


Resilient Localization and Coverage in the Internet of Things

Yaser Al Mtawa ^{1,*} , Hossam S. Hassanein ² and Nidal Nasser ³

¹ Applied Computer Science, The University of Winnipeg, Winnipeg, MB R3B 2E9, Canada

² School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada; hossam@cs.queensu.ca

³ College of Engineering, Alfaisal University, Riyadh 11533, Saudi Arabia; nnasser@alfaisal.edu

* Correspondence: y.almtawa@uwinnipeg.ca

Abstract: The proliferation of the Internet of Things (IoT) has revolutionized traditional services, giving rise to emerging smart infrastructures by connecting the physical and digital worlds. Sensory data is essential in IoT-based systems for providing context-aware and location-based services. Hence, the accurate localization of IoT devices is paramount. Anchor misplacement can significantly affect location information and coverage services in IoT. We study the effect of anchor misplacement in typical IoT settings where sensors are randomly deployed, can be mobile, and may belong to multiple providers. We identify sensing coverage holes formed by anchor misplacement and analyze their presence and impact. To mitigate the impact of anchor misplacement on network reliability, we propose a framework to identify the affected sensor nodes and then identify and remove misplaced anchor nodes. The validity of our approach is verified, and its effectiveness is demonstrated by several experiments with different network topologies and parameters. Our results are promising and can be utilized in multiple coverage applications, such as smart agriculture systems and habitat monitoring, regardless of the sensors or deployment types. It also sheds light on best practices and methods for a reliable design of IoT-based systems.

Keywords: wireless sensor network; Internet of Things (IoT); IoT deployment; localization; sensing coverage; reliable services; intra-triangle coverage; Delaunay triangulation



Citation: Al Mtawa, Y.; Hassanein, H.S.; Nasser, N. Resilient Localization and Coverage in the Internet of Things. *Electronics* **2023**, *12*, 3172. <https://doi.org/10.3390/electronics12143172>

Academic Editors: Syed Muhammad Raza and Duc Tai Le

Received: 14 June 2023

Revised: 15 July 2023

Accepted: 17 July 2023

Published: 21 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the Internet of Things (IoT) applications expand into many different domains, it is important to consider the reliability of these services. The sensors' physical *location* and *sensing coverage* are two primary services of IoT applications, and these services must be delivered reliably for the IoT to be successful. In this research, we concentrate on improving IoT localization and sensing coverage by building upon our previous work found in [1].

The location of sensors and sensing coverage are directly influenced by the strategic deployment of nodes, ensuring efficient data collection and monitoring.

The methodologies for deploying sensor nodes depend on various factors, including the specific application, dimensions of the targeted area, degree of knowledge about the node density and positions, as well as the accessibility of the target region. Generally, two primary deployment approaches can be identified: deterministic and random. The deterministic deployment approach considers several predetermined factors, such as network topology, sensor positioning, inter-sensor distance, and density. This method of sensor deployment offers greater control over resource constraints. Energy consumption optimization is typically the primary metric in many deterministic deployment strategies [2–4]. A common example of this type is grid-based deployment, which includes hexagonal, square, and equilateral triangular patterns. The equilateral triangle configuration provides full coverage while minimizing the required sensor nodes [5]. The Art Gallery problem is a traditional problem of this type of deployment [6], where the objective is to position the minimum number of sensors to ensure complete monitoring of every point in the gallery.

In contrast to deterministic deployment, random deployment lacks information regarding the network topology and the specific positions of sensors. This mode of deployment is well-suited for extensive networks in harsh and remote environments, such as forests, mountainous regions, and hazardous locations, such as chemical leak sites and nuclear facilities. Nevertheless, random deployment cannot ensure complete sensing coverage [7]. Continuously maintaining all sensors in an active state results in rapid energy consumption, ultimately leading to network disconnection. As a consequence, some regions within the network may not receive sensing reports [8,9].

In both deployment methodologies, the communication between sensor nodes themselves and sensor and anchor nodes can be classified as either single-hop or multi-hop. Due to the increased transmission power required for longer distances, sensors with limited power reserves utilize shorter transmission ranges to conserve energy. As a result, the collective sensing data is transmitted from the originating node to the destination through multiple intermediate nodes or relays. Conversely, sensor nodes equipped with a continuous power supply maintain the capacity for single-hop communication with a sink node, enabling direct peer-to-peer exchanges.

1.1. Localization in IoT Networks

The Internet of Things (IoT) establishes a large-scale network composed of diverse sensor-enabled nodes. Wireless Sensor Networks (WSNs) are one of the primary drivers of IoT advancements. Typically, a WSN contains thousands of low-cost, low-power sensor nodes. These networks have numerous applications, including military surveillance, wildlife tracking, and environmental monitoring. Determining the sensors' physical positions is crucial for maintaining trusted IoT-based services.

Localization is fundamental to data collection and routing planning, enabling efficient information delivery to the target station. For instance, autonomous vehicles would face significant safety risks without precise positioning, resulting in increased accidents. Similarly, incorrect data from sensors in wildfire monitoring systems would undermine the service quality. Sensor localization methods can be categorized as anchor-based or anchor-free approaches.

Anchor nodes have more capabilities than sensors and are aware of their actual positions. Consequently, they serve as reference points for determining the unknown locations of other sensors. Conversely, anchor-free localization leverages tools such as relative positioning, mapping, and embedded GPS clients. In this study, we adopt an anchor-based strategy to localize IoT sensors. Specifically, we intend to utilize time of arrival (ToA) or time difference of arrival (TDoA) alongside the path loss model for signal measurements received at the sensor.

1.2. Sensing Coverage in IoT Networks

In a sensing field, a point is considered covered if it lies within the sensing proximity of a sensor node, meaning that the Euclidean distance between this point and any sensor node is less than the designated sensing range. Consequently, the sensing field may have either full or partial coverage. To address partial coverage, cooperative sensor networks can be employed, allowing for the deployment of sensor nodes belonging to multiple entities. For instance, when heterogeneous temperature sensors owned by three disparate sensing providers are deployed within a specified region, as illustrated in Figure 1, they can be regarded as shared resources. Cooperation among these sensors can yield improved quality of service (QoS) while also enabling the incorporation of participatory sensors such as smartphones.

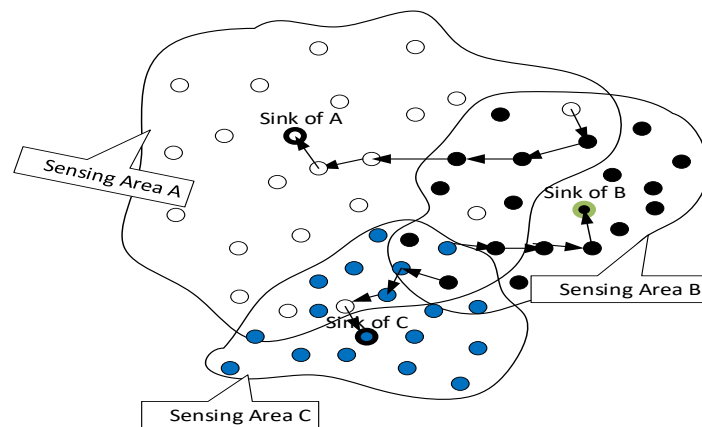


Figure 1. Multi-hop cooperative sensing networks.

1.3. The Relationship between Localization and Sensing Coverage

Identifying the presence of coverage gaps and outlining them constitutes a significant obstacle in the IoT context. Coverage gaps arise when sensing fields exhibit incomplete coverage. The sensing coverage requirements of IoT applications vary; for instance, certain applications necessitate single-sensing coverage where a minimum of one sensor is needed to observe any point within the target zone. Conversely, other applications demand extensive coverage, which entails k sensors to oversee each point within the area [10].

Additionally, there exist multiple methods for approximating a sensor's position; however, GPS is typically not employed due to energy constraints and the need for cost-effective sensors. Alternative solutions include anchor-based and anchor-free approaches, with the former being prevalent owing to its accuracy, while the latter offers rough localization precision. This study utilizes the anchor-based strategy.

The sensing coverage quality directly correlates with the credibility of aggregated sensing reports at the sink node; nevertheless, these reports are influenced by various error factors such as measurement discrepancies and anchor misplacement. These error components affect sensors' location estimates, consequently leading to inaccurate location imprints on aggregated reports and ultimately degrading the quality of sensing services.

1.4. The Impact of Anchor Misplacement

Anchor misplacement pertains to the issue wherein an anchor node occupies a particular position, yet assumes it is situated elsewhere [11]. For instance, an anchor node B erroneously assumes that it is located in a specific position while it is actually elsewhere. Numerous factors contribute to this issue, including cyberattacks, soil erosion, potential anchor displacement due to animal or human activities, or simply human error. Utilizing misplaced or inaccurate anchor nodes for localizing sensor nodes results in localization inaccuracies of sensor nodes [11,12]. Likewise, the quality of coverage is impacted as incorrectly estimated sensor locations yield invalid sensing reports. Consequently, the presence of anchor misplacement exacerbates sensing coverage. For example, in a smart agriculture system, anchor misplacement may prompt the inadvertent irrigation of perceived dry areas or the application of pesticides to healthy plants. Figure 2 illustrates the effects of a misplaced anchor on localizing 80 sensor nodes and 20 anchor nodes within a 200×200 target field. The actual location of this specific anchor is (13.1, 115), while its perceived or declared location is (6.1, 108). The figure demonstrates the impacted sensors and their imprecise locations.

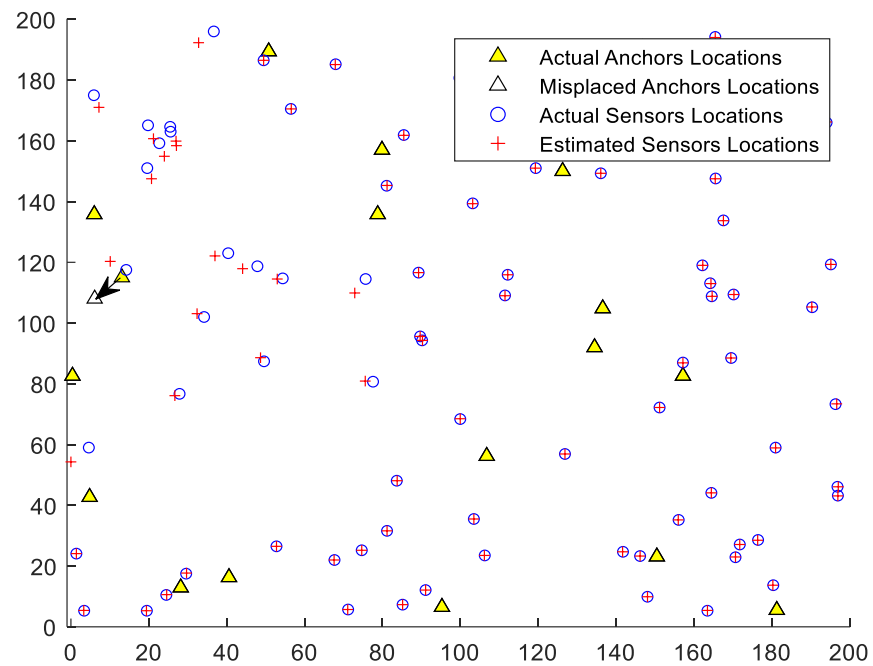


Figure 2. The impact of anchor misplacement on localizing sensor nodes.

