

DPLHD: A computational geometry based provision for coverage hole detection in critical Wireless Sensor Network Applications

Anitha Christy Angelin (✉ babyanithaphd2020@gmail.com)

Karunya Institute of technology and Science: Karunya Institute of Technology and Sciences

Salaja Silas

Karunya Institute of technology and Science: Karunya Institute of Technology and Sciences

Research Article

Keywords: Hole detection, Region of Interest, Computational Geometry, Visibility estimation, Polygon Triangulation

Posted Date: March 15th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1412609/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

DPLHD: A computational geometry based provision for coverage hole detection in critical Wireless Sensor Network Applications

Anitha Christy Angelin¹, Salaja Silas²

¹Research Scholar, Department of Computer Science and Engineering,
Karunya Institute of Technology and Sciences, India
babanithaphd2020@gmail.com

²Associate Professor, Department of Computer Science and Engineering,
Karunya Institute of Technology and Sciences, India

Abstract Wireless Sensor Networks (WSNs) have quite an exceptionally significant effect in diverse fields correlated to surveillance in which coverage plays an essential role. In specific, the coverage holes are actually instigated by both randomized distribution of the wireless sensor nodes and the failure of nodes. Recharging or repairing the battery is difficult, and thus the collaborative identification and estimation of coverage holes and the reclamation of these holes has been considered significant. When organizing the sensor nodes in a large scale WSNs, it is indeed complicated to cover the target or the region of interest (ROI). Coverage quality is compromised by the occurrence of holes in the monitored area of surveillance. This paper proposes a collaborative distributed point location based coverage hole detection methodology to spot coverage holes in an energy efficient manner. Firstly, construct a polygon using a visibility estimation approach on the basis of neighborhood information. Then a point location based hole detection algorithm is used to diagnose whether coverage hole is existing or not in the ROI. Also, the extent of the coverage hole is estimated based on computational geometry based modified partitioning approaches which not only determines the exact region of the hole but also accurately detects the point where the node fails. The accuracy of the algorithm is evaluated here based on statistical proofs. Comparing current coverage hole detection approaches, the output results of the proposed framework surpasses existing algorithms in standings of coverage rate, energy efficiency and network lifetime.

Keywords Hole detection, Region of Interest, Computational Geometry, Visibility estimation, Polygon Triangulation.

[Type here]

1 Introduction

There are enormous numbers of sensor nodes in wireless sensor (WSN) networks spread homogeneously or randomly within the surveillance zone [3, 18] and they have the ability to independently organize and form the network [3]. Since a random deployment [1] is usually used in mission critical applications of WSNs and the probability of node failures also is high, detecting coverage holes is very important. Resources like inadequate memory, battery power, processing and communication potentials for sensor node were indeed intensively restricted. The energy draining of each sensor is a key consideration for its contact spectrum and the formulation of algorithms for a sensor's uniform or random distribution, localization [19], scheduling [19], coverage [1] and other problems in WSNs. In large number of essential applications like fire detection [30], intruder detection in battleground [27], disaster relief [28], healthcare [22] and industrial surveillance [29], sensor nodes carrying essential information may die due to battery drain [15] or environmental conditions [28]. If a node fails during data transmission, it will either prompt data loss or defer the propagation delay time. Therefore coverage hole detection [39] is essential to improve coverage [23]. Contemplate the topology of a WSN given in Figure 1, where an interconnection is formed when the sensor nodes intersect which leads to overlapping of sensing areas of nodes. Here all sensor nodes are believed to have the same sensing range.

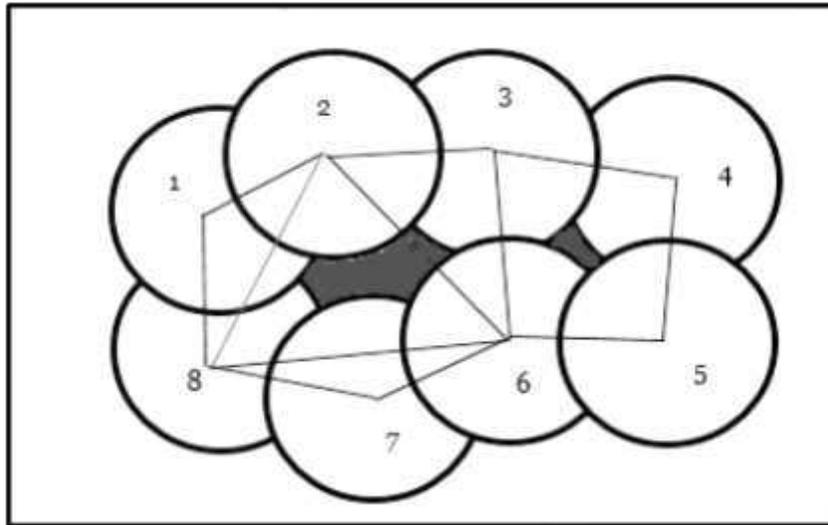


Fig 1 Coverage hole scenario

We have a closed region, as shown in this Figure 1, defined by the links between boundary nodes. The closed ROI has to be fully covered to avoid the existence of coverage hole. In fact, the polygon regions 1-2-6-7-8, 2-3-6 and 3-4-5-6 are not fully covered, while the sensor nodes cover the triangle regions 1-2-8 and 6-7-8. In this case, an effective technique is obviously needed to quickly detect coverage hole in the entire closed area based on the relationship between the sensor node disks. Therefore, to guarantee the quality of network services [5], finding and identifying coverage holes within the network is very important. In general, algorithms to recognize different coverage holes within the sensor networks can be divided into two main classes, depending on the system domains and the degree of data constraints such as probabilistic approaches and computational geometry based approaches. Compared to the probabilistic approaches, the computational geometry based approaches improve the precision of identification by finding the accurate size and shape of the coverage holes by coordinate information on the nodes and by providing better information for restoring of coverage holes [6, 7].

