Coverage Hole Detection Using Lattice-based Approach in Wireless Sensor Networks

Anitha Christy Angelin P.[1], Salaja Silas[2]

Division of CSE, Karunya Institute of Technology and Sciences, Coimbatore, India.

Email: [1]anithachristyangelin1p@gmail.com, [2]salaja_cse@karunya.edu

Abstract

Mission-critical applications such as environmental sensing, battlefield monitoring, and disaster management are increasingly using wireless sensor networks, or WSNs. These networks rely on widely dispersed sensor nodes to monitor and transmit physical conditions in real time when connected to the Internet of Things (IoT). The overall reliability of the network can be impacted by issues like energy depletion, node failure, and environmental damage, which can lead to coverage gaps in places where communication or sensing is interfered with. A lattice-based coverage hole detection method based on a modified discrete computational geometry model is presented in this paper. The suggested approach accurately determines the precise nodes causing coverage holes and determines the size of each uncovered region with high spatial precision by arranging sensor nodes in a lattice configuration and using a triangulation-based detection algorithm. In comparison to traditional coverage hole detection techniques like Delaunay triangulation and simple grid-based methods, simulation results from a 1000-node network show that the suggested method achieves over 93% energy efficiency, extends network lifetime to over 95%, and reduces control packet overhead by more than 90%. These improvements guarantee more reliable data transfer and a longer running life, which makes the technique ideal for extensive, long-term WSN deployments in demanding and dynamic settings.

Keywords: Coverage Hole Network, Sensor Network, Node, Communication, Wireless Sensor Network.

DOI: https://doi.org/10.36548/jtcsst.2025.2.005
Received: 20.04.2025, received in revised form: 17.05.2025, accepted: 02.06.2025, published: 13.06.2025

1. Introduction

Many mission-critical applications, such as surveillance [1], healthcare [2], and warfare [3], have had a significant impact on research advancements in wireless sensor networks over the past decade. Motes are tiny embedded devices that form the foundation of wireless sensor networks (Tosun et al., 2023). These devices can be distributed randomly or uniformly across the region of interest. Using multi-hop routing techniques, gateways receive redirected data during the data transmission process. Sensor nodes are used to transmit critical data for many critical software applications, including industrial monitoring, disaster response [4], fire detection [5], intruder identification in combat [6], and healthcare [7]. However, external factors (Nandi et al., 2023) or battery drain [8] could jeopardize these nodes. This can create gaps in coverage within the region of interest, as illustrated in Figure 1. Moreover, because of random environmental factors, nodes can deviate from their designated positions. These gaps are detrimental to wireless sensor network performance, influencing its lifespan and bandwidth. Coverage gaps can be damaging to the overall wireless sensor network performance [9]. Some of the consequences include decreased network lifespan, disruption of communication channels, higher transmission loads on boundary nodes, and performance degradation [10]. Failure of a node during data transmission may result in loss of data or delayed propagation time.

Hence, detection of coverage holes plays a crucial role in enhancing the coverage rate [11]. This may lead to holes in the region of concern, as illustrated in Figure 1. Also, nodes can drift away from their allocated positions because of unpredictable environmental conditions. Coverage holes decrease the lifetime and bandwidth of wireless sensor networks. The entire performance of these networks can be impacted by coverage holes [9]. These gaps can result in reduced network longevity, interference in communication channels, increased transmission loads on boundary nodes, and performance degradation [10]. Failure of a node during data transmission may result in loss of data or propagation time delay. Therefore, identification of coverage gaps is required to improve the coverage rate [11]. Since energy consumption increases tremendously in the case of high node density, such excessive communication load can decrease the lifespan of wireless sensor networks. It is also difficult to fix the sensor nodes manually when placed randomly in inaccessible regions like dense forests or disaster areas [14]. Besides, as the network expands, it tends to cluster unevenly, which can significantly shorten its lifespan. This grouping raises sensor node energy waste, increases the total energy consumption, and lowers network connectivity. When gaps in coverage are caused by damaged sensor nodes, coverage efficiency is compromised. There must be a protocol set or remote-control methods used to restore network operation rapidly.

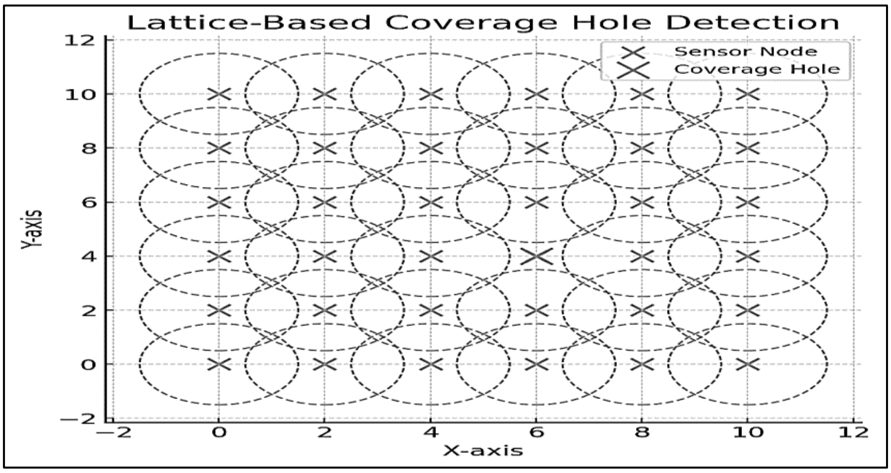

Figure 1. Coverage Holes in Forest Fire Detection Application

The article is organized as follows: An overview of the literature on hole detection techniques is given in Section 2. The problem of identifying coverage holes is covered in Section 3. Section 4 presents extensive simulations that provide a detailed explanation of the proposed methodology and validate the proposed conclusions. Lastly, the research is concluded in Section 5.