1.5. Motivation

Cooperative sensing offers substantial benefits to WSNs by enhancing sensing reliability and expanding sensing coverage [12]. Nonetheless, current studies concerning sensing coverage typically assume an accurate anchor node position. In other words, literature research overlooks the anchor misplacement issue that affects both localization and sensing coverage. Figure 3 demonstrates the significance of mitigating the influence of anchor misplacement on localization precision and, consequently, on sensing coverage: an increased number of misplaced anchor nodes leads to a decline in IoT system service dependability.

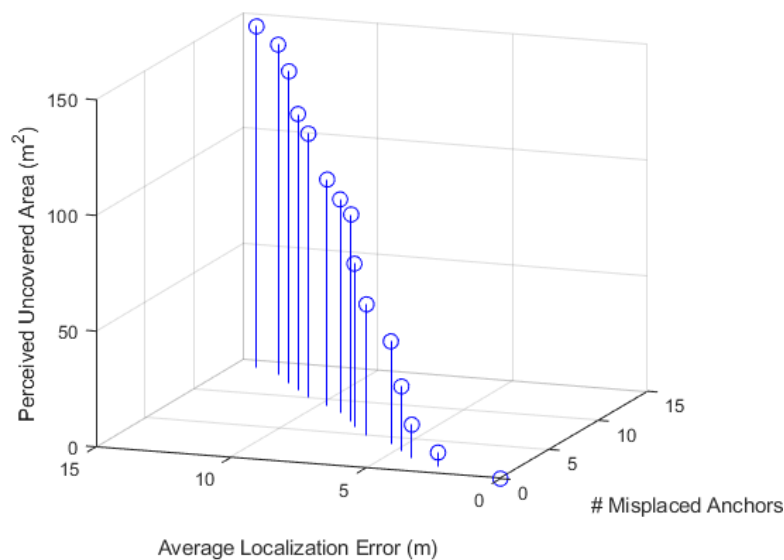


Figure 3. Anchor misplacement influence on network reliability.

Moreover, related research presumes that sensors are homogeneous and singularly affiliated with one sensing service provider. Relevant research focuses on deterministic sensor placement and deployment planning to achieve superior coverage and lengthen net-

work lifetime [13,14]. In contrast, this study considers sensors as heterogeneous, randomly deployed, and associated with multiple sensing service providers.

In the domain of IoT security, anchor misplacement complications prove more perilous. For example, a hacker may target anchor nodes to tamper with their locations, resulting in inaccurate data collection and subsequent unreliability. Such issues become life-threatening when software/hardware errors or malicious attacks aim at autonomous vehicular sensor nodes, like LiDAR (Light Detection and Ranging). LiDAR technology is intended to supply precise geospatial data swiftly for informed decision-making. Consequently, the malicious alteration of LiDAR data such as modifying positioning coordinates or introducing inaccuracies in distances to nearby vehicles or objects causes severe safety concerns.

These points raise the following primary research questions: Q1: How does anchor misplacement influence the balance between detection latency and service reliability? Detection of misplaced anchors is more effective when more non-misplaced nodes participate in validating localization results for a fixed number of misplaced anchors executing sensor localization. However, an excess of validators magnifies communication overhead, subsequently increasing latency in the detection phase completion. Q2: What impact does anchor misplacement have on sensory data aggregation coverage? Q3: Can new types of coverage holes be identified and their areas quantified?

In this study, our objective is to address these research questions by devising a distributed, fast, and dependable method to detect misplaced anchors in WSNs. We strive to minimize anchor validators to achieve high reliability with reduced latency. Furthermore, we analyze the effects of anchor misplacement on sensory coverage in cooperative WSNs.

1.6. Contributions

The contributions of our work can be summarized as follows:

1. We propose an efficient framework for detecting anchor misplacement, which necessitates minimizing anchor validators' involvement to ensure service reliability and minimal convergence latency;
2. We investigate and evaluate the presence of anchor misplacement issues in IoT systems and their effect on reliability. We show that using misplaced anchors for localizing sensors leads to imprecise positioning and compromised sensory data integrity;
3. Our study provides an in-depth analytical examination of how anchor misplacement impacts sensor localization and sensing coverage;
4. We suggest approaches for detecting coverage gaps resulting from anchor misplacement and estimating their size, along with determining upper and lower limits for these uncovered regions. By employing Delaunay Triangulation (DT), we divide the target sensing area into equilateral triangles. As intra-triangle coverage gaps are irregular, we aim to identify each gap locally and establish its boundaries. Our Intra-triangle Coverage (ITC) assessment method is decentralized, requiring only the vertices of each triangle to participate in calculations, thus ensuring scalability and energy efficiency;
5. We validate our proposed methodology's efficacy under various conditions while demonstrating superior localization accuracy and reduced perceived uncovered regions.

To the best of our knowledge, this is the only research investigating IoT sensing coverage under anchor misplacement by locally identifying new coverage holes and quantifying their area.

The structure of the rest of this paper is organized as follows. Section 2 provides an overview of fundamental definitions and models related to sensing coverage in WSNs, as well as various sensor deployment approaches. It also introduces the problem formulation and assumptions of our study. Preliminary findings utilizing the Voronoi Diagram (VD) and Delaunay Triangulation (DT) for efficient coverage are discussed in Section 3. A comprehensive investigation of ITC is conducted in Section 4. The effects of anchor misplacement on sensing coverage, along with a method for determining the lower and upper bounds of sensing coverage gaps, are presented in Section 5. A resilient framework

for detecting and mitigating the influence of misplaced anchors is described in Section 6. Experimental results that validate our proposed framework and demonstrate its effectiveness are detailed in Section 7. Finally, Section 8 provides a conclusion and suggests future research directions.

2. System Model and Problem Description

Ensuring proper coverage is essential for reliable and effective sensing services. To achieve this, it is important to measure the accuracy of sensing reports in reflecting the physical surroundings of the target sensing field. Without adequate coverage, such services may become unreliable or obsolete. A list of notations and acronyms can be found in Table 1.

Table 1. Notations and acronyms.

Notation	Description
$G(V, E)$	A network graph G .
S	Set of sensor nodes.
B	Set of anchor nodes ($B_b \cup B_m$).
V	Set of sensor and anchor nodes ($S \cup B$).
E	Set of links/edges.
n	$ S $, the number of sensor nodes.
m	$ B $, the number of anchor nodes.
B_b	Set of benign anchor nodes (i.e., not misplaced).
B_m	Set of misplaced anchor nodes
DT	Delaunay triangulation.
VD	Voronoi diagram.
$d_{a,b}$	Euclidean distance between node a and node/point b .
$d_{a,b}'$	The distance between node a and node b measured via received signal strength (RSS) method.
s_i	A sensor node in a sensing field S , $1 \leq i \leq n$
R_{s_i}	A sensing range of sensor node s_i .

2.1. Sensing Coverage Formulation

Sensing coverage holes are one of the main reasons that degrade the quality of sensing services. They exist when any sensing node does not cover some points in the sensing field. The following is a definition of sensing coverage.

Definition 1. Let S denote the target sensing field and $N = \{s_i : s_i \text{ is a sensing node}; 1 \leq i \leq n\}$, where n is the number of sensing nodes. (x_i, y_i) refers to the location of s_i in a plane which is unknown initially. Each s_i has estimated location (x_i', y_i') and a sensing range R_{s_i} . Let p be a point in S , then p is covered if there is at least one s_i such that p is within a distance of R_{s_i} from s_i . In other words, $\{\exists s_i | d_{p,s_i} \leq R_{s_i}, 1 \leq i \leq n\}$, where $d_{a,b}$ is the Euclidean distance between a and b .

2.2. Types of Sensing Coverage Holes under Anchor Misplacement

Figure 4 shows the full and partial coverage.

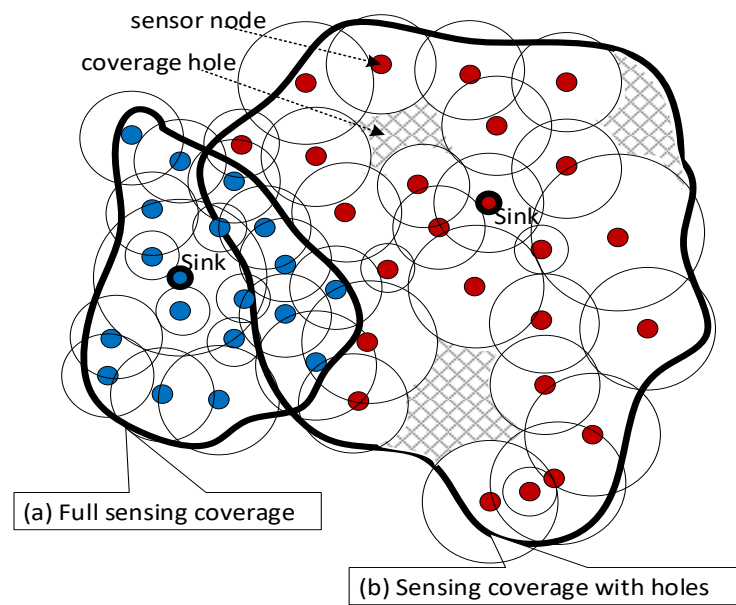


Figure 4. Full and partial sensing coverage using heterogeneous sensor nodes.

Anchor misplacement introduces new categories of coverage gaps, such as the perceived and hidden unreported actual coverage holes.

- (1) In the case of the perceived coverage gap, Figure 5a illustrates an ideal scenario in which sensor s_1 is precisely localized, unaffected by anchor misplacement. Conversely, Figure 5b demonstrates a situation where the localization of s_1 is dependent on a misplaced anchor, subsequently generating a false coverage hole. This hole is present in two triangular regions: $s_1's_2's_3'$ and $s_1's_3's_4'$.

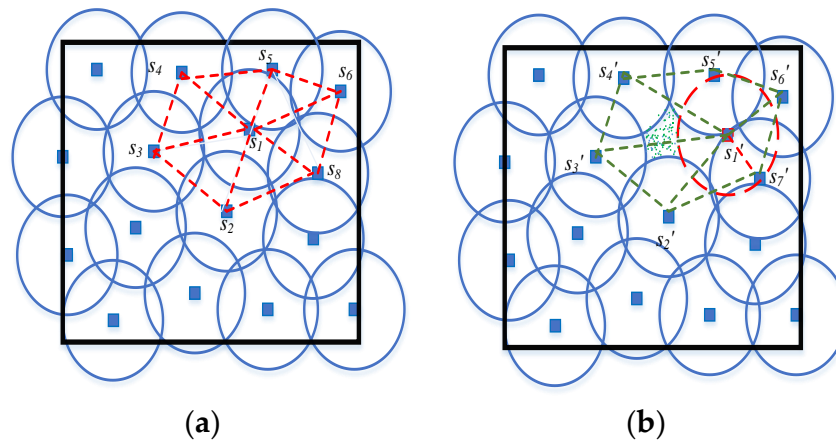


Figure 5. A triangulation in the vicinity of sensor node s_1 . (a) An ideal case is where there are no coverage holes. (b) A perceived coverage hole due to anchor misplacement.

- (2) The hidden, unreported actual coverage gap is illustrated in Figure 6. Figure 6a shows the presence of actual coverage holes within the triangular regions $s_1s_6s_7$ and $s_1s_7s_2$. Conversely, Figure 6b demonstrates that a misplaced anchor hides these holes due to the imprecise localization of sensor s_1 .