The paper proposes a collaborative framework based on estimates relying on the geometry of wireless sensor networks that can classify the coverage hole as a distributed and efficient point location. First, the limits of the set of nodes containing a coverage hole are established with a visibility estimation algorithm. Details of nodes positioning, the intersection of the sensors and the sensor range shall be taken into account. A Polygon triangulation approach with a one-hop neighbor of the nodes that contain the hole then defines the covering hole. This approach outperforms existing approaches in terms of energy efficiency [14].

The rest of this paper is structured as follows. The next section briefly reviews the various existing hole detection approaches. Section 3 elaborates on the proposed coverage hole detection framework, and Section 4 is about the experimental setup. Section 5 provides a detailed performance evaluation results comparing the proposed work with the existing algorithms. Section 6 gives an overview on applications and future work. Lastly, we conclude this paper in Section 7.

2 Background

A broad literature survey is carried out on the coverage hole detection schemes based on different classifications of coverage hole detection approaches such as the probabilistic approaches and the computational geometry approaches. The first is probabilistic node density estimation approaches required for good coverage [17], but these findings do not provide enough proof of hole detection strategy [16, 19]. The probabilistic coverage algorithm gives a binary decision, a 1 or 0, based on whether the region is covered with the required detection probability or not. This is

[Type here]

accomplished by distributed decision making at each sensor node. The exact location based data is less important in the probabilistic approach [32] but it is necessary to increase node density. In the meantime, the coverage issues are impossible to detect exactly. The latter uses techniques in computational geometry to classify coverage holes. To find the holes, the computational geometry methods use the positioning details of the nodes or the corresponding distance between the nodes. The tree based coverage hole detection is presented in [9] to locate the coverage hole and to identify coverage hole position, shape and size. Nevertheless, the relative position of the adjacent nodes is considered to identify the coverage holes. A boundary node structure algorithm for the sensor network region is implemented in [33], but nodes with higher densities are needed to presume the location of coverage holes. Suppose that the unit disk graph model determines the communication between the nodes in [34] and the linear-time algorithm is used to locate the holes. Nevertheless, the algorithm can not distinguish between two identical holes. In [35], the issue of coverage under the poisson deployment scheme with a 2-D random walking mobility model for a heterogeneous wireless sensor network[8] is addressed. The coverage levels will be increased with the adoption of mobility. Even then, due to false detection, the detection precision is diminished. In[4], the author proposed the algorithm by using the empty circle property to determine the exact boundaries of the nodes. This algorithm is useful in removing the false boundary nodes. False boundary nodes are removed by clustering different holes. Compared to other algorithms, this proposed algorithm will recognise the boundary nodes more accurately.

A distributed hole detection and tree-dissect based hole-healing algorithm (THH) is recommended for coverage hole detection in [40]. In the tree, IECs that are related to just one extra IEC are called boundary IECs and are contained in a list containing their sizes. The largest IEC in the list is omitted and is set as the root to create a new sub-tree. The IECs adjacent to the source are excluded from the list and added to the sub-tree and are set as boundary IECs. If the radius of the IEC in the list is less than that of the equivalent corresponding boundary IEC, the IEC is excluded from the list. In addition, the IEC is applied to the sub-tree and is set as the boundary IEC of the sub-tree before the list is empty. The question of false border identification is a concern in this strategy.

In [10], the author has proposed the distributed solution to detect boundaries and holes accurately using Boundary Detection based on Connected Independent Set (BDCIS) algorithm. BDCIS algorithm is suggested in which its one hop neighbor's connectivity information [20] was gathered by the nodes and distinct sets of data were formed with that data. This algorithm prevents the detection of false boundaries. Though energy consumption is high, the precision rate is low compared to other existing techniques. [30] An improved hole identification and healing system is recommended that could be used in WSNs for replacement of obsolete nodes and holes[30]. An efficient

computational geometric analytical approach is used to find the wireless sensor network coverage hole where the communication range and the node sensing range are the same[27].However the algorithm suffers from reduction in network lifetime compared with other algorithms.

Table I Evaluation of current coverage hole detection methods

Algorithm	Network Type	Drawback	Model	Type	Complexity
Tree based detection [9]	Static	Cannot detect holes that are very close to each other.	Centralized	geometrical	$O(n)$
Poisson deployment scheme [35]	Mobile	Coverage distance is overlooked and is detected wrongly.	Centralized	probablistic	$O(n^3)$
Empty circle property[4]	Static	Suffers high computational overhead.	Centralized	probablistic	$O(n)$
THH[40]	Distributed	False boundary detection rate is high.	Distributed	geometrical	$O(n^3)$
Graph based[21]	Static	High computational overhead	Centralized	geometrical	$O(bn)$
Novel hole detection method[38]	Static	It cannot detect the boundary holes	Distributed	probablistic	$O(h^3n \log n)$
BDCIS[10]	Static	Reduction in network life time	Distributed	probablistic	$O(n \log n)$
Modified hole detection and healing method[30]	Mobile	Suffers from reduction in network lifetime compared with other algorithms.	Distributed	probablistic	$O(n)$
DCHD [2]	Mobile	High	Distributed	geometrical	$O(n \log n)$

[Type here]