2. Related Work

Figure 2 illustrates the results of a comprehensive literature review on coverage hole detection algorithms based on several coverage categories, including area, point, and barrier coverage. Area coverage in an observation field refers to the whole area or territory that has been detected and assessed. Using current information gathered from the targeted area, point coverage will encompass the area of interest. Intruders stepping over the obstacle are identified using barrier coverage. Depending on the characteristics of the algorithmic structure, the techniques for identifying coverage holes are potentially centralized Amgoth et al. [5] or distributed [15]. The coverage hole was fixed using a coverage hole detection and repair algorithm (CHDR) introduced by Verma et al. [22]. To enable the repair of the hole, the network's dormant elements were turned into clustering nodes by first calculating the border node of the resulting polygon region. When utilizing the CHDR method instead of other computations, the accuracy rate is increased by 5%. Additionally, it adds 40% more to the network lifespan. For accurate border and hole detection, Khedr et al. [10] proposed a distributed technique that relies on boundary determination using a Connected Independent Set (BDCIS) technique. The BDCIS technique is proposed, in which the nodes collect connection information from their one-hop neighbors and create unique sets of data. This method prevents incorrect boundaries from being detected. Compared to other methods currently in use, the accuracy rate is low despite the significant energy usage. Both pro-active and reactive techniques are offered for coverage hole healing. The message's overhead will be O (n). This method increases the coverage time and longevity of networks by up to 90%. To identify and compute the holes, Robinson et al [24] proposed FL-TD (Fuzzy Logic-based Topology Detection) , a coverage hole detection method that leverages fuzzy logic to assess the coverage state of a wireless sensor network. It evaluates parameters such as node density, connectivity, and redundancy using fuzzy inference rules, which allows it to handle uncertainty and imprecision effectively. FL-TD uses fuzzy logic to detect coverage holes by analyzing node density and connectivity, achieving around 91% detection accuracy in uncertain environments. Gou et al [23] proposed DHD-MEPO (Distributed Hole Detection using Modified Energy Potential Optimization), a decentralized technique designed to detect coverage holes by utilizing an energy-aware potential field approach. In this method, nodes compute and exchange energy potential values, and inconsistencies in these values can indicate the presence of coverage gaps. DHD-MEPO applies energy potential optimization in a distributed manner for scalable hole detection, reaching approximately 88% accuracy in dynamic WSNs.

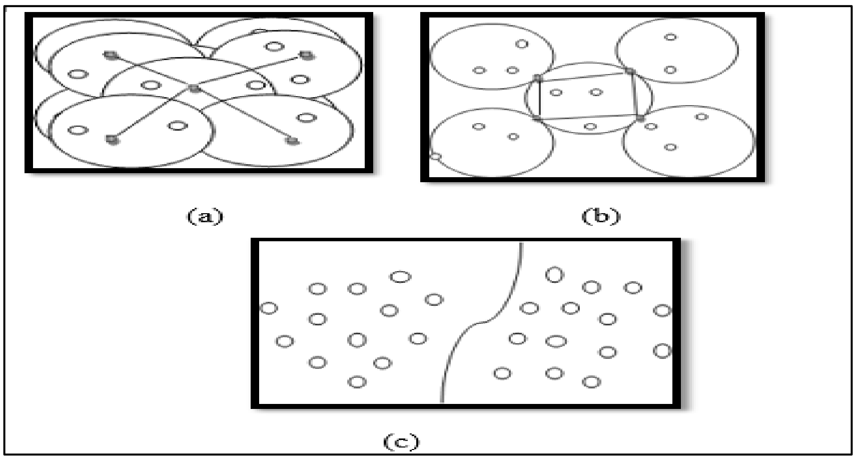

Figure 2. Coverage Types Include: a) Area, b) Point, and c) Barrier.

The review of the literature demonstrates that the centralized strategy has a single point of failure issue [11]. Dense deployment is required for wireless sensor networks because of dynamic topology and environmental conditions. High messaging volume [2], excessive energy usage, prolonged decision-making time, and coarse border selection result in multiple inside nodes being mistakenly identified as boundary nodes, which are the main drawbacks of centralized algorithms [16]. As shown in Table 1, distributed techniques are relatively costly but have great scalability in hole identification and recovery as compared to centralized techniques [17] [18]. Whenever the node density rises, centralized algorithms produce accurate outputs but incur a transmission cost.