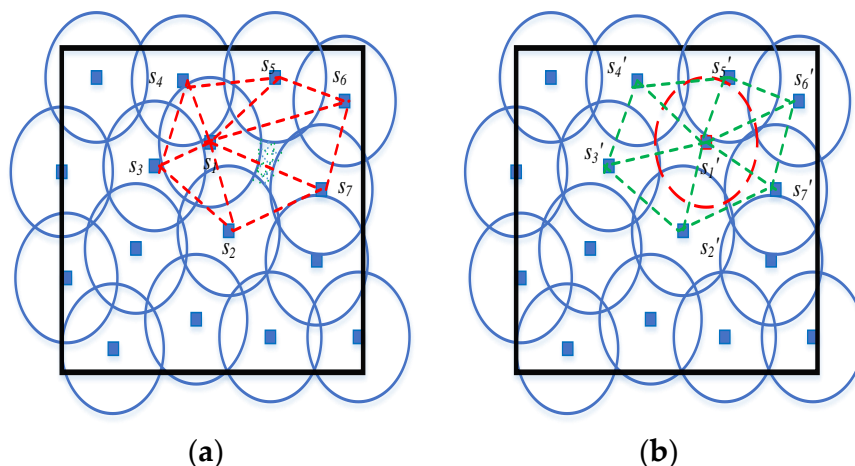


Figure 6. An actual unreported coverage hole can be identified by investigating the triangles in the vicinity of the affected sensor s_1 . (a) Original deployment with actual coverage hole. (b) An actual coverage hole is masked, and, unreported.

2.3. Network and Sensing Models

In the implementation of collaborative sensing, sensor nodes are attributed to distinct providers. A collection N comprises sensor nodes, while a collection M includes anchor nodes. These anchor nodes are strategically positioned at locations of interest (i.e., predetermined coordinates) to facilitate the localization of sensor positions. WSNs predominantly employ multi-hop communication for routing sensed information from source sensors towards a sink node. Within cooperative sensing, an originating node possesses the potential to direct its acquired data towards multiple sinks (i.e., data processing facilities).

The generation rate of sensed data packets by a sensor s_i will highly depend on its capabilities. Nevertheless, in WSNs, this rate is maintained at a lower threshold to conserve battery power and avoid network congestion.

Numerous sensing models can be developed in accordance with application prerequisites and the ambient environment. However, these models suggest that sensing efficacy fades as distance increases. This observation is captured in the subsequent formula [15].

$$SN(s_i, p) = \frac{\lambda}{(d_{s_i,p})^K} \tag{1}$$

where SN denotes the sensibility between the sensor node s_i and point p , and both λ and K are positive constants related to the sensor’s technology.

There are two types of sensing models: binary disc and probabilistic.

(1) Binary Disc Model

In the binary disc model, a sensor node is presumed to perform 360° monitoring. Consequently, a point within the sensing field is considered covered if it lies inside the circular sensing range of at least one sensor node. If this condition is not met, the point is deemed uncovered, as per the equation provided.

$$C(p) = \begin{cases} 1 & \text{if } d_{s_i,p} \leq R_{s_i} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Thus, the binary disc model abstracts the sensing coverage of s_i by a disc of radius R_{s_i} as shown in Figure 7.

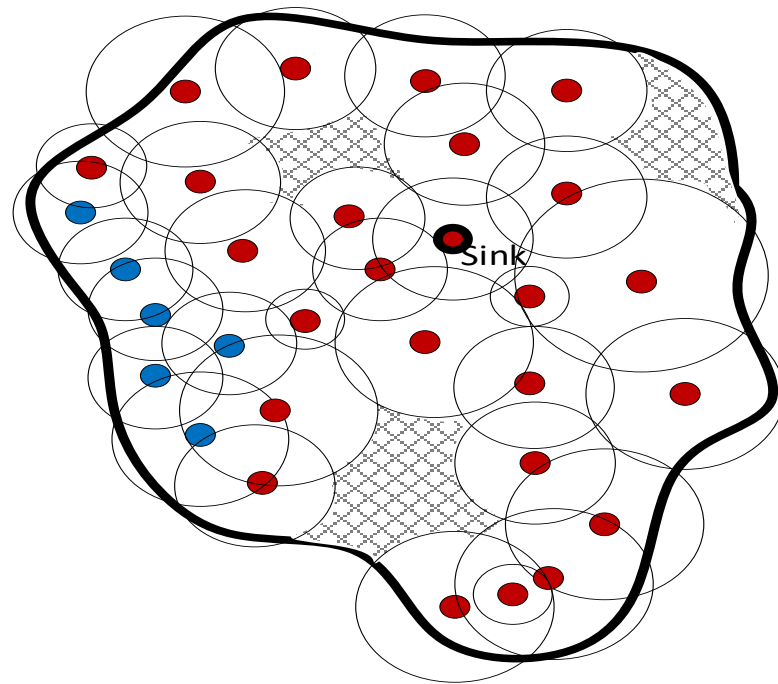


Figure 7. Disk model of overlapped IoT sensing providers with coverage holes.

(2) Probabilistic Sensing Model

The model is contingent upon the uncertainty associated with sensor detection. It employs the detection probability, considering a point p situated at a distance exceeding the uncertainty value, yet remaining within the specified range [16].

$$C(p) = \begin{cases} 1 & \text{if } d_{s_i,p} \leq R_{s_i} - \epsilon \\ 0 & \text{if } d_{s_i,p} \geq R_{s_i} + \epsilon \\ e^{-\alpha\beta^\gamma} & \text{if } R_{s_i} - \epsilon < d_{s_i,p} < R_{s_i} + \epsilon \end{cases} \quad (3)$$

In this context, ϵ represents the uncertainty value inherent in the sensor’s detection capabilities. The variable β is defined as $\beta = d_{s_i,p} - (R_{s_i} - \epsilon)$. Both α and γ serve as parameters assessing the likelihood of detecting object p when it is situated at a distance exceeding ϵ yet still lies within the range of R_{s_i} . Variations in these parameter values are indicative of distinct characteristics associated with different sensor types, thus resulting in diverse detection probability scales.

In this research, we adopt the binary disk sensing model for the sake of simplicity. Figure 7 shows the disk model representation of the overlapped region in Figure 1.

2.4. Channel Model

Both anchor and sensor nodes have wireless transceivers. The transmission power of a specific node, s_i , is represented by P_{s_i} . These nodes engage in communication through wireless channels, adhering to the established path loss model. When assessing the path loss between two nodes, s_i and s_j , with a Euclidean distance of d , it can be quantified in decibel scale by utilizing the following formula.

$$PL(d) = PL(d_0) + 10\eta \log\left(\frac{d}{d_0}\right) \quad (4)$$

where $PL(d_0)$ is the path loss at the reference distance d_0 , and η is the path loss exponent.

To better estimate the distance between the IoT nodes s_i and s_j , we follow the formulation adopted in [17] for the variance:

$$\sigma_{s_i, s_j}^2 = \frac{d_{s_i, s_j}^2}{SNR} \quad (5)$$

where SNR is the signal-to-noise ratio.

2.5. Communication Model

Two sensor nodes are adjacent/neighbors if they can communicate directly (i.e., have 1-hop wireless-link communication). We define $NH(s_i)$ to denote the set of sensor nodes in the neighborhood of s_i , that is, $NH(s_i) = \{s_j \mid d_{s_i, s_j} \leq P_{s_i} \text{ and } d_{s_i, s_j} \leq P_{s_j}, s_i \neq s_j, s_j \in N\}$. Upon receiving the transmitted data, each sensor node is capable of determining its proximity to the emitting sensor through the utilization of a signal strength indicator (SSI). The resulting graph generated by the collective sensor nodes and wireless connections is referred to as a routing graph. Multiple potential routing paths exist within the routing graph that can link the source sensor node to a sink. A routing path is characterized as a series of wireless connections which initiate at the source sensor and end at the sink, with no repetition of nodes.

Throughout the process of validating anchor misplacement, a node may function as either the transmitter or recipient of a message. The propagation of this message occurs through either 1-hop or multi-hop methods. As detailed in Section 1, it becomes evident that a node necessitates increased power capacities for single-hop communication with other nodes. Denote $P_{s_i, s}$ as the transmission power prerequisite for node s_i to engage in single-hop communication with any node located within the target field. Similarly, $P_{s_i, m}$ signifies the required transmission power for node s_i to establish multi-hop communication toward the base station via adjacent nodes. Clearly that $P_{s_i, s} \geq P_{s_i, m}$. Nodes typically employ single-hop communication techniques for broadcasting messages to all other participating nodes comprehensively.

2.6. Problem Description

Anchor misplacement adversely affects the dependability of IoT-based systems, resulting in special sensing coverage holes due to imprecise positioning of impacted sensor nodes. Consider B_j as a misplaced anchor node, with $M' \subseteq M$ representing the collection of misplaced anchor nodes. Analogous to the sensor node neighborhood discussed in Section 2.3, the neighborhood, $NH_B(B_j)$, of the anchor node B_j consists of all interconnected anchor nodes via a single-hop connection.

This research delves into the impact of anchor misplacement on sensing coverage efficacy. Assuming random deployment of sensor nodes and location inaccuracies induced by misplaced anchors, we investigate the new types of coverage holes, detect them, and determine the size proportions for coverage holes in relation to the overall area. Moreover, we aim to establish upper and lower bounds on coverage holes in a distributed manner.

Leveraging sophisticated computational geometric constructs such as the Voronoi Diagram (VD) and Delaunay Triangulation (DT), our analysis exploits these powerful structures. Our method for identifying and constraining coverage holes relies on the localities of each convex polygon within the computational architecture representing the sensing field.

In the context of this study, we establish the following assumptions:

- Initially, IoT sensors have the capability to transmit and receive data packets amongst neighboring devices, a crucial aspect for disseminating localized sensor information and facilitating the creation of a distributed computational framework;
- The positions of these sensors are predetermined, enabling the effective development of the VD and DT;

- To ensure successful construction of the DT, no three adjacent sensors align in a straight line;
- The sensing target area is bounded, which is consistent with the majority of IoT application scenarios.

3. Computational Geometry Tools for Analyzing Sensing Coverage: Auxiliary Results

In this section, we introduce auxiliary lemmas and corollaries that will be employed as essential tools for identifying coverage gaps and their corresponding proportions. The foundation of these auxiliary findings is derived from VD and DT. Subsequently, we explore the possibility of achieving full sensing coverage of a target field, denoted as S .

Consider a point p belonging to the field S . We define s_i as a dominant sensor of point p if it holds the shortest distance to p in comparison to all other sensors within S . In other words, $dom(p) = \{s_i \mid d_{s_i,p} = \min(d_{s_j,p}, 1 \leq j \leq n)\}$, where $s_i, s_j \in S$. Let $MaxMin(S) = \max_{p \in S} (d_{dom(p),p})$. Assume $s_i = dom(p)$ has the maximum distance to a point p among all other sensor nodes; if $MaxMin(S) \leq R_{s_i}$, then S is fully covered. However, calculating $MaxMin(S)$ across an infinite number of points p within S is not practically viable. To address this challenge, we employ VD for segmenting the sensing field S into adjacent convex polygons, called cells, denoted as $Vor(s_1), Vor(s_2), \dots, Vor(s_n)$. Each cell $Vor(s_i)$ is connected to only one sensor s_i , with $1 \leq i \leq n$, as illustrated in Figure 8.