In [38] Kanno et al. suggested a novel approach for the quantitative measurement of coverage holes along with their position in a coordinate-free sensor network. In addition, hole-boundary is detected by processing information embedded in a non-planar communication graph. From this non-planar graph, a planar graph is obtained and then by using a 'partition network' algorithm, it is further subdivided into sub graphs. By assuming that the outermost boundary of the sensor network is known, a path passing through the center of the network that separates it into two without losing holes is found that takes us from one boundary node to another. The disadvantage of the approach is that it cannot detect the boundary holes. In [2] the authors suggested the Distributed Coverage Hole Detection (DCHD) algorithm for segregating the bounded or unbounded coverage holes in a WSN. Before the hole detection, the set of boundary nodes containing the hole must be identified. The algorithm includes: the positioning of the nodes, the overlapping of sensing coverage and the range of non-overlapping sensing regions. The drawback of this approach is large computational overhead.

In [37], Z. Kang et al. proposed a distributed algorithm which is coordinate-free and connectivity-based. The algorithm is based on BCPs and can run on a single node by verifying the BCPs from neighbors. The connecting consecutive BCPs can form the BLs. A node in the network is identified and its neighbors are found and sorted clockwise and a list of intersection points are created based on either node & its sorted sensing neighbors or node & the boundary of the monitoring region and finally each intersection point is further verified whether it is a BCP or non-BCP and continue the process for all nodes. The algorithm suffers from high computational complexity.

The Wireless Sensor Network employs the Unit Disk Graph UDG (V), where V is a vertex, and there is a boundary between two nodes if and only if their Euclidean distance is at most one. In addition, each node has a Global Position System (GPS) which provides information on the location of the node itself. If the GPS is unavailable, the distance between the nearest nodes can be determined on the basis of the input signal power, which is an intimidating task in probabilistic approaches. Relative coordinates of neighboring nodes can be acquired in geometric approaches by sharing information between neighbors [26]. With the position information, we can apply computational geometry techniques [4]. Here the evaluation parameters are: boundary discovery time [4], energy efficiency [4], detection accuracy [41] and communication overhead [40] based on the effects of number of nodes[2], number of holes[4], size of holes[41], number of message exchanges[40], average node degree[26,40] etc. In order to provide a robust infrastructure, protocols intended for these complex networks should be distributed. The main drawback of existing

algorithms are: the large amount of exchanged messages[24], energy consumption and selection time[31], and selection of coarse boundaries in which several internal nodes are wrongly recognized as boundary nodes[28]. As referred in Table 1, the geometric methods using delaunay and voronoi have high computational complexity[2,41] compared to the existing method due to its tree structure. When the node density increases, node failure occurs often and the search process becomes more complicated failing to achieve the target coverage. The proposed technique is built based on the sensors which can be distributed into disjoint groups such that every group fully covers all the targets in the region of interest which is called as point coverage. The proposed distributed point location based coverage hole detection framework in Figure II aims at detecting the location of the coverage hole using minimum energy consumption considering redundant nodes in the hole region. The nodes collaborate to determine the location of coverage hole in the network. The proposed Distributed Point Location based Hole Detection (DPLHD) framework consists of following approaches which includes:

- Polygon construction using visibility estimation approach
- Triangulation using Polygon triangulation approach
- Coverage hole detection using Point location based coverage hole identification algorithm.

3 Proposed framework

3.1 System model

Let the sensor nodes in the network are positioned in a 2-D plane. The primary nodes in the target area are evenly segregated and the frontier nodes beyond the external boundaries of the target area are distributed similarly. The node may be identified as the internal node or not depending on the initial environment and correct positional knowledge is unaware of each node. Let us consider the following conditions:

- Let R_{sense} be the radius of the sensing range and R_{comm} be the radius of the communication range of a sensor node such that $R_{comm} = 2R_{sense}$.
- A binary sensor model is implemented in the network for each node and every node has its own exclusive ID.

The Point location based coverage hole detection framework in Figure II consists of a Geometric visualization component comprising of modules for visibility estimation for identifying the neighbor nodes and a minimum cost triangulation algorithm for identifying the boundary nodes[10, 11] of the polygon. The Coverage hole detection component consists of a point location based hole detection algorithm which determines the location of the failed node causing the coverage hole. The hole data evaluated by the hole detection algorithm is stored in the hole information database for future use. Take into account a wireless sensor network where the

number of sensors is randomly deployed over the rectangular sensor area R . Presumably, certain parts of the network are deployed with a large overlap in the sensing spectral range, and some other parts of the network are slightly deployed, which is noticeable because the nodes are randomly