Table 1. Comparison of Existing Coverage Hole Detection Approaches

Algorithms	**Coverage Rate**	**Network Dynamic**	**Computational Model**	**Scalability**	**Type**	**Complexity**
CHDR[1]	80%	Static	Distributed	High	Statistical	O(n)
BDCIS[2]	93.5%	Static	Distributed	High	Statistical	O(n3)
Collaborative[3]	90%	Mobile	Distributed	High	Statistical	O(n)
NOVEL[4]	93%	Static	Centralized	Low	Topological	O(n3)
Graph based[5]	95.5%	Static	Centralized	Low	Topological	O(bn)
DVOC[6]	96%	Static	Distributed	High	Statistical	O($k2n$ log n)

The three main categories of coverage hole detection techniques now in use are connectivity-based, range-based, and location-based techniques. However, they frequently need accurate geographic information, which might be difficult to obtain in some circumstances. Due to the unreliability of data availability, connectivity-based approaches are crucial. Among these methods, homology theory-based algorithms stand out because they evaluate a system's topological properties using logarithmic tools. A chord-based approach was proposed by Lee et al. [19] to efficiently detect coverage holes. By restricting the coverage zone overlay for detection purposes and utilizing the fewest possible sensor nodes, the CBHC approach guarantees complete network coverage. The outcomes of the simulation demonstrate how well the suggested techniques detect and close coverage holes. To solve coverage limitations, Saipulla et al. [20] presented a unique technique based on reinforcement learning and game theory. Due to the intrinsic characteristics of sensor nodes, computational geometry techniques are shown to be more reliable in locating holes when the number of nodes increases with minimal communication latency and optimum energy consumption [21] [22]. This depends on a thorough examination of the literature. To ensure that every target in the region of interest is covered by every group, the sensors might be arranged in discontinuous groups. Point coverage is the method that forms the basis of the suggested approach. The proposed lattice-based coverage hole detection framework in Figure 3 looks for the coverage hole that uses the least amount of energy while accounting for extra nodes in the hole location. The nodes cooperate to determine the location of the network's coverage hole. The framework for lattice-based coverage hole identification includes the following set of functions.

1. Energy-efficient hole identification with a lattice-based method that consists of
   - Network segmentation via lattice construction.
   - Hole boundary finding to pinpoint the precise position of malfunctioning nodes.
   - The identification of holes for node failure assessment.
2. Using the hole recovery algorithm for hole-repairing

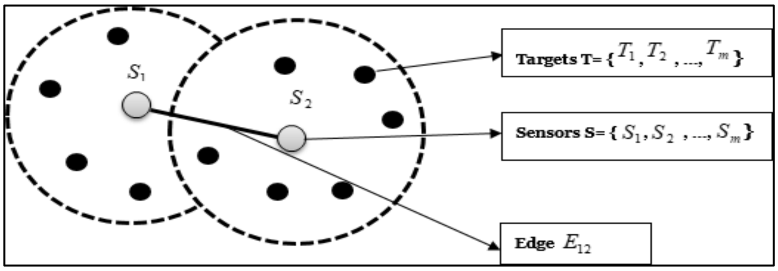

Figure 3. Communication Graph

3. Proposed Work

WSN is represented by a communication graph, $G(S, E)$, as seen in Figure 3. Let T stand for the target to be monitored, such that $T = \{T_1, T_2, \dots, T_m\}$, as m denotes the total number of targets, and let S stand for the group of sensor nodes, such that $S = \{S_1, S_2, \dots, S_n\}$

n represents the number of sensor nodes. Each sensor node is equipped with sensing, communication, and processing capabilities. Nodes have a fixed sensing radius R_s, within which they can detect events or monitor the environment. The communication range R_C is assumed to be equal to or larger than the sensing radius to enable node coordination. Each node has limited energy, which decreases over time based on sensing and communication activities. Nodes can enter active or sleep modes to conserve energy, impacting coverage dynamically.

E_{ij} indicates the communication linkages between the targets. If two nodes can interact with one another, a connection is formed. Each $S_i \in S$, where $T \subset S$, can cover a portion of the targets in set T, which is defined as $T \subset S$. Depending on the coverage attribute of the set S, coverage can be described as complete, partial, or nonexistent.

- If $Covered\ T(S_i) = n$ then the sensors are fully covered and no coverage hole exists.
- If $Covered\ T(S_i) \leq n$, then the sensors are partially covered and a coverage hole exists.

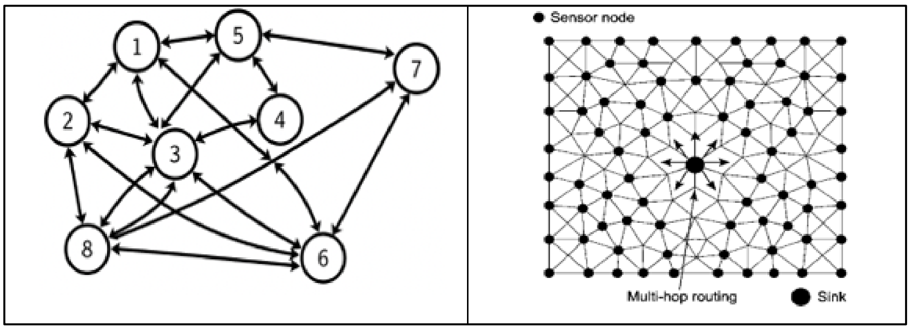

Figure 4. Representation of Coverage Graph and Network Model

Consider the coverage graph and network model shown in Figure 4. The following represents the connectedness between nodes i and j: (β_{ij}) A coverage matrix can be constructed to assist in the identification of holes present in the connectivity information of the deployed network. Whenever the coverage matrix represents 0, it indicates there is no exchange of information due to a coverage hole. When the coverage matrix represents 1, there is connectivity with no coverage hole. When an element of the coverage matrix becomes 1, it indicates that the corresponding cell is covered by at least one active sensor node, meaning the area is effectively monitored. The value 1 here denotes a binary indicator (covered = 1, uncovered = 0), and is not related to a unity or identity matrix. The coverage matrix, therefore, is a binary matrix used to reflect coverage status across the monitored region. Rows and columns of the coverage matrix represent spatial grid indices (i,j). A matrix entry becomes 1 when the cell is covered by at least one node based on its sensing range.

