha, "The Holes Problem in Wireless Sensor Networks: A Survey," *SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, Apr. 2005.

[73] H. Chizari, M. Hosseini, T. Poston, S. A. Razak, and A. H. Abdullah, “Delaunay Triangulation as a New Coverage Measurement Method in Wireless Sensor Network,” *Sensors*, vol. 11, no. 3, pp. 3163–3176, Mar. 2011.

[74] G. Takahara, K. Xu, and H. Hassanein, “Efficient Coverage Planning for Grid-Based Wireless Sensor Networks,” in *IEEE International Conference on Communications, 2007. ICC ’07*, 2007, pp. 3522–3526.

[75] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang, “Localized Delaunay triangulation with application in ad hoc wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 10, pp. 1035–1047, Oct. 2003.

[76] P. Bourke, “Calculating the area and centroid of a polygon,” 1988. [Online]. Available: http://local.Wasp.Uwa.Edu.Au/-pbourke/geometry/polyarea.

[77] “Delaunay Triangulation Library for C++,” 2014. [Online]. Available: http://www.geom.at/fade2d/html/. [Accessed: 03-Oct-2014].

[78] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, “Localization algorithms of Wireless Sensor Networks: a survey,” *Telecommunication Systems*, vol. 52, no. 4, pp. 2419–2436, Apr. 2013.

[79] A. Vempaty, O. Ozdemir, K. Agrawal, H. Chen, and P. K. Varshney, “Localization in Wireless Sensor Networks: Byzantines and Mitigation Techniques,” *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1495–1508, Mar. 2013.

[80] J. Zhao *et al.*, “Localization of Wireless Sensor Networks in the Wild: Pursuit of Ranging Quality,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 311–323, Feb. 2013.

[81] N. A. Alrajeh, M. Bashir, and B. Shams, “Localization Techniques in Wireless Sensor Networks,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 6, p. 304628, Jun. 2013.

[82] D. Macagnano, G. Destino, and G. Abreu, “Indoor positioning: A key enabling technology for IoT applications,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 117–118.

[83] S. K. Pandey and M. A. Zaveri, “Localization for Collaborative Processing in the Internet of Things Framework,” in *Proceedings of the Second International Conference on IoT in Urban Space*, New York, NY, USA, 2016, pp. 108–110.

[84] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. B. Srivastava, “On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks,” in *Information Processing in Sensor Networks*, F. Zhao and L. Guibas, Eds. Springer Berlin Heidelberg, 2003, pp. 317–332.

[85] J. F. Sanford, M. Potkonjak, and S. Slijepcevic, *Localization in wireless networks: Foundations and applications*. Springer Science & Business Media, 2012.

[86] Z. Chaczko, R. Klempous, J. Nikodem, and M. Nikodem, “Methods of Sensors Localization in Wireless Sensor Networks,” in *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS’07)*, 2007, pp. 145–152.

[87] P. Kumar, L. Reddy, and S. Varma, “Distance measurement and error estimation scheme for RSSI based localization in Wireless Sensor Networks,” in *2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN)*, 2009, pp. 1–4.

[88] B. Alavi and K. Pahlavan, “Modeling of the TOA-based distance measurement error using UWB indoor radio measurements,” *IEEE Communications Letters*, vol. 10, no. 4, pp. 275–277, Apr. 2006.

[89] H. Shen, Z. Ding, S. Dasgupta, and C. Zhao, “Multiple Source Localization in Wireless Sensor Networks Based on Time of Arrival Measurement,” *IEEE Transactions on Signal Processing*, vol. 62, no. 8, pp. 1938–1949, Apr. 2014.

[90] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, “An analysis of error inducing parameters in multihop sensor node localization,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, pp. 567–577, Nov. 2005.

[91] R. Fan, H. Jiang, S. Wu, and N. Zhang, “Ranging error-tolerable localization in wireless sensor networks with inaccurately positioned anchor nodes,” *Wireless Communications and Mobile Computing*, vol. 9, no. 5, pp. 705–717, May 2009.

[92] K. Fukunaga and P. M. Narendra, “A Branch and Bound Algorithm for Computing k-Nearest Neighbors,” *IEEE Transactions on Computers*, vol. C-24, no. 7, pp. 750–753, Jul. 1975.

[93] G. Takahara, K. Xu, and H. Hassanein, "How Resilient is Grid-based WSN Coverage to Deployment Errors?," in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 2872–2877.

[94] H. Kenniche and V. Ravelomananana, "Random Geometric Graphs as model of Wireless Sensor Networks," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 2010, vol. 4, pp. 103–107.

[95] Y. Al Mtawa, "Histories of Iterated Path Graphs," *Journal of Combinatorics, Information & System Sciences (JCISS)*, vol. 32, pp. 175–188, 2007.

[96] Y. Al-Mtawa and M. Jazzer, "On Independence Problem of P2-Graph," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 9, no. 1, p. 205, 2009.

[97] N. Lasla, M. Younis, A. Ouadjaout, and N. Badache, "On optimal anchor placement for efficient area-based localization in wireless networks," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 3257–3262.

[98] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[99] W. M. Ibrahim, H. S. Hassanein, and A. E. M. Taha, "Does multi-hop communication enhance localization accuracy?," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 121–126.

[100] S. Qin, Q. Wan, and Z. X. Chen, "Fast subspace approach for mobile positioning with time-of-arrival measurements," *IET Communications*, vol. 5, no. 14, pp. 2035–2039, Sep. 2011.

[101] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2003, pp. 28–39.

[102] D. Tian and N. D. Georganas, "Connectivity maintenance and coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 6, pp. 744–761, Nov. 2005.