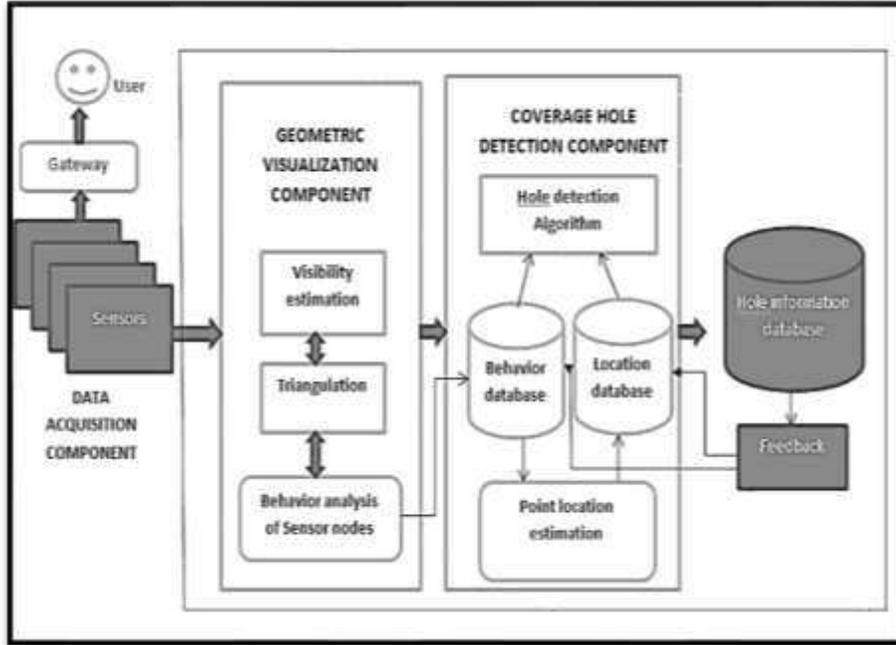


Fig II Proposed point location based coverage hole detection framework

distributed. When node deployment is huge, the coverage intersects, while the sparse deployment leads to coverage holes [11]; and, irrespective of the existence of the holes, the entire network is well integrated. Coverage holes would also be generated due to the death of exhausted nodes.

3.2 Visibility estimation algorithm

The visibility estimation algorithm in Figure III is used to construct a polygon from a deployed set of sensor nodes where every edge of the polygon determines the connected link between the neighboring nodes. This algorithm operates the points in the order in which they appear. The visibility estimation approach begins from $m=0$ and a point E_0 known to be on the polygon. For example, take the leftmost point, and choose the point E_{m+1} such that all points are to the right of the line E_{m+1} . Then

update the polygon point set and repeat it until we reach the left end. In $O(n^1)$ time, this point can be defined by associating polar angles from all points to E_m in the polar coordinate center. The inner loop monitors each point in the set S, and the outer loop continues with each point in the polygon. Therefore, the gross run time is $O(nh)$.

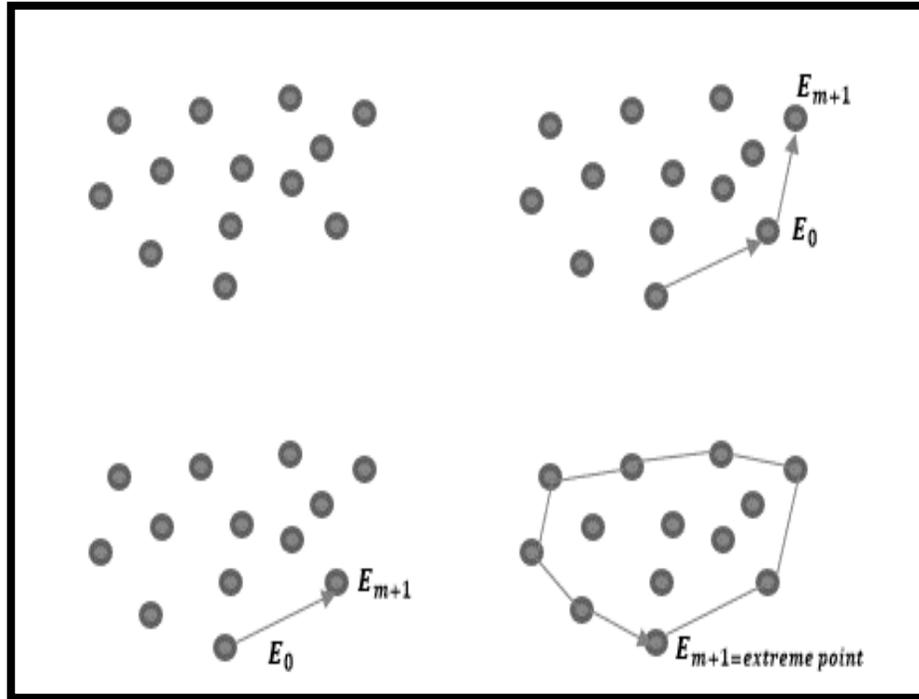


Fig III Visibility estimation algorithm

Algorithm
Input: Set of points $S \leftarrow \{ \}$ enclosing the sensor nodes.
Output: Polygon P with boundary points E_1, E_2, \dots, E_n
 Let S be the set of deployed sensor nodes in the region.
 Let $m := 0$
 Repeat
 $E[m] := \text{position}$
 $\text{ending_position} := S[0]$
 for n from 0 to $|S|$ do
 If ($\text{ending_position} == \text{position}$) then
 $\text{ending_position} := S[n]$
 $m := m + 1$

[Type here]

position: = ending_position
 Until ending_position:= E [0]

3.3 Modified minimum cost triangulation algorithm

The constructed polygon is triangulated using a Modified Partition algorithm as in Figure IV. A polygon with n+1 points may be shown in a counter-clockwise series of vertices, i.e., $P = \langle \beta_1, \beta_2, \dots, \beta_n \rangle$ has n+1 sides, $\langle \beta_0, \beta_1 \rangle, \langle \beta_1, \beta_2 \rangle, \dots, \langle \beta_{n-1}, \beta_n \rangle, \langle \beta_n, \beta_0 \rangle$. Bringing diagonals from opposite vertices becomes a convex polygon's

triangulation such that the diagonal elements never intersect. Let $\psi(p, q)$ be the cost of an ideal triangulation of the polygon $\langle \beta_{p-1}, \beta_{p+1}, \dots, \beta_q \rangle$, where

$$\psi(p, q) = \text{minimum cost for } p-1 \text{ to } q, \psi(p, q) = 0, \text{ if } p=q \text{ otherwise,}$$

$$\min_{p \leq r \leq q-1} \{ \psi(p, r) + \psi(r+1, q) + \text{cost}(\langle \beta_{p-1}, \beta_{p+1}, \dots, \beta_q \rangle) \text{ if } p < q$$