Therefore, the coverage equation is as follows.

$$\beta(s,t) = \begin{cases} 1, if\ E \geq n \\ 0, Otherwise \end{cases} \tag{1}$$

where E is the energy of the node, n represents the target node among the sensor nodes to which a sensor node transmits data if it is covered and T(n) is the threshold value.

3.1 Lattice based Coverage Hole Detection Algorithm

Lattice-based coverage hole identification method is shown in Figure 5, which finds coverage gaps in any monitoring zone, regardless of its size or shape [30]. It is assumed in this suggested hole detection technique that once the installation operation is completed, every sensor has position information. Every node goes through the neighbor discovery process to update its position and availability before the hole identification algorithm is run, as expressed below.

3.1.1 Grid Partitioning

The entire area is divided into uniform grid cells of 25 × 25 m². The grid size is independent of node distribution and is selected based on average sensing range to balance resolution and computational cost. In constrained spaces, edge cells are padded or resized slightly to fit the space uniformly. The lattice graph, represented by the notation, $G = (V, E)$, is given by the following eqn (2), where V is the set of vertices that represent grid cells and E is the set of edges that indicate connections between adjacent cells.

$$V = \{(i,j\}|1 \leq i \leq m1, 1 \leq j \leq n1\}\ and\ E = \{(u,v)|\ u,v \in V, and\ |u-v| = 1\} \tag{2}$$

In this case, m1 and n1 stand for the grid's row and column counts, respectively.

3.1.2 Node Activation

Each cell's node count, N_{ij}, is determined by variables such as coverage requirements and node density. It represents the minimum number of sensor nodes required to ensure acceptable coverage for cell (i,j). It is determined based on the application coverage threshold, node sensing radius and data redundancy needs.

3.1.3 Coverage Evaluation Metrics

Let C_{ij} represent the coverage metric for cell (i,j). Coverage can be quantified based on the number of nodes, signal strength, or sensed data quality.

3.1.4 Hole Detection Algorithm

A hole detection algorithm analyzes coverage metrics within each cell and its neighboring cells to identify coverage deficiencies.

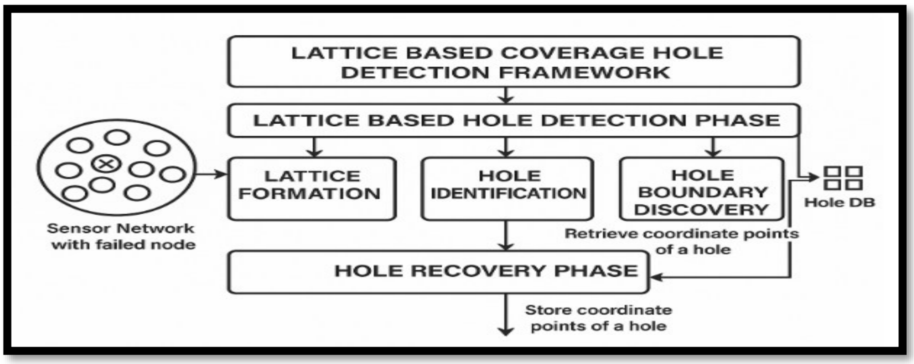

Figure 5. Lattice based Coverage Hole Detection Framework

3.2 Lattice Formation

As seen in Figure 6, the sensor nodes are positioned at random throughout the ROI. Each sensor node will initially have the same characteristics and energy output. Every sensor node uses the information it has collected from its neighbors to construct a lattice. Every point in a Voronoi polygon is closer to every sensor node than it is to any other point. To create a Voronoi polygon, sensor nodes must first compute their own and their neighbors' bisectors.