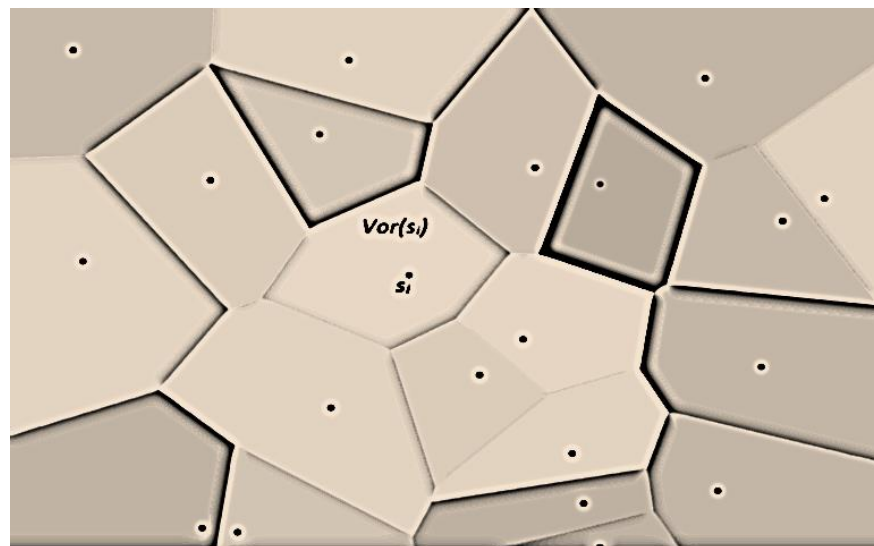


Figure 8. VD partitions sensing field into convex cells.

The perpendicular bisector of the line segment linking sensors 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 $h(s_j, s_i)$ denotes the half-plane that contains s_j . Note that a point $p \in h(s_i, s_j)$ if and only if $d_{s_i,p} < d_{s_j,p}$. 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 its neighborhood, i.e., $NH(s_i)$. Each bisector line segment is called an edge, and the endpoints of this edge are called vertices. For any point p in $Vor(s_i)$, $1 \leq i \leq n$, s_i is the closest sensor to p . Note that if p is on a common edge of two neighboring polygons, then it is equidistant from the two sensors associated with these polygons [1].

The following lemma provides the necessary and sufficient conditions to have full coverage in VD.

Lemma 1 ([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_i} to at least one of their associated sensors, s_i .*

The coverage problem of sensing field S is now converted, by Lemma 1, from checking a non-finite set of points in S into testing a finite set (The size of this set is at most $2n - 5$ [18]) of points representing the cell's VD vertices. This lowers the computational cost, adding to the feasibility of the solution. If $\text{MaxMin}(S) \leq R_{s_i}$ is maintained, the full coverage of S is guaranteed. Thus, VD is a powerful tool to show the existence of coverage holes in WSNs. However, VD is unable to quantify the area size of coverage holes. This is due to the fact that 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.

Therefore, a more effective structure is required to manage and monitor the borderlines of individual coverage gaps. To accomplish this objective, it becomes essential to convert each VD cell into elemental structures that facilitate local and precise tracking of such coverage holes. This transformation entails triangulating Voronoi cells such that sensor vertices configure generated triangles. The edge of a triangle, $s_i s_j$, is formed if $\text{Vor}(s_i)$ and $\text{Vor}(s_j)$ have a Voronoi edge e in common, perpendicular to e and bisected by e . This process generates DT, which produces angle-optimal planar triangles such that the circle that circumscribes any triangle, with non-collinear sensors, is devoid of any other sensors. It is noteworthy that the convexity property of VD persists in DT, as each triangle constitutes a basic convex polygon. Figure 9 demonstrates various Voronoi cell triangulations. Subsequently, we present a corollary linking the coverage problem to DT's edges.

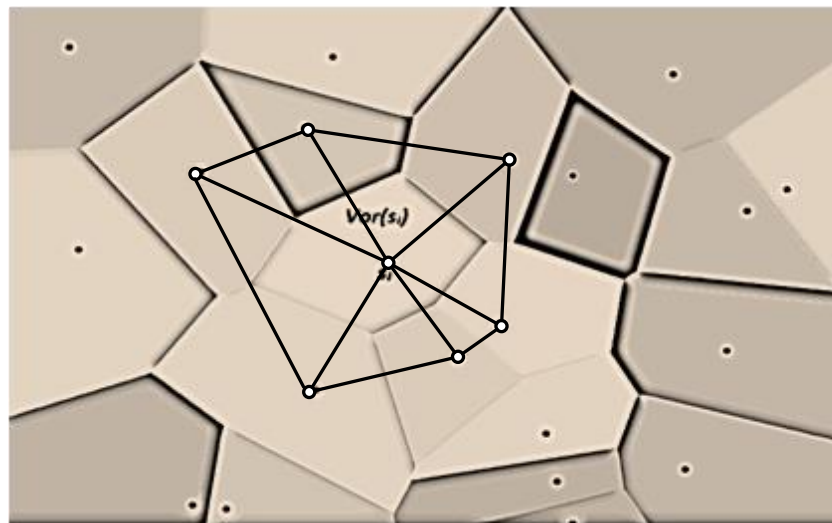


Figure 9. Sample DT of VD: Triangulations incident to sensor s_i .

Corollary 1 ([1]). *If a sensing field is fully covered, the distance of each edge, $s_i s_j$, within the triangles does not exceed $R_{s_i} + R_{s_j}$.*

Considering the Delaunay triangulation of sensing field S , let s_i and s_j be two triangle vertices that have sensing ranges R_{s_i} and R_{s_j} , respectively, then the following lemma holds.

Lemma 2. *Let Δ be a Delaunay (acute) triangle with vertices s_i, s_j , and s_k , and let r be a radius of the circle that circumscribes Δ . Δ is fully covered if and only if the following formula holds.*

$$R_{s_i}^2 + R_{s_j}^2 + R_{s_k}^2 \geq \frac{9\sqrt{3}}{2\pi} r^2 \quad (6)$$

Proof. Assume that Δ is completely covered. The circle with radius r that circumscribes Δ has a center denoted by c , which represents the farthest point from vertices s_i, s_j , and s_k . The distance between any of these three vertices and c equals to r . Each of the sensor vertices will contribute to covering the intra-triangle of Δ . This contribution depends on the sensing range of each sensor vertex. Figure 10 illustrates the concept of sensing and intra-triangle coverage. Since Δ is covered, then every point within Δ including c is covered as well. Since Δ is fully covered, the sum of contributions of the three sensors must cover the entire intra-triangle of Δ . This is formulated as follows:

$$CNT(s_i, \Delta) + CNT(s_j, \Delta) + CNT(s_k, \Delta) \geq A(\Delta) \tag{7}$$

where $CNT(s_l, \Delta)$ denotes the contribution of sensor s_l in covering the area of triangle Δ . This contribution represents a sector area with a radius R_{s_l} .

$$CNT(s_l, \Delta) = \frac{\alpha}{2} R_{s_l}^2 \tag{8}$$

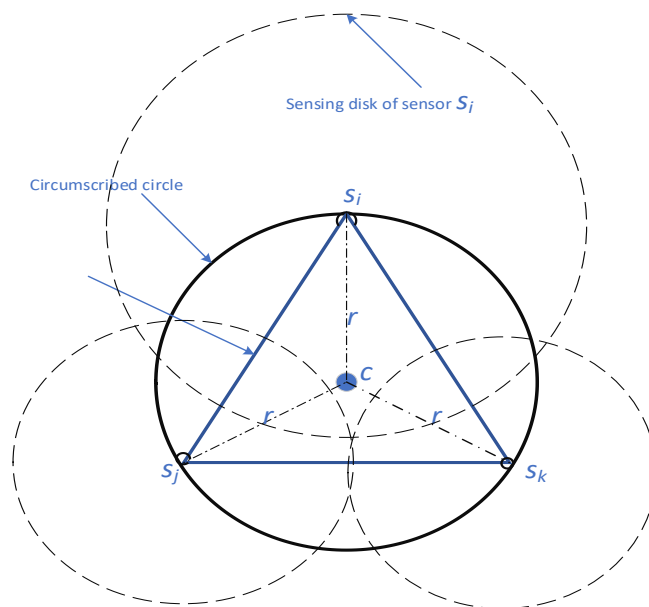


Figure 10. An illustration of sensing coverage using the concept of the intra-triangle contribution of a sensor node.

For acute triangulation, $\alpha = \frac{\pi}{3}$. $A(\Delta) = \frac{\sqrt{3}}{4} a^2$, where a is Δ 's side length. Consequently, (7) can be written as follows:

$$\frac{\pi}{6} R_{s_i}^2 + \frac{\pi}{6} R_{s_j}^2 + \frac{\pi}{6} R_{s_k}^2 \geq \frac{\sqrt{3}}{4} (\sqrt{3} r)^2 \tag{9}$$

This completes the proof of the “if” part.

To prove the (only if) part, assume Formula (6) is held. Let us prove the full coverage of Δ by contradiction. Assume that Δ is not fully covered. Consequently, there exists a point p in Δ that is not covered by any sensor, i.e., $d(p, s_l) > R_{s_l}$, where $l \in \{i, j, k\}$. This means

$$CNT(s_i, \Delta) + CNT(s_j, \Delta) + CNT(s_k, \Delta) < A(\Delta)$$

□

In the case of homogeneous sensors, we derive the following corollary about a lower bound for sensing coverage.

Corollary 2. *In the domain of cooperative sensing, utilizing a network of homogeneous sensors with a sensing range R_{s_i} , full coverage is effectively achieved when the following formula holds.*

$$R_{s_i}^2 \geq \frac{3\sqrt{3}}{2\pi} r^2 \tag{10}$$

As depicted in Figure 11, the minimum sensing ranges for both homogeneous and heterogeneous sensor types to attain full sensing coverage within triangular region Δ are illustrated. For instance, homogeneous sensors necessitate a sensing range of no less than $0.91r$, which effectively guarantees full sensing coverage, as shown by the circle-marked blue curve. Conversely, to accomplish full sensing coverage with heterogeneous sensors, the square root of the left-hand side of Equation (10) must adhere to the boundary defined by the red curve.

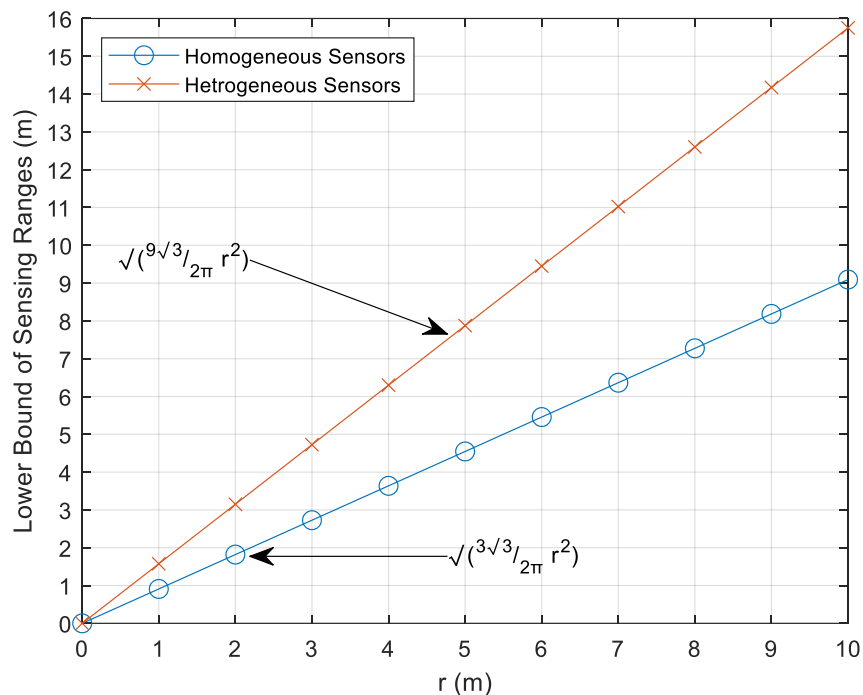


Figure 11. An illustration of sensing ranges lower bounds presented by Lemma 2 and Corollary 2.