Whereas we have one line segment, we emphasize the polygon $\langle \beta_{p-1}, \beta_q \rangle$, so $p=q$, then $\psi(p, q)$ is just 0. Otherwise, we let r go from p to q-1, looking at the sum of the costs of all triangles $(\langle \beta_{p-1}, \beta_r, \beta_q \rangle)$ and all polygons $\langle \beta_{p-1}, \beta_r \rangle$ and $\langle \beta_{r+1}, \beta_q \rangle$ finding the minimum. Then $\psi(p, q)$ is the cost of an ideal triangulation for the entire polygon. It requires $O(n^2)$ for larger arrays, and $O(n^3)$ time for a function that does 'n' operations on $O(n^2)$ elements in an array.

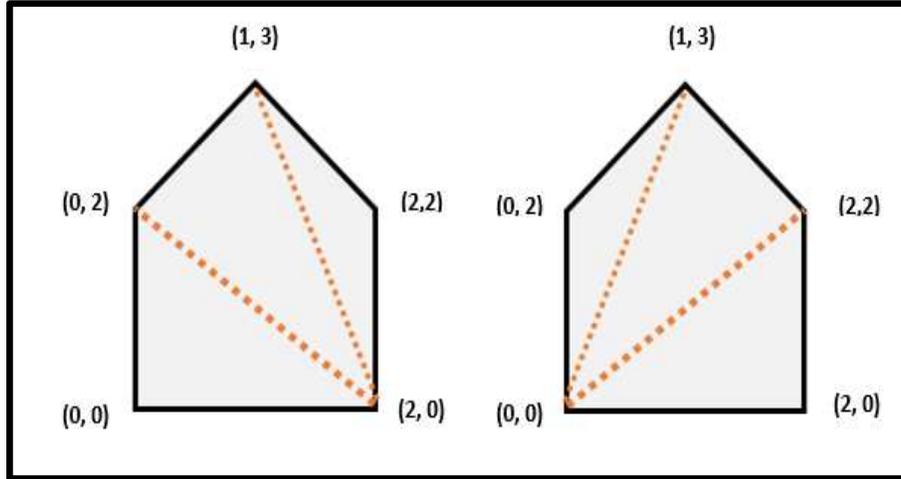


Fig IV Modified Partition method

Algorithm

Input: Polygon P with boundary points E_1, E_2, \dots, E_n

Output: Set of monotone triangles T with vertices $\langle \beta_1, \beta_2, \dots, \beta_n \rangle$

Step 1: Let the least total of triangulation of vertices from p to q be $\psi(p, q)$

Step 2: If $q \leq p + 2$ then $\psi(p, q) = 0$
else

Step 3: $\psi(p, q) = \text{Min} \{ \psi(p, r) + \psi(r + 1, q) + \text{cost}(p, r, q) \}$

Step 4: Cost of a triangle formed by edges (p, q), (q, r) and (r, q) is $\text{cost}(p, r, q) = \partial(p, q) + \partial(q, r) + \partial(r, q)$ which is the sum of the distances between the vertices.

Step 5: Repeat for all the vertices.

3.3 Point location estimation algorithm

The position of the failure node is determined by the point location estimation algorithm as shown in Figure V, from the specified set of monotonous triangles T. The monotonous polygons can be separated into slabs. A unique S-side corresponds to the region between two consecutive segments within a slab. Determine which area contains a given point when a slab is subdivided by non-cross sections, which cross the slab from left to right, into regions. Consequently, we can reduce the problem of

[Type here]

point location to two simpler problems: as the plane divides into vertical slabs, decide which slab contain a sensor node. Determine the area encompasses a given sensor point when a slab is divided into non-intersecting regions, which completely crosses the slab from left to right. The space needed for constructing the slabs and regions in the slabs can be as large as $O(n^2)$ because this method allows for a point location in logarithmic time and is easy to enforce because each slab can cross a substantial part of the segments. A planar partition is divided by the sorted segments of the vertices of the graph into horizontal slabs. Each slab comprises trapezoids that move through the non-intersected parts of the graph edges. Finding the region containing a failed sensor node means identifying first the slab, then the trapezoid that contains it. Segment tree T is the main data structure. Each segment constitutes a plane horizontal slab. Every T leaf contains an balanced tree. These secondary structures contain all the edges of the graph which pass through the slab, ordered from left to right. To find a sensor node, look for the interval containing P in the segment list. Then search for the edges between which P lies within the secondary tree. These two searches give us a processing time of $O(\log n)$. First, a $O(n \log n)$ sort of the vertices by y coordinate is needed. A line-sweep algorithm is then used to build the tree the following way. View each vertex in ascending y order, and hold an ordered list of the graph edges at all times