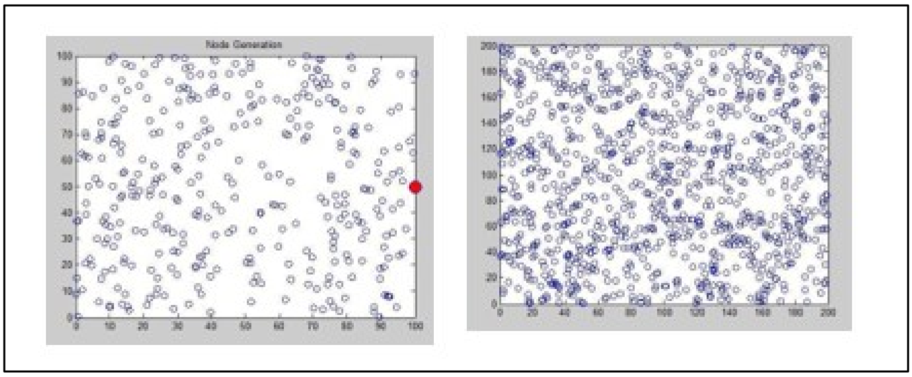

Figure 6. Installation of Sensor Nodes

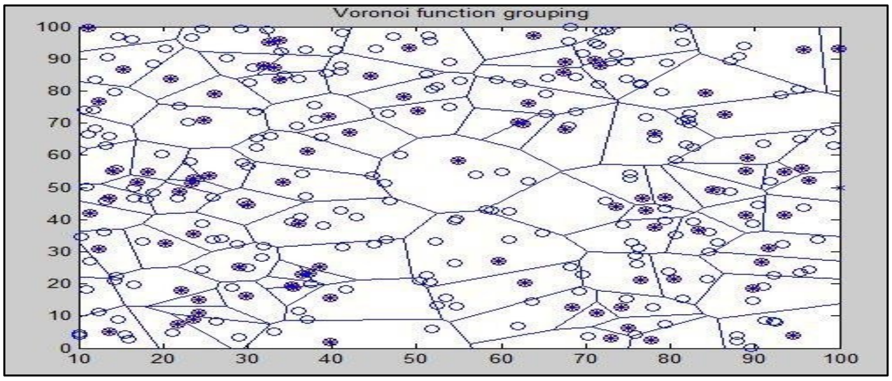

Figure 7. Lattice Formation

Every sensor-representing point has a Voronoi polygon encircling it. The whole region of concern is covered by this Voronoi polygon. Sensor nodes are clustered together to create clusters in large-scale networks. The delimited plane is divided into lattices, as shown in Figure 7, to provide a variety of sensor nodes. Each node's cell contains precisely one sensor node. Additionally, according to the Voronoi diagram's partitioning property, the distance between every target in set T in a given partition and its sensor S is less than the distance between the object being targeted and nodes that are adjacent in the neighboring partition. The following equation can be derived from the definition of the 2D Voronoi diagram partition division:

$$S^i = \{\cap_{i=1}^{n} / V(S^i, T) \ < \ V(S^i, T), j = 1,2, \dots, n-1\} \tag{3}$$

$$V(S^j, T) = \sqrt{(x^2 - x^1)^2 \ + (y^2 - y^1)^2} \tag{4}$$

where $T(x, y)$ denotes the coordinates of any target inside the monitoring region. The Euclidean distance between nodes or targets is given in eqn. (4). As clusters are formed, every node chooses whether or not to become the cluster leader for the present phase. The number of times the node has functioned as a cluster head and the network's intended cluster head percentage are taken into consideration while making this decision. The process of making this decision involves the node n picking a number at random between 0 and 1. Nodes that have held the position of cluster head in the past are not eligible to hold it again for P rounds, where P is the number of cluster heads that must be present. After that, there is a 1/P probability for any node to take over as the cluster leader.

In the probabilistic sensing paradigm, the chance that a sensor node in S covers the subset of target $T(x, y)$ at any place on Disk1 is as follows.

$$\beta(s,t) = P^i e^{-£_i}\left|S_i - S_j\right| \tag{5}$$

where Pi refers to probability either 0 or 1, £ is a positive constant, and $\left|S^i - S^j\right|$ represents the Euclidean distance between any two nodes that contain sensors in WSN. In addition, the sensor node has unidirectional sensing capabilities and is a declining and derivative value. Let the threshold value of the energy E be,

$$T(n) = prob_i/(1 - prob_i \ * \ (P * mod(1/prob_i)) \tag{6}$$

where probi is the probability of node sensing target, and P is the no. of rounds. The total coverage area is given by eqn. (7) as follows.

$$Total\ Coverage\ Area = \sum_{x=1}^{m}\sum_{y=1}^{n} prob(t,s) \tag{7}$$

WSN deployment area is represented by the set $G = (V, E)$, where E is the set of edges connecting neighboring lattice points, and V is the set of lattice points. C(v) is the coverage area of lattice point v, which is the area that sensor node v can sense. The region within G that is not completely covered by the union of coverage areas C(v) for all lattice points v is called a coverage hole. The coverage area C(v) related to lattice point v may be found using a function named EstimateCoverage(v).

DetectHoles(G) is the definition of the function that locates coverage holes inside the deployment region. Logic and Properness1.Completeness: The algorithm is complete if it detects all coverage holes present in the deployment region.

$$\forall\ coverage\ hole\ H \subset G, \exists v \in V: H \subseteq \neg C(v)\ \Rightarrow\ DetectHoles(G)\ = \{v'\} \tag{8}$$

$$\forall\ coverage\ hole\ H \subset G, \exists v \in V: H \subseteq \neg C(v)\ \Rightarrow\ DetectHoles(G) = \{H\} \tag{9}$$

$$\forall\ H' \in DetectHoles(G), \exists v' \in V: H' \subseteq \neg C(v') \tag{10}$$