Next, we investigate how to detect and define the bounds of each uncovered area in DT.

4. Intra-Triangle Coverage: Beyond Ideality

By employing DT, the coverage issue is systematically converted into an analysis of individual triangle coverage within the comprehensive triangulation. Lemma 2 asserts that a failure to satisfy its condition results in an uncovered area within the circumcircle of Δ . With respect to the maximum angle θ in Δ , we can distinguish three distinct circumcenter scenarios: residing within Δ when $\theta < \frac{\pi}{2}$, outside Δ when $\theta > \frac{\pi}{2}$, or on the longest side opposite to $\theta = \frac{\pi}{2}$. Despite DT offering superior optimal-angle planar triangles (angles approximating $\pi/3$), random deployment scenarios may yield angles larger than $\pi/3$. This gives rise to the following question: What constitutes the minimum sensor density for well-behaved DT?

Consider R_{s_i} as the minimum sensing range among all IoT sensors present within target field S . Under optimal conditions, all angles of Δ equate to $\pi/3$ (forming an equilateral triangle) with the triangle's side length represented by $a = \sqrt{3}R_{s_i}$ [19]. Consequently, the triangle's area in this instance is $\frac{3\sqrt{3}}{4}R_{s_i}^2$. Additionally, any triangulation includes $2N - 2 - \beta$ triangles, with N representing sensor quantity and β indicating sensors lo-

cated along the convex hull boundary of field S [20]. Assuming sensing field S exhibits dimensions $L \times L$, each triangle must cover an area size of $\frac{L^2}{2N-2-\beta}$. Thus, we deduce the following equation:

$$\frac{L^2}{2N-2-\beta} < \frac{3\sqrt{3}}{4} R_{s_i}^2 \tag{11}$$

This gives

$$\frac{2L^2}{3\sqrt{3}R_{s_i}^2} + \frac{2+\beta}{2} < N \tag{12}$$

We assume that the minimum density is achieved. As demonstrated in the proof of Lemma 2, the coverage contribution of a sensor s_i corresponds to the size of the angular sector centered at s_i with radius R_{s_i} . Calculating the contribution of s_i in Δ requires the angle at s_i as shown in (8). Since the lengths of all edges of Δ are known, we use the cosine formula to extract the angle at each sensor.

$$\alpha = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \tag{13}$$

where a, b , and c are the lengths of Δ 's sides, and α is the angle opposite to the side of length c . Therefore, α is plugged into (8) to get $CNT(s_i, \Delta)$, where α is the angle at s_i in a triangle Δ . The following formula gives the intra-triangle coverage of Δ , denoted by $ITC(\Delta)$.

$$ITC(\Delta) = \sum_{s_i \in V(\Delta)} CNT(s_i, \Delta) - A_{mut}, \tag{14}$$

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

Next, we incorporate our intra-triangle coverage examination to detect coverage gaps within a target sensing field.

5. A Deeper Look at the Effect of Anchor Misplacement on Sensing Coverage

We revisit the concept that anchor misplacement results in two distinct types of coverage gaps: inaccurately perceived and hidden, unreported true coverage holes. To tackle these emerging coverage gaps near the sensor nodes affected by anchor misplacement (i.e., localized using misplaced anchor nodes), we employ our analysis from the preceding section. Let $N' \subseteq N$ represent the set of impacted sensor nodes. Furthermore, let C_{s_i} denote the actual sensing coverage area encompassed by sensor node s_i . We now present several crucial auxiliary definitions.

Definition 2. The collective actual sensing coverage (C_{act}) of all affected sensor nodes in WSN is defined as a union of their physical sensing coverage in the target field. 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 an inaccurate location, creating a *perceived* coverage around s'_i . Further, let $C_{s'_i}$ denote the *perceived* sensing coverage area that is covered by the affected sensor node s_i as if s_i is in s'_i coordinates.

Definition 3. The collective perceived sensing coverage (C_{per}) of all affected sensor nodes in WSN is defined as follows. $C_{per} = \bigcup_{i=1}^{|N'|} C_{s'_i}$.

In order to analyze and distinguish between C_{s_i} and $C_{s'_i}$, as well as to identify the scenarios resulting in different types of coverage holes, we employ DT in examining the problem within the vicinity of each impacted sensor node. We utilize a distributed algorithm, as described in [21], to construct the DT representing the targeted sensing field

S. Lemma 2 offers a suitable criterion for demonstrating the presence of coverage holes. Based on the evaluation presented in Section 4, subtracting $ITC(\Delta)$ from Δ 's full area size reveals the unmonitored region within Δ as follows:

$$UNC(\Delta, G) = A(\Delta) - ITC(\Delta) \tag{15}$$

where $A(\Delta)$ refers to the area size of Δ and is given by the following equation.

$A(\Delta) = \sqrt{d(d-a)(d-b)(d-c)}$, where $d = \frac{a+b+c}{2}$ and $a, b,$ and c are the length of the sides of Δ .

Assume s_2 is a sensor node affected by anchor misplacement, with its incorrect estimated location represented by s_2' , having coordinates (x', y') . Let $\vec{v} = (\Delta x, \Delta y)$ be the corresponding localization error vector. The adjusted coordinates for s_2' 's position are $(x' - \Delta x, y' - \Delta y)$. Figure 12 illustrates the structural modification of the Delaunay triangle due to anchor misplacement's influence on localizing sensor s_2 .

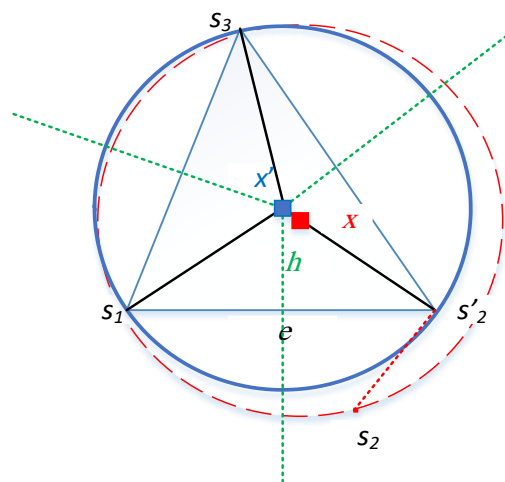


Figure 12. An example of structural change on DT due to correcting the location of s_2' to s_2 .

To evaluate the impact of anchor misplacement on sensing coverage gaps, we must assess the sensing coverage under two different scenarios: in the presence and absence of anchor misplacement. Specifically, for each sensor node s_i affected by anchor misplacement, we determine the coverage gap by contrasting the sensing coverage of s_i and its neighboring nodes with that of s_i' and its surrounding nodes. This enables us to compare C_{s_i} and $C_{s_i'}$ in their vicinities. Figures 5 and 6 show the vicinity of affected sensor s_1 (i.e., $s_2s_3s_4s_5s_6s_7$) and the vicinity of s_1' (i.e., $s_2's_3's_4's_5's_6's_7'$). The triangulation process within these vicinities allows us to examine the uncovered areas in each triangle.

Next, we employ the notion of history in graph theory to illustrate our analysis further and compute the sensing coverage for every affected sensor node in both scenarios: when anchor misplacement is present and when it is absent.

5.1. Anchor Misplacement as a Graph Operator

Let D_T be a Delaunay Triangulation of IoT sensors in the target field with no anchor misplacement. Such misplacement triggers alterations in D_T , subsequently influencing the localization accuracy pertaining to sensor nodes. The change in D_T can manifest in either the length metric of the edges or modifications in the D_T structure, as certain sensors establish or extinguish connections based on their erroneous locations. Let D_T' denote the new triangulation after anchor misplacement.

Consequently, anchor misplacement functions as a graph operator capable of mapping an input-provided graph (i.e., D_T) onto a distinct graph, D_T' . For an impacted sensor $s_i \in D_T$, its counterpart $s_i' \in D_T'$ signifies accurate localization prior to anchor misplace-

ment. However, erroneous localization post-anchor misplacement would classify it as s_i' . Thus, from a geolocation standpoint, s_i and s_i' can be perceived as distinct sensors.

Given $D_T(s_i)$, denote the set of all triangles within D_T induced by s_i and its neighbors, i.e., $NH(s_i)$. Analogously, $D_T'(s_i')$ represents the set of all triangles in D_T' induced by s_i' and its neighbors. For instance, the triangulation process of $D_T(s_1)$ includes all triangles in D_T induced by s_1 and its proximate nodes, denoted as $NH(s_1) = \{s_2, s_3, s_4, s_5, s_6, s_7\}$ in Figures 5 and 6. Consequently, we term $D_T(s_i)$ the graph history of $D_T'(s_i')$ due to its preceding graph state prior to anchor displacement. This notation is represented as $D_{Th}'(s_i') = D_T(s_i)$. The notion of graph history has been previously explored in various graph theory domains, such as characterizing the asymptotic behavior of iterated line and path graphs [22,23].

Clearly, the subgraphs $D_T'(s_i')$ and their history counterpart, $D_{Th}'(s_i')$, may exhibit dissimilarities. Indeed, the inaccurate localization of sensors results in the presence of certain vertices in $D_T'(s_i')$ but not in $D_{Th}'(s_i')$, or vice versa. The precise positioning of each sensor within the target field serves as a crucial aspect of our investigation, as both subgraphs $D_T'(s_i')$ and $D_{Th}'(s_i')$ could be isomorphic but yet display variations concerning edge lengths. Two graphs are isomorphic if they contain the same objects (i.e., vertices) linked in the same way. A demonstration of this observation can be found in Figure 13, which depicts one triangle of $D_T'(s_2')$ alongside its graph history, $D_{Th}'(s_2')$.

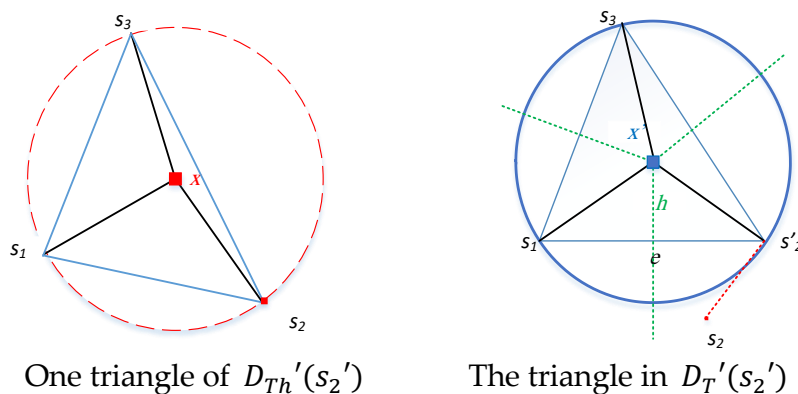


Figure 13. A partial snapshot of $D_T'(s_2')$ and its history.

To construct D_{Th}' from D_T' , we proceed as follows: Firstly, identify the misplaced anchor nodes. Subsequently, eliminate the impacted sensor nodes s_i , along with their associated edges, and reintroduce them in their accurate positions. Ultimately, form the triangulation within their vicinity. The average number of neighbors for s_i , denoted as $|NH(s_i)|$, does not surpass six; thus, the average quantity of triangles in both subgraphs $D_T'(s_i)$ and $D_{Th}'(s_i)$ remains below six [20]. These findings show the minimal computational cost entailed by our methodology.