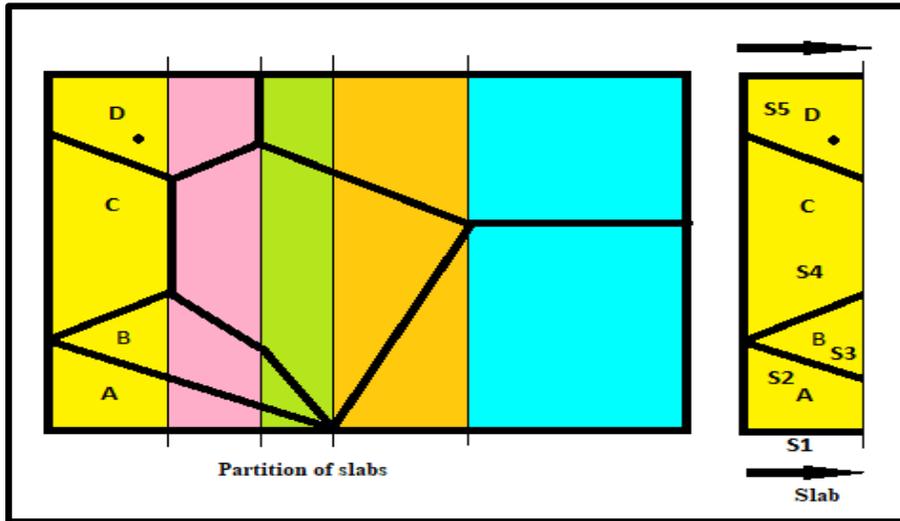


Fig V Slab algorithm

from left to right. The list initially is incomplete. While every vertex is detected, the edges below and incident on it are removed from the list and the edges above are added and its incident on it. If the list is maintained as a balanced tree, the secondary

structure for the slab above it is produced by one single upward sweep at each vertex. Since each edge is inserted and deleted exactly once, the sweep's time-complexity is $O(\log n)$.

Algorithm

Input: Set of monotone triangles T with vertices Set of monotone triangles T with

vertices $\langle \beta_1, \beta_2, \dots, \beta_n \rangle$

Output: Segment queue enclosing segments $SQ = \{ S_1, S_2, \dots, S_n \}$ with failed sensor nodes.

Step 1: : In the adjacent slab $Slab(S) = \{ SL_1, SL_2, \dots, SL_n \}$, find the failed node abscissa. Step 2: For each segment of the initial decomposition, store a pointer to the face above it.

Step 3: Accumulate the segments traversing each slab allowing binary searching.

Step 4: Retrieve the segment inside the slab, in which failed node F (p, q) resides.

Step 7: Insert that segment in the priority queue SQ such that

for $i:=0$ to n

If $SQ = \{ \}$ or \emptyset , then create S_i

else

Insert S_i at the end

3.4 Hole identification

The detection and calculation of the holes in the ROI is at this phase As shown in Figure VI, the point location based hole detection algorithm approach operates separately for every segment. The hole regions are calculated by selecting the sensor nodes closest to segment coordinates and determines the position of the inactive sensor node within that segment. The hole area is calculated using a hole identification method where the energy of the nodes are compared with the threshold value determined using a probabilistic approach. The nodes with energy less than the threshold value are identified as failure node causing coverage holes.

Algorithm

Input: Segment queue enclosing segments $SQ = \{ S_1, S_2, \dots, S_n \}$ with failed sensor nodes.

Output: Location $\Delta(p,q)$ of a failed node causing coverage hole

Step 1: Find the connected set of links for all possible points enclosed in segments where

$$\forall SQ, \text{Number_of_links} = n! / (n-2)! \times 2!$$

Step 2: If $C \neq 0$, and it is a non-negative integer, then exchange hello messages between the possible combinations.

Step 3: Fill the matrix with negative acknowledgement in case of failed reply and

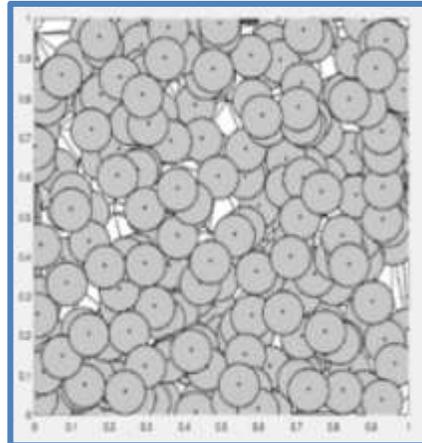
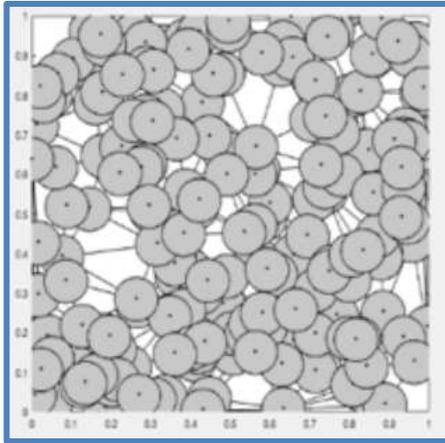
[Type here]

positive acknowledgement in case of success.

Step 4: Retrieve co-ordinate location of negative acknowledgement $\Delta(p,q)$.

4 Experimental setup

Experimental scenario is developed with communication range of 40m, since the communication range is perceived to be twice the sensor range. The stipulations for the model are demarcated rendering to the typical IEEE 802.15.4 MAC / PHY. The sensor node's initial energy is intended to be 50 J of allotted energy and the energy outflow attributed to delivery of every control packet is 0.14 J. Hole is randomly created between the one-hop and completely attached nodes so that diverse groups of disjoint nodes can be formed. Both nodes are provided location coordinates such that they can know about the choice of their one-hop neighbors sensors and locate the failed node.