3.3 Hole Identification and Recovery

The lattice-assisted hole identification algorithm functions differently for each lattice, as seen in Fig. 6. The inactive sensor node's location inside each lattice is ascertained by choosing the sensor nodes that are nearest to the lattice coordinates to compute the hole areas. Consider the failed node F which has the location information as (x^f, y^f). Let us use Lagrange multipliers to determine the closest point in the region of interest within a circular communication range where ($R^C < R^S$). This can be denoted as, $(a - x^f)^2 + (b - y^f)^2$ subjected to the constraint $a^2 + b^2 - 1 = 0$. According to the definition of Lagrange multipliers,

$$f^x = \lambda g^x \tag{11}$$

Determining the partial derivative concerning λ gives,

$$a(1-\lambda) = x^f \tag{12}$$

$$b(1-\lambda) = y^f \tag{13}$$

$$a^2(1-\lambda)^2 + b^2(1-\lambda)^2 = (1-\lambda)^2 \tag{14}$$

$${x^f}^2 + {y^f}^2 = (1-\lambda)^2 \tag{15}$$

$$(1-\lambda) = \sqrt{{x^f}^2 + {y^f}^2} \tag{16}$$

$$\therefore\ \lambda = \sqrt{{x^f}^2 + {y^f}^2} + 1 \tag{17}$$

Lagrange multipliers indicate that $\lambda = \sqrt{{x^f}^2 + {y^f}^2} + 1$ is the nearest point in the region of interest where there is no chance that a coverage hole would exist. Node density is used to calculate the sensitivity analysis. Node Density (ρ) is the measure of how many sensor nodes each grid cell has in relation to coverage quality. The average coverage metric (C_{avg}) is given in the following equation.

$$\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} Ci\,j \tag{18}$$

where m and n denote the grid's row and column counts, and C_{ij} is the coverage metric for *cell (i,j)*. Several parameters are taken into account while simulating situations for coverage hole identification to guarantee the repeatability of results.

3.4 Complexity Analysis of Lattice-based Coverage Hole Detection

The computational complexity of the proposed lattice-based coverage hole detection algorithm is primarily influenced by two operations: (1) determining the coverage matrix C_{ij}

by calculating the number of active nodes N_{ij} in each grid cell, and (2) detecting coverage holes by scanning the matrix for uncovered regions.

Let us assume N is the total number of sensor nodes; M × M: total number of grid cells (lattice points) and R_s: sensing radius of each sensor node

- **Coverage Matrix Construction (Computing N_{ij})**

Each sensor node potentially contributes coverage to a set of nearby grid cells within its sensing radius R_s. For a uniform lattice, the number of cells within a node's coverage area is approximately proportional to $\pi r^2 / d^2$, where d is the grid spacing. Assuming k is the average number of cells covered per node, the total time to update all relevant N_{ij} values is:

$$O(N \cdot k) \approx O(N)$$

This assumes k is constant for fixed r_s and d, which holds in a lattice-based deployment.

- **Coverage Hole Detection**

Once the binary coverage matrix C_{ij} is constructed (where C_{ij}= 1 if $N_{ij} \geq 1$, else 0), the algorithm performs a scan across the M × M matrix to identify contiguous regions of 0s (uncovered areas). Using a flood-fill or connected-component labeling approach, this step has a complexity of $O(M^2)$.

- **Overall Complexity**

Combining the two main steps, the overall time complexity of the algorithm becomes:

$$O(N + M^2)$$

This is efficient for practical network sizes, especially when $M^2 \approx N$, which is typical in structured grid-based deployments.

- **Space Complexity**

The algorithm requires storage for the coverage matrix C_{ij} , which is $O(M^2)$, and possibly for a neighborhood matrix or visited flag array during hole detection.

ISSN: 2582-4104

4. Results and Discussions

Table 2. Simulation Parameters

Parameter	Value
Number of sensor nodes	1000
Deployment area	500 × 500 m^2
Communication range	30 m
Initial energy per node	2 Joules
Sensing range	15 m
Packet size	512 Bytes
Simulation time	1000 rounds
Energy model	First-order radio
Grid cell size	25 × 25 m^2

The size of the sensing region and the necessary resolution dictate how many rows and columns are included in the grid created by splitting the region. The time complexity of this process is typically O(1) since it is dependent on predetermined parameters. Coverage metrics are evaluated by aggregating data from sensor nodes within each grid cell. O(k), where k is the number of nodes in a cell, and the chosen measure determine the complexity. Finding coverage gaps begins with examining coverage metrics both within and between each grid cell and its neighbors.

4.1 Simulation Findings

Simulation results indicate that the node death rate of the suggested algorithm is slower than that of the existing protocols (Table 3). The ideal approach to maintain the network's maximum activity is to adaptively change the nodes' status. Consequently, the suggested method's node loss rate is extremely sensitive.