5.2. Identifying Coverage Holes under Anchor Misplacement

In this subsection, we detect/delineate the inaccurately perceived and hidden unreported coverage holes. To accomplish this, we focus on the mutual triangles present in both $D_T'(s_i')$ and $D_{Th}'(s_i')$. An empty intersection signifies that $D_{Th}'(s_i')$ corresponds to an entirely novel structure, with none of the $NH(s_i)$ elements belonging to $D_T'(s_i')$. This scenario arises when anchor misplacement results in an extremely inaccurate location for sensor s_i , causing the estimated position s_i' to fall outside of s_i 's proximity. We recognize the subsequent classifications of a potential shared triangle Δ existing between $D_T'(s_i')$ and $D_{Th}'(s_i')$:

1. Full local coverage of Δ is maintained in both $D_T'(s_i')$ and $D_{Th}'(s_i')$;
2. Full local coverage of Δ exists in $D_{Th}'(s_i')$, but not in $D_T'(s_i')$. This is the case of false perceived coverage hole;
3. There is no full coverage in both $D_T'(s_i')$ and $D_{Th}'(s_i')$;

- The full local coverage of Δ exists in $D_T'(s_i')$, but not in $D_{Th}'(s_i')$. This is the case of hidden actual unreported coverage holes as shown in Figure 14.

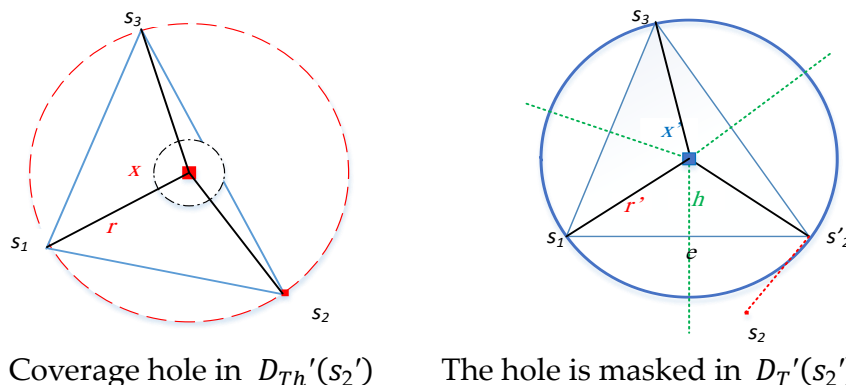


Figure 14. Actual unreported coverage hole with center x .

Categories 1 and 3 deal with extreme cases where the triangle Δ is either covered in both $D_T'(s_i')$ and its history or not.

Our primary objective involves detection, identification, and quantification of coverage gaps corresponding to categories 2 and 4. In Section 3, we present a comprehensive analysis that effectively detects such coverage anomalies. Subsequently, we introduce an intra-triangle coverage technique in Section 4, which serves as an effective tool for estimating the magnitude of the coverage hole. Leveraging the concept of history delineated in Section 5.1, we identify the type of coverage gap. We then proceed to compute the proportionality between perceived sensing coverage, C'_{s_i} , to the actual sensing coverage, C_{s_i} . This coverage ratio is designated as CRC1-CRC4 for categories 1–4, respectively.

$$\begin{aligned}
 CRC1 &= \frac{Area(\Delta, D_T'(s_i'))}{Area(\Delta, D_{Th}'(s_i'))} \\
 CRC2 &= \frac{UNC(\Delta, D_T'(s_i'))}{Area(\Delta, D_T'(s_i'))} \\
 CRC3 &= \frac{UNC(\Delta, D_T'(s_i'))}{UNC(\Delta, D_{Th}'(s_i'))} \\
 CRC4 &= \frac{UNC(\Delta, D_{Th}'(s_i'))}{Area(\Delta, D_{Th}'(s_i'))}
 \end{aligned} \tag{16}$$

In the context of graph G , $Area(\Delta, G)$ denotes the area of triangle Δ , while $UNC(\Delta, G)$ signifies the magnitude of uncovered space within Δ , as per Equation (15). It is crucial to recognize that if the ratio does not equal 1 for both categories 1 and 3, inaccuracies in sensing reporting from Δ sensor nodes may occur. Under such circumstances, the actual coverage in the history graph could be either overestimated or underestimated.

The methodology remains applicable even when multiple sensors within triangle Δ are impacted. Let us consider s_i and s_j as two affected sensors and neighbors in $D_{Th}'(s_i')$. Redundancy in calculations arises due to shared triangles by $D_{Th}'(s_i')$ and $D_{Th}'(s_j')$, leading to $ITC(\Delta)$ being computed twice since both s_i and s_j are affected. Consequently, it is necessary to adjust the overall ITC value accordingly within both $D_T'(s_i')$ and $D_{Th}'(s_i')$.

The coverage ratio expressed in Equation (16) is straightforward. By applying it to intra-triangle scenarios, we can identify varying coverage hole types caused by anchor misplacement and compute their respective percentages. Subsequently, our resilience framework enables detection of misplaced anchors and determination of impacted sensor node sets.

5.3. Calculating Lower and Upper Bounds of Coverage Holes

The intra-triangle coverage gap zones, or uncovered regions, exhibit varying geometric shapes. Nonetheless, we choose to represent the lower and upper boundaries of each

uncovered region within a triangular area Δ via circular models. The lower boundary circle is defined as a circle with its center at the polygon's centroid, which exclusively encompasses the uncovered region in Δ and serves as the largest circle inscribed within this space. Conversely, the upper boundary circle pertains to the smallest circumscribing circle that completely surrounds the uncovered region of Δ . To calculate the lower and upper boundaries for Δ 's uncovered region, we employ the subsequent methodology: initially, we find a set U consisting of intersection points—those situated between angular sectors and Δ 's perimeters, along with any intersection points existing among angular sectors themselves. Let s_i and s_j be two vertices in Δ . If $R_{s_i} + R_{s_j} > d(s_i, s_j)$, we dismiss intersection points located between circles centered at s_i and s_j and the edge $s_i s_j$. Let U' represent the revised set comprising intersection points. The points contained in U' formulate a polygon P . Our objective lies in determining the minimal/maximal circle that circumscribes/inscribes polygon P . To accomplish this, it is imperative to pinpoint centroid c of this specific polygon with coordinates derived from the following formula [24].

$$c_x = \frac{1}{6A} \sum_{i=0}^{|U'|-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (17)$$

$$c_y = \frac{1}{6A} \sum_{i=0}^{|U'|-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (18)$$

where A is the area and 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 $R_l = \text{Min}_{p_i \in U'} d(c, p_i)$. The circle centered at c with radius R_l represents a lower bound of the uncovered area in Δ . Likewise, let $R_u = \text{Max}_{p_i \in U'} d(c, p_i)$. Then πR_u^2 represents the size of the minimum circle that circumscribes P and, thus, serves as an upper limit for the uncovered area within Δ .

Both bounding circles are centered at the centroid of a polygon P encompassing the uncovered area. Utilizing centroid c , as opposed to the circumcenter of Δ , proves more efficient for these reasons: (1) IoT sensor range variability causes the circumcenter of Δ to not always belong to the uncovered area. (2) If the circumcenter resides outside Δ , intra-triangle coverage calculations become irrelevant. (3) Centroid c -based bounds offer tighter representation as they more accurately represent the uncovered area. Deriving from our analysis, below is a summary of steps for calculating lower and upper bounds within this advanced Algorithm 1.

Algorithm 1 [1]: Lower and Upper Bounds (LUB)

Input: triangle Δ
Output: c , $lowerBound$, $upperBound$
 If $HasCoverageHole(\Delta)$ then
 $P = findPolygon(\Delta)$;
 $c = findCentroid(P)$;
 $R_l = findRadiusLowerBound(P, c)$;
 $R_u = findRadiusUpperBound(P, c)$;
 $lowerBound = \pi R_l^2$;
 $upperBound = \pi R_u^2$;
 return c , $lowerBound$, $upperBound$;
 End if

In the LUB algorithm, it is assumed that every sensor has been localized and their positions are identified. During the construction of DT, each sensor recognizes its adjacent neighbors in every triangle Δ present in DT. The LUB algorithm commences by verifying the presence of a coverage hole through the $HasCoverageHole(\Delta)$ function, which evaluates Δ against the coverage criteria detailed in Lemma 2. Upon identifying a coverage hole, the $findPolygon(\Delta)$ function comes into action to find the polygon that strictly circumscribed

the uncovered. The *findCentroid* (P) method employs Equations (17) and (18) to determine P 's centroid. The subsequent step involves calling the *findRaduisLowerBound* (P, c) function to compute the shortest distance between P 's vertices and c , signifying the radius of the lower bound. In a similar manner, *findRaduisUpperBound* (P, c) yields the greatest distance between P 's vertices and c .

6. Resiliency Approach for Misplaced Anchors

In this Section, we propose a distributed scheme that takes input from affected sensors during the validation process, and outputs the set of misplaced anchors.

6.1. The Proposed Scheme

The proposed methodology focuses on establishing the accuracy of localization and subsequently, the validity of sensing coverage by pinpointing misplaced anchors via their impacted sensors. This approach is carried out through a two-phase process: **Phase 1 (detection of affected sensors)** involves each node calculating its distance to all other nodes through two methods: (a) Euclidean distance, and (b) using received signal strength (RSS). Denote d_{s_i, s_j} and d_{s_i, s_j}' as the distance between nodes s_i and s_j utilizing Euclidean distance and RSS, respectively. We introduce $x^{(i)}$ as the distortion vector between d_{s_i, s_j} and d_{s_i, s_j}' for node s_i in relation to all other nodes s_j as illustrated below.

$$x^{(i)} = \left\| d_{s_i, s_j} - d_{s_i, s_j}' \right\|^2 \quad (19)$$

where $j \in [1, n]$, $j \neq i$. It is clear that the total number of vectors corresponds to a multiple of the sensor nodes count. In order to calculate the average two-way distances, which include d_{s_i, s_j} , d_{s_j, s_i} , d_{s_i, s_j}' , and d_{s_j, s_i}' , an extensive computational effort is necessary. This leads to increased latency in identifying misplaced anchors. To address this challenge, we investigate two scenarios:

(1) Computation of a single vector originating from one sensor to all other sensors. For enhanced efficiency, we strategically select a sensor node situated near the target field's center;

(2) Calculation of multiple vectors followed by obtaining the average for two-way distances.

Regardless of the case employed, Phase 1 will yield a single vector output. Subsequently, we arrange this vector in descending order based on distortion levels from high to low. Employing the K-means algorithm with $K = 2$ clusters, we classify the vector elements into two distinct categories: sensors with relative accuracy and those affected by misplaced anchors.

In **Phase 2 (determining misplaced anchors)**, the objective is to leverage the outcomes of the K-means algorithm from the prior phase to pinpoint misplaced anchor sets. Among the two produced classes, the one exhibiting a high mean localization error signifies impacted sensors. As a result, misplaced anchors reside within those that have localized the affected sensors. To establish the misplaced anchor set, let $BL(s_i) = \{b_k, k = \{1, 2, 3\}\}$ represent a collection of anchors involved in localizing sensor node s_i via the trilateration technique. Additionally, let $S_a \subseteq S$ be a subset of relatively precise sensor node locations. We delineate benign anchors, B_b , as a collective set of anchor nodes participating in localizing all relatively accurate sensors as follows.

$$B_b = \bigcup_{s_i \in S_a} BL(s_i) \quad (20)$$

Consequently, the misplaced anchors, B_m , belong to the complement set as follows. $B_m = B \setminus B_b$. It is crucial to recognize that B_m and B_b are not necessarily mutually exclusive. In fact, these misplaced anchors may contribute to the localization of certain sensors, even though the K-means algorithm deems their positions relatively accurate. Next, we enhance the detection scheme to address this particular aspect.