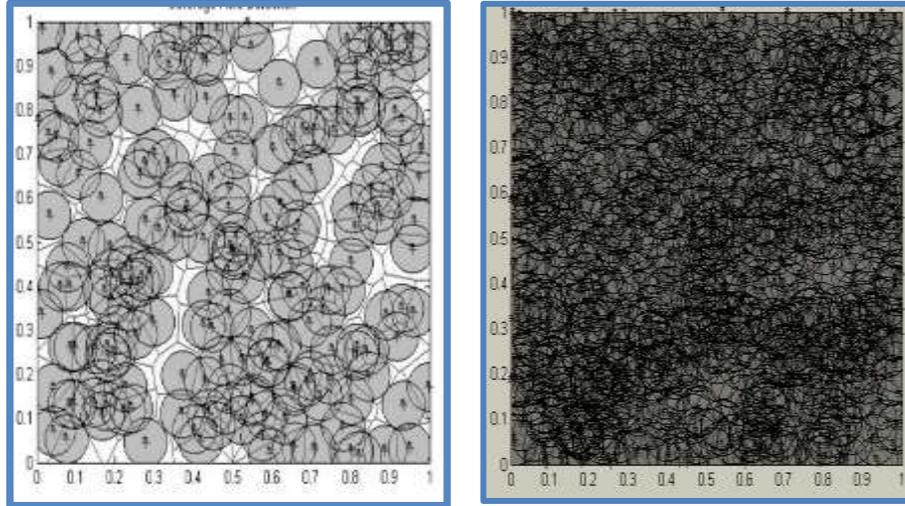


Fig VI Hole Identification for different node densities

5 Performance Evaluation

5.1 Number of Alive nodes

Simulation results show that the node death rate in the proposed algorithm DPLHD is slower than the existing protocols. The proposed algorithm DPLHD tries to vary the standing of the network nodes adaptively so it will keep the network as active as possible. Therefore, the death rate of the nodes within the planned algorithm is sort of delicate. The analysis of alive nodes for different rounds of sending of packets after applying coverage hole detection algorithm is shown in Figure VII.

Goodness of fit

The simple fitting analysis shown in Figure VIII shows that the proposed algorithm shows 100% fitting compared to the existing distributed coverage hole detection algorithm. R-square statistics measure how successful the fit is in explaining the variation of the number of alive nodes. R-Squared is a quantitative key factor contributing to the mean model. R-squared value ranges from 0–1 and the nearer it is to 1 the better it describes the variation of the response results around the mean. So the higher the R-squared value means the better the model. Adjusted R-squared is always lower than R-squared, but it is usually very small if too few sample coefficients in the presence of too much noise are estimated. Table II shows that the R-Squared value of DPLHD is 1 which is greater than the existing DCHD approach [41].

[Type here]

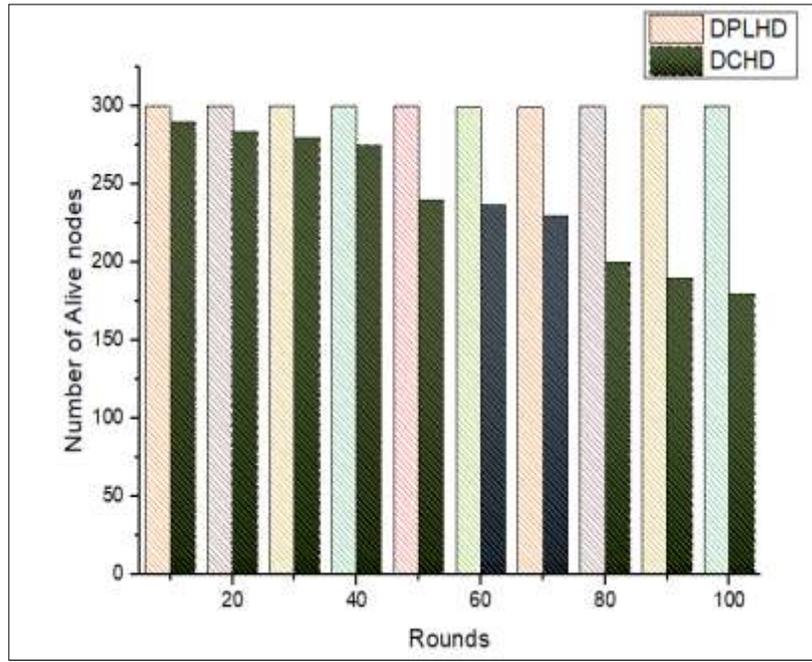


Fig VII Rounds Vs Number of alive nodes

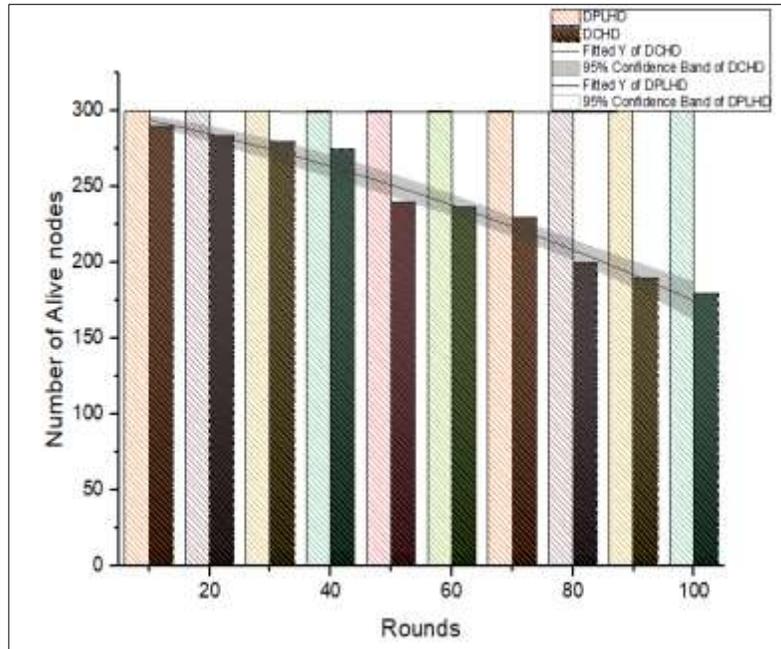


Fig VIII Simple fitting analysis for Rounds Vs Number of alive nodes

Table II Comparison of R-Square values to prove fitness

Method	Number of Points	Degrees of Freedom	Residual Sum of Residual	R-Square(COD)	Adj. Square	R-
DCHD	10	8	447.73788	0.99925	0.99906	
DPLHD	10	8	1.1803	1	1	