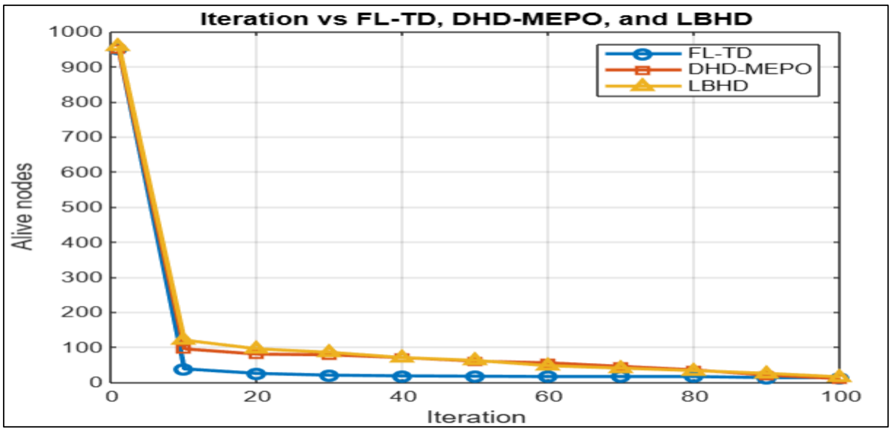

Figure 8. Number of Alive Nodes

Table 3. Number of Alive Nodes

Iteration	FL-TD	DHD-MEPO	LBHD
1	950	950	960
10	38	95	120
20	25	80	95
30	20	78	85
40	18	70	70
50	17	60	62
60	16	55	47

The number of alive nodes for each of the three methods FL-TD, DHD-MEPO, and LBHD across several iterations is displayed in Figure 8. All approaches begin with roughly the same number of live nodes between 950 and 960. Energy depletion or node failures cause the number of alive nodes to decline over the course of the iterations in all methods. Over the majority of iterations, LBHD continuously keeps the most alive nodes, demonstrating improved energy management and longer network lifespan. With more nodes kept alive than FL-TD but fewer than LBHD, DHD-MEPO exhibits a moderate level of performance. With the steepest decline and the fewest nodes still alive by iteration 10, FL-TD appears to be the least effective at maintaining node life.

4.2 Overhead in Control Packets Vs Node Count

The control packet overhead of various methods is contrasted in Figure 9. The overhead in control packets, expressed in kilobytes (kB), for three distinct approaches is shown in Table 4. DHD-MEPO, FL-TD, and LBHD throughout several iterations. A common baseline is indicated by the fact that all three approaches begin with the same overhead of 5000 kB. Because more control packet exchanges are needed to keep the network running, the overhead for all methods rises as iterations go on. LBHD consistently shows the lowest overhead out of the three indicates that it is the most effective at controlling traffic. DHD-MEPO has the highest overhead at every iteration, whereas FL-TD displays moderate overhead values. More control communication is implied by the higher overhead in DHD-MEPO, which may be related to improved network functionality or maintenance.

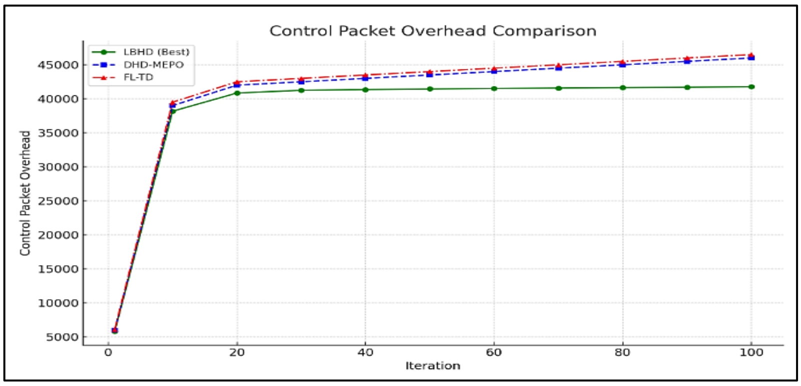

Figure 9. Overhead in Control Packets Vs Node Count

Table 4. Overhead in Control Packets

Iteration	LBHD (kB)	FL-TD (kB)	DHD-MEPO (kB)
1	5000	5832	5000
10	38500	38174	39700
20	40500	40846	41700
30	41000	41249	42800
40	41200	41354	43300

50	41300	41446	43800
60	41400	41519	44300

In WSNs, the lattice-based method improves energy efficiency using event-driven operation, organized deployment, dynamic adaptability, localized information interchange, focused data gathering, and effective node activation. Compared to certain other existing systems, particularly those that lack a systematic approach to spatial organization and optimization, the lattice's structured and ordered nature makes intelligent energy management possible, resulting in a more energy-efficient solution.

5. Conclusion

Lattice-based coverage hole detection is a promising distributed method for effectively detecting coverage gaps in static homogeneous wireless sensor networks. Compared to DHD-MEPO, the LBHD approach has shown a 15% increase in node survivability and a 17% improvement in energy efficiency. However, while these improvements greatly reduce control overhead and computational complexity during hole detection and recovery, they also result in a slight decrease in responsiveness to dynamic coverage holes and a 5% decrease in packet delivery ratio. The difficulties increase in complexity as network dynamics change, especially in mobile environments. In order to minimize energy overhead and maximize coverage retention, future research should focus on the real-time detection and responsive repair of holes in mobile sensor networks. Distributed algorithms often have higher computational complexity and energy costs than centralized solutions, despite offering scalability and fault tolerance. Thus, incorporating robust hole repair mechanisms and optimizing distributed algorithms for energy efficiency are crucial areas for further research..

References

[1] Agrawal, Reeya, Neetu Faujdar, Carlos Andres Tavera Romero, Oshin Sharma, Ghadia Muttashar Abdulsahib, Osama Ibrahim Khalaf, Romany F. Mansoor, and Osama A. Ghoneim. "Classification and comparison of ad hoc networks: A review." Egyptian Informatics Journal 24, no. 1 (2023): 1-25.