6.2. Enhanced Detection Scheme

The earlier proposed methodology had the potential to yield an increased rate of false alarms regarding incorrectly positioned anchors. To address this issue, we propose the subsequent refinement process. Firstly, we organize the category/cluster of relatively precise sensors, arranged in descending order according to their localization inaccuracies. Next, we examine each sensor within the sorted group individually.

Considering s_i as the sensor with the most significant localization error amidst others in the ordered category, we look into b_1 , b_2 , and b_3 —the anchors responsible for localizing s_i . We then assess these three anchors based on specific criteria: should any of these anchors be nearer to the influenced group's centroid rather than that of the relatively precise sensors cluster, and exhibit a distortion greater than a particular threshold compared to a benign anchor, it is then reclassified as B_m . This threshold depends on historical information accounting for SNR. In evaluating an anchor, b_j , we employ a technique similar to our earlier discussion concerning distortions between Euclidean and RSS-based distances as highlighted below:

$$\left\| d_{b_i, b_j} - d_{b_i, b_j}' \right\|^2 \geq \varepsilon \quad (21)$$

where b_i denotes a benign anchor. We continue repeating the second step as long as there is at least one anchor participating in localizing sensor s_i that fulfills the specified criteria. To guarantee this procedure's efficiency, we select b_i from among benign anchors that have contributed to localizing high-accurate sensors.

7. Numerical Results and Discussion

7.1. Simulation Results

In this section, we rigorously evaluate our resiliency methodology and demonstrate its effectiveness. Our analysis involves experimental trials on both a homogeneous and a heterogeneous network, leveraging the devised framework to assess network dependability. The aggregate of simulation outcomes is achieved by iteratively performing the experiments and computing the mean values from the distinct results. This specific figure correlates to result convergence and fluctuation. For instance, if the outcomes reach stability and exhibit minimal variability after x iterations, then we embrace x (or a value exceeding x) as an optimal number of repetitions.

7.2. Validation

Initially, we demonstrate the effectiveness of our technique by identifying the impacted sensors and subsequently the misplaced anchors. Our goal is to categorize these two sensor groups utilizing the K-means clustering algorithm. We implement this in the experimental environment illustrated in Figure 2. Upon executing Phase 1 of our method, Figure 15 emerges, presenting the groups of affected and relatively unaffected sensors alongside their centroids. Most affected sensors are accurately integrated into group 1, showcasing our approach's technical proficiency.

We validate our approach by running an enhanced version of Phase 2. We have calculated the localization errors for different trials (1, 2, 7, 11, 16, 20, 27, 40) and found that the results converge after 16 trials. Measuring the variance of localization errors suggests that 20 iterations are enough to provide convergent and stable results. Thus, each experiment is repeated 20 times for any given network topology. We take a random sample of 15 observations of averaged localization error values generated by the two anchor groups per iteration. To analyze the statistical significance of the difference between the localization errors generated by the two anchor classes, we adopt the following hypotheses:

- Null hypothesis H_0 : $\mu = 0$, which means the true difference of means for the corresponding anchor groups is equal to zero;
- Alternative hypothesis H_a : $\mu \neq 0$.

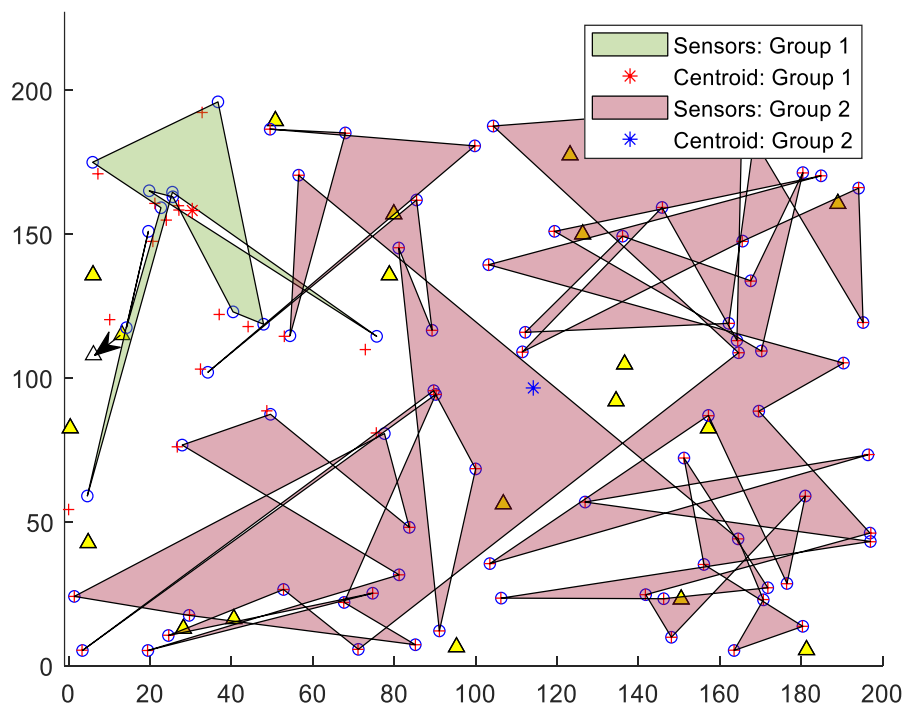


Figure 15. An illustration of K-means application on sensors, generating two polyshape regions/groups of sensors: affected and relatively non-affected by anchor misplacement.

We employ a confidence interval of 95%, which is equivalent to a significance level of $\alpha = 0.05$. If the p -value is less than or equal to α , we discard the null hypothesis. This rejection conveys that there is a statistically significant difference between the localization errors induced by the two anchor classes, thus favoring the alternative hypothesis. In this case, the confidence interval does not contain zero.

According to our analysis, with a p -value as low as 1.9551×10^{-4} , we can confidently reject the null hypothesis as it validates significant differences in mean values between anchor classes. The lower and upper limits of confidence intervals for mean differences fall between -4.505 and -3.4446 , further supporting this conclusion.

Our results show a validation of our approach in identifying misplaced anchors with a confidence level of 95%.

7.3. Verify the Effectiveness

We demonstrate our framework’s resilience against unreliable nodes in this section. By effectively mitigating the impact of misplaced anchors on IoT nodes, we aim to minimize their involvement in the localization process. We employ the NS-3 simulator to model various experimental scenarios featuring randomly dispersed, non-uniform IoT sensors across the target field. These experiments intend to show our framework’s efficacy. Additionally, we utilize a distributed algorithm outlined in [21] to construct a DT that accurately represents the target sensing field. A comprehensive overview of the simulation settings can be found in Table 2.

Table 2. Simulation settings.

Parameter	Range
Target Field Area	300 × 300 m ² (divided into 2 × 2 grid)
Number of sensor nodes (per cell)	125
Number of anchor nodes (per cell)	30
Transmission range	50–100 m
Sensing range	5 m (with a variance of 2 m)
Anchor displacement value	7 m (with a variance of 2 m)
SNR	30 db

(1) The Effectiveness of Localization and Perceived Coverage

The objective of this cutting-edge experiment is to investigate the influence on Root Mean Square Distance (RMSD) and the perception of coverage gaps as the proposed framework identifies and discards an increasing number of misplaced anchors. We employ RMSD as a performance metric to quantify the localization inaccuracies present in the network.

$$RMSD = \sqrt{\frac{\sum_{j=1}^n \left((x_j - x_j')^2 + (y_j - y_j')^2 \right)}{n}} \quad (22)$$

where $(x_j - x_j')$ and $(y_j - y_j')$, respectively, are the actual and estimated positions of sensor node j and n is the total number of sensor nodes.

As illustrated in Figure 16, our framework demonstrates an effective ability to reduce both the RMSD and the percentage of perceived coverage gaps as the count of identified anchor nodes increases. The perceived coverage reduction exhibits a sharper decline when compared to RMSD, attributable to its direct correlation with the number of misplaced anchors within the network. Conversely, RMSD is primarily influenced by multiple factors, such as sufficient reliable anchor nodes per sensor and the SNR impact on distance measurement quality.

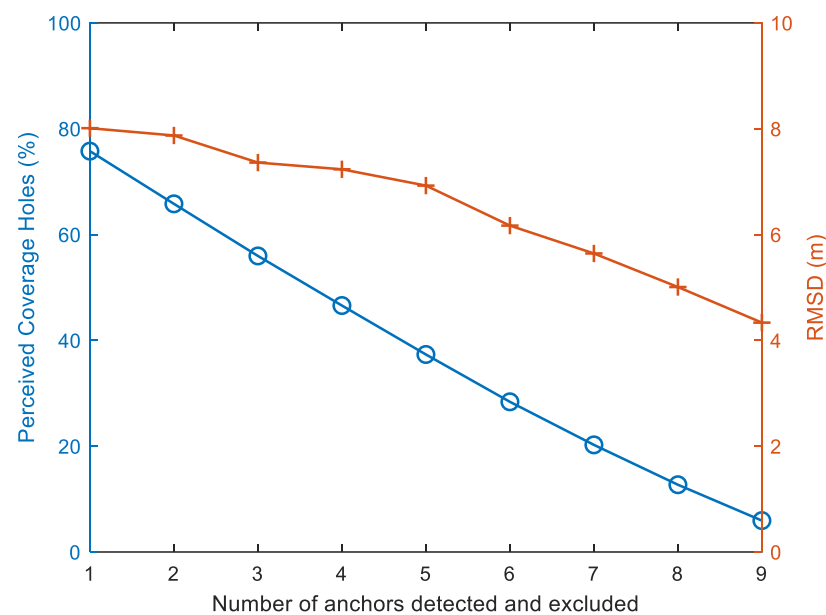


Figure 16. The behavior of RMSD and perceived coverage as the number of anchors detected and excluded increase.

(2) The Effectiveness of Perceived Coverage Holes for Different Sensing Range Values

In order to demonstrate the efficacy of our novel approach when applied to a heterogeneous network consisting of varying sensing range values, we conducted simulations using the following scenario: 40 distinct objects within a terrain, such as gas pipelines. Sensor nodes have been strategically placed to ensure comprehensive monitoring for gas leaks throughout all pipes. Our study assumes that anchor-based localization is utilized for sensor positioning. In cases where anchor misplacement occurs, we aim to highlight the influence of our proposed method on the percentage of misreported objects. How significantly does our proposed framework contribute to mitigating false coverage perception as an increasing number of misplaced anchors are identified and eliminated from network services? The findings of this investigation are presented in Figure 17.

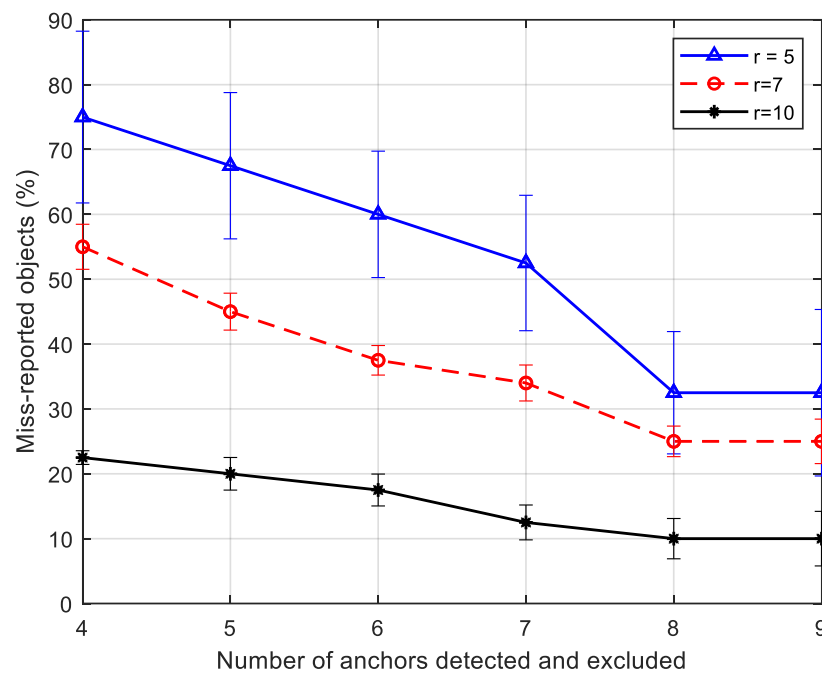


Figure 17. Number of misplaced anchors vs. percentage of misreported objects.

The 95% confidence interval reveals that the proportion of misreported objects within the three curves becomes statistically negligible as an increasing number of misplaced anchor nodes are identified and removed. This occurs due to the fact that, irrespective of the sensing range, the elimination of more misplaced anchors leads to an enhanced validity in sensing reports and subsequently a decrease in misreported objects. Our findings indicate that employing a smaller number of sensing nodes with larger sensing ranges results in a more rapid recovery from anchor misplacements. In essence, the proposed framework considerably reduces the number of misreported objects as the sensing range expands, showcasing its technical efficiency.

(3) The Effectiveness of Perceived Coverage Holes for Different SNR Values

Through a series of experiments, we effectively show the high capabilities of our novel framework at varying SNR levels, namely 10, 20, and 30 dB. The primary objective is to evaluate the influence of anchor misplacement and measurement inaccuracies on localization precision. As depicted in Figure 18, our cutting-edge framework exhibits superior performance with higher SNR levels. This performance can be explained as follows: as the SNR values decrease, the measurement errors escalate due to weakened signals, undermining the reliability of the localization system. Interestingly, for an SNR of 10 dB, the RMSD remains relatively stable—excluding additional misplaced anchors does not affect the RMSD. Furthermore, the application of a 95% confidence interval unveils the

statistical significance of RMSD values for diverse SNR trajectories, thus highlighting the crucial role that reduced SNR values play in determining sensing dependability.

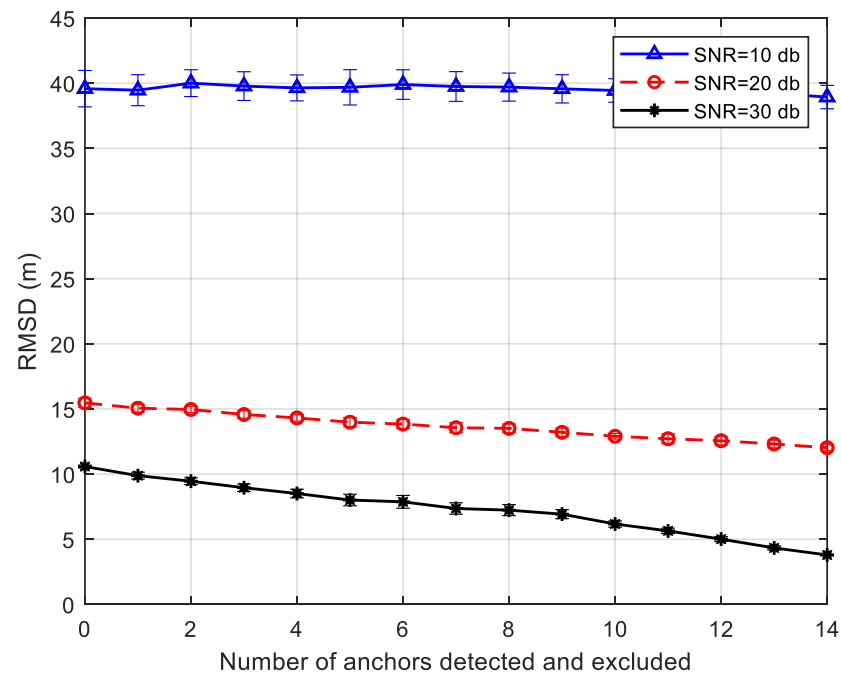


Figure 18. Number of misplaced anchors detected and excluded vs. localization RMSD.

8. Conclusions

In this paper, we proposed a novel resilience framework for detecting anchor misplacement, identifying coverage holes/gaps, and accurately measuring their dimensions. To achieve this, we performed a comprehensive computational geometry-based examination of the targeted sensing field, taking into account various error components, particularly anchor misplacement. By implementing Voronoi Diagrams (VD) and Delaunay Triangulation (DT), we strategically divided the sensing field to effectively evaluate coverage deficiencies. Utilizing VD and DT allowed us to address the coverage gap issue with a minimal number of points within the target sensing field, subsequently reducing computational expenses and enhancing the feasibility of our approach in IoT environments.

We then evaluated intra-triangle coverage, revealing novel categories of coverage holes: actual unreported and perceived coverage holes. The experimental results demonstrated the validity and efficiency of our proposed resilience framework in attaining localization precision and sensing coverage while bolstering overall IoT-based system dependability. These findings provide valuable insights for cooperative and overlapping IoT systems to optimize resource allocation by either tolerating coverage loss or deploying supplementary sensors to fill the coverage holes.

As future work, we aim to develop a cutting-edge framework that prolongs the lifespan of IoT networks through collaborative sensing and leveraging the intra-triangle analytics provided in this research. For instance, overlapping sensors from multiple network operators could devise an intelligent sleep mode schedule based on each sensor's role. Additionally, mobile sensors may serve as backup support for low-energy sensors while covering actual unreported sensing gaps.

Author Contributions: Conceptualization, Y.A.M., H.S.H. and N.N.; Methodology, Y.A.M.; Software, Y.A.M.; Validation, Y.A.M.; Formal analysis, Y.A.M.; Investigation, Y.A.M.; Writing—original draft, Y.A.M.; Writing—review & editing, H.S.H. and N.N.; Visualization, Y.A.M., H.S.H. and N.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number RGPIN-2019-05667.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al Mtawa, Y.; Hassanein, H.S.; Nasser, N. Identifying Bounds on Sensing Coverage Holes in IoT Deployments. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 4–8 December 2016; pp. 1–6.
2. Mondal, A.; Mishra, D.; Prasad, G.; Hossain, A. Joint Optimization Framework for Minimization of Device Energy Consumption in Transmission Rate Constrained UAV-Assisted IoT Network. *IEEE Internet Things J.* **2022**, *9*, 9591–9607. [[CrossRef](#)]
3. Raza, S.M.; Jeong, J.; Kim, M.; Kang, B.; Choo, H. Empirical Performance and Energy Consumption Evaluation of Container Solutions on Resource Constrained IoT Gateways. *Sensors* **2021**, *21*, 1378. [[CrossRef](#)] [[PubMed](#)]
4. Martínez Rosabal, O.; López, O.L.A.; Pérez, D.E.; Shehab, M.; Hilleshein, H.; Alves, H. Minimization of the Worst Case Average Energy Consumption in UAV-Assisted IoT Networks. *IEEE Internet Things J.* **2022**, *9*, 15827–15838. [[CrossRef](#)]
5. Xu, K.; Takahara, G.; Hassanein, H. On the robustness of grid-based deployment in wireless sensor networks. In Proceedings of the IWCMC06: 2006 International Wireless Communications and Mobile Computing Conference, Vancouver, BC, Canada, 3–6 July 2006; pp. 1183–1188. [[CrossRef](#)]
6. Bottino, A.; Laurentini, A. A nearly optimal algorithm for covering the interior of an Art Gallery. *Pattern Recognit.* **2011**, *44*, 1048–1056. [[CrossRef](#)]
7. Kumar, R.; Amgoth, T.; Sah, D.K. Deployment of Sensor Nodes for Connectivity Restoration and Coverage Maximization in WSNs. In Proceedings of the 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 25–27 March 2021; pp. 209–213. [[CrossRef](#)]
8. Shanthi, D.L. Maximization of Disjoint K-cover Using Computation Intelligence to Improve WSN Lifetime. *Lect. Notes Netw. Syst.* **2022**, *458*, 223–238. [[CrossRef](#)]
9. Touati, H.; Aboud, A.; Hnich, B. Named Data Networking-based communication model for Internet of Things using energy aware forwarding strategy and smart sleep mode. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6584. [[CrossRef](#)]
10. Wang, S.C.; Hsiao, H.C.W.; Lin, C.C.; Chin, H.H. Multi-Objective Wireless Sensor Network Deployment Problem with Cooperative Distance-Based Sensing Coverage. *Mob. Netw. Appl.* **2021**, *27*, 3–14. [[CrossRef](#)]
11. Al Mtawa, Y.; Nasser, N.; Hassanein, H.S. Mitigating anchor misplacement errors in wireless sensor networks. In Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015; pp. 569–575.
12. Sorbelli, F.B.; Das, S.K.; Pinotti, C.M.; Silvestri, S.; Sil, S. Precise Localization in Sparse Sensor Networks using a Drone with Directional Antennas. In Proceedings of the ICDCN'18: 19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018. [[CrossRef](#)]
13. Maheshwari, A.; Chand, N. A Survey on Wireless Sensor Networks Coverage Problems. *Lect. Notes Netw. Syst.* **2019**, *46*, 153–164. [[CrossRef](#)]
14. Farsi, M.; Elhosseini, M.A.; Badawy, M.; Arafat Ali, H.; Zain Eldin, H. Deployment techniques in wireless sensor networks, coverage and connectivity: A survey. *IEEE Access* **2019**, *7*, 28940–28954. [[CrossRef](#)]
15. Sowe, S.K.; Kimata, T.; Dong, M.; Zettsu, K. Managing heterogeneous sensor data on a big data platform: IoT services for data-intensive science. In Proceedings of the 2014 IEEE 38th International Computer Software and Applications Conference Workshops, Vasteras, Sweden, 21–25 July 2014; pp. 295–300. [[CrossRef](#)]
16. Shakhov, V.V.; Koo, I. Experiment Design for Parameter Estimation in Probabilistic Sensing Models. *IEEE Sens. J.* **2017**, *17*, 8431–8437. [[CrossRef](#)]
17. Al Mtawa, Y.; Hassanein, H.S.; Nasser, N. Measuring the validity of sensing coverage in the presence of anchor misplacement. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017.
18. de Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M. Voronoi Diagrams. In *Computational Geometry*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 147–171.
19. Takahara, G.; Xu, K.; Hassanein, H. Efficient coverage planning for grid-based wireless sensor networks. In Proceedings of the 2007 IEEE International Conference on Communications, Glasgow, UK, 24–28 June 2007; pp. 3522–3526. [[CrossRef](#)]
20. de Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M. Delaunay Triangulations. In *Computational Geometry*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 191–218.
21. Delaunay Triangulation C++ Library for 2D and 2.5D Point Clouds. Available online: <https://www.geom.at/products/fade2d/> (accessed on 14 August 2022).
22. Al-Mtawa, Y.; Jazzer, M. On Independence Problem of P2-Graph. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 205.

23. Al Mtawa, Y. Histories of Iterated Path Graphs. *J. Comb. Inf. Syst. Sci.* **2007**, *32*, 175–188.
24. Bourke, P. Calculating the Area and Centroid of a Polygon. 1988. Available online: https://lexrent.eu/wp-content/uploads/torza/artikel_groep_sub_2_docs/BYZ_3_Polygon-Area-and-Centroid.pdf (accessed on 14 April 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.