[2] AlSkaif, Tarek, Manel Guerrero Zapata, and Boris Bellalta. "Game theory for energy efficiency in wireless sensor networks: Latest trends." Journal of Network and Computer Applications 54 (2015): 33-61.

[3] Alam, SM Nazrul, and Zygmunt J. Haas. "Coverage and connectivity in three-dimensional networks." In Proceedings of the 12th annual international conference on Mobile computing and networking, 2006, 346-357.

[4] Ammari, Habib M. "A Spatial Approach for Physical Security Through Connected k-Barrier Coverage." In Theory and Practice of Wireless Sensor Networks: Cover, Sense, and Inform, Cham: Springer International Publishing, 2022, 655-682.

[5] Amgoth, Tarachand, and Prasanta K. Jana. "Coverage hole detection and restoration algorithm for wireless sensor networks." Peer-to-Peer Networking and Applications 10 (2017): 66-78.

[6] Chao, Chih-Min, Ming-Wei Lu, and Yu-Cheng Lin. "Energy-efficient multichannel MAC protocol design for bursty data traffic in underwater sensor networks." IEEE Journal of oceanic engineering 40, no. 2 (2014): 269-276.

[7] Feng, Xin, Xin Zhang, Jing Zhang, and Ali Ahmed Muhdhar. "A coverage hole detection and repair algorithm in wireless sensor networks." Cluster Computing 22, no. Suppl 5 (2019): 12473-12480.

[8] Guo, Jianxia. "[Retracted] Mathematical Modeling and Simulation of Wireless Sensor Network Coverage Problem." Wireless Communications and Mobile Computing 2021, no. 1 (2021): 5537004.

[9] Hua, Guan, Yan-Xiao Li, and Xiao-Mei Yan. "Research on the wireless sensor networks applied in the battlefield situation awareness system." In International Conference on Electronic Commerce, Web Application, and Communication, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, 443-449.

[10] Khedr, Ahmed M., Walid Osamy, and Ahmed Salim. "Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks." Computer Communications 124 (2018): 61-75.

[11] Khalifa, Banafsj, Zaher Al Aghbari, Ahmed M. Khedr, and Jemal H. Abawajy. "Coverage hole repair in WSNs using cascaded neighbor intervention." IEEE Sensors Journal 17, no. 21 (2017): 7209-7216.

[12] Lee, Sookyoung, Mohamed Younis, and Meejeong Lee. "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance." Ad Hoc Networks 24 (2015): 1-19.

[13] Li, Wei, and Yuwei Wu. "Tree-based coverage hole detection and healing method in wireless sensor networks." Computer networks 103 (2016): 33-43.

[14] Li, Wei, and Wei Zhang. "Coverage hole and boundary nodes detection in wireless sensor networks." Journal of network and computer applications 48 (2015): 35-43.

[15] Lin, Jieni, Junren Qin, Shanxiang Lyu, Bingwen Feng, and Jiabo Wang. "Lattice-based minimum-distortion data hiding." IEEE Communications Letters 25, no. 9 (2021): 2839-2843.

[16] Malathi, L., R. K. Gnanamurthy, and Krishnan Chandrasekaran. "Energy efficient data collection through hybrid unequal clustering for wireless sensor networks." Computers & Electrical Engineering 48 (2015): 358-370.

[17] Nandi, M. R. I. N. A. L. "Coverage problem in two and three dimensions." European Journal of Mathematics and Applications 3, no. 11 (2023).

[18] Park, Junyoung, Sunggu Lee, and Sungjoo Yoo. "Time slot assignment for convergecast in wireless sensor networks." Journal of Parallel and Distributed Computing 83 (2015): 70-82.

[19] Qiu, Chenxi, Haiying Shen, and Kang Chen. "An energy-efficient and distributed cooperation mechanism for $ k $-coverage hole detection and healing in WSNs." IEEE Transactions on Mobile Computing 17, no. 6 (2017): 1247-1259.

[20] Saipulla, Anwar, Cedric Westphal, Benyuan Liu, and Jie Wang. "Barrier coverage of line-based deployed wireless sensor networks." In IEEE INFOCOM 2009, IEEE, 2009, 127-135.

[21] Tosun, Mustafa, Umut Can Cabuk, Elif Haytaoglu, Orhan Dagdeviren, and Yusuf Ozturk. "CoRMAC: A connected random topology formation with maximal area coverage in wireless ad-hoc networks." IEEE Internet of Things Journal 10, no. 14 (2023): 12379-12392.

[22] Verma, Manoj, and Sanjay Sharma. "A greedy approach for coverage hole detection and restoration in wireless sensor networks." Wireless Personal Communications 101 (2018): 75-86.

[23] Gou, Pingzhang, Miao Guo, Baoyong Guo, and Shun Mao. "DHD-MEPO: a novel distributed coverage hole detection and repair method for three-dimensional hybrid wireless sensor networks." Electronics 12, no. 11 (2023): 2445.

[24] Robinson, Y. Harold, T. Samraj Lawrence, E. Golden Julie, and S. Vimal. "Development of fuzzy enabled coverage hole detection algorithm in wireless sensor networks." Wireless Personal Communications 119, no. 4 (2021): 3631-3649.