[Type here]

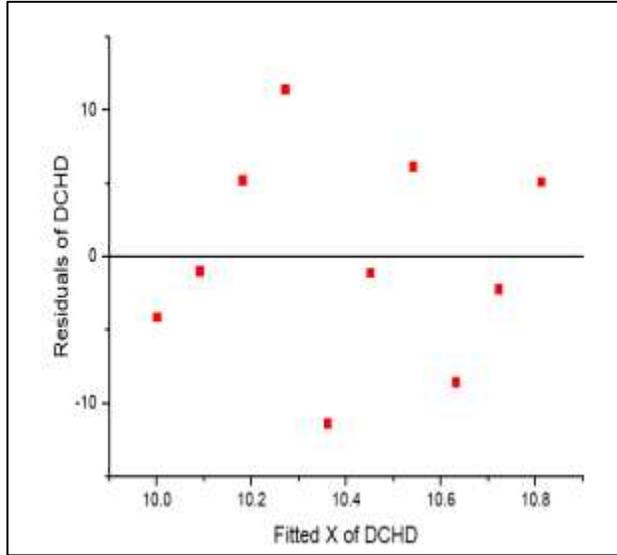


Fig IX a, Graphical Representation of R-squared

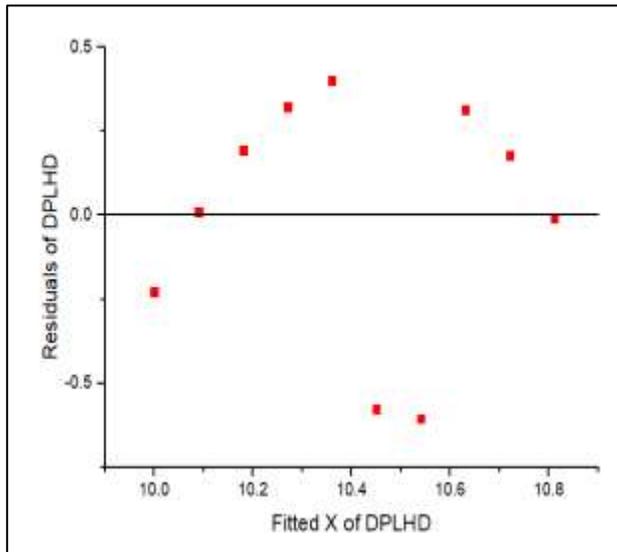


Fig IX b, Graphical Representation of R-squared

5.2 Control packet overhead

Control packet overhead is a measure of the sender's data number bytes for the path between sender and receiver and the cumulative number of application data bytes between sender and receiver. The monitoring overhead packet of DPLHD is equivalent to DHCD for certain numbers of nodes [41]. It is observed that the number of holes available in our protocol is less than DHCD. The control packet overhead is 0 for 100 sensors. Even then, as the number of sensors is over 200, the sensor count improves and a greater number of redundant sensors can be generated in the network by the number of control packets as shown in Fig X and Fig XI..

Table III Comparison of descriptive statistical parameters for control packet overhead

Methods	DPLHD (kB)	DCHD (kB)
N total	10	10
Mean	2.6	4.61
Standard Deviation	1.50555	2.39186
Lower 95% CI of Mean	1.523	2.89897
Upper 95% CI of Mean	3.677	6.32103
Variance	2.26667	5.721
Sum	26	46.1
Minimum	0	0.1
Median	3	5
Maximum	4	8

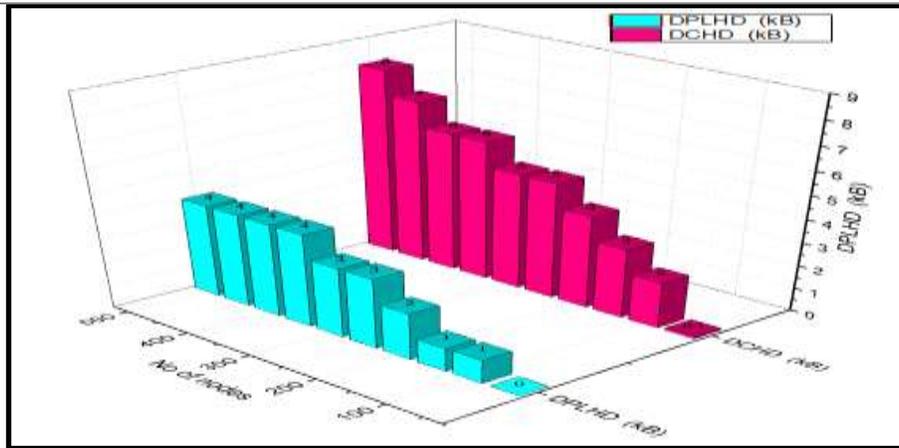


Fig X Control packet overhead VS Number of nodes

[Type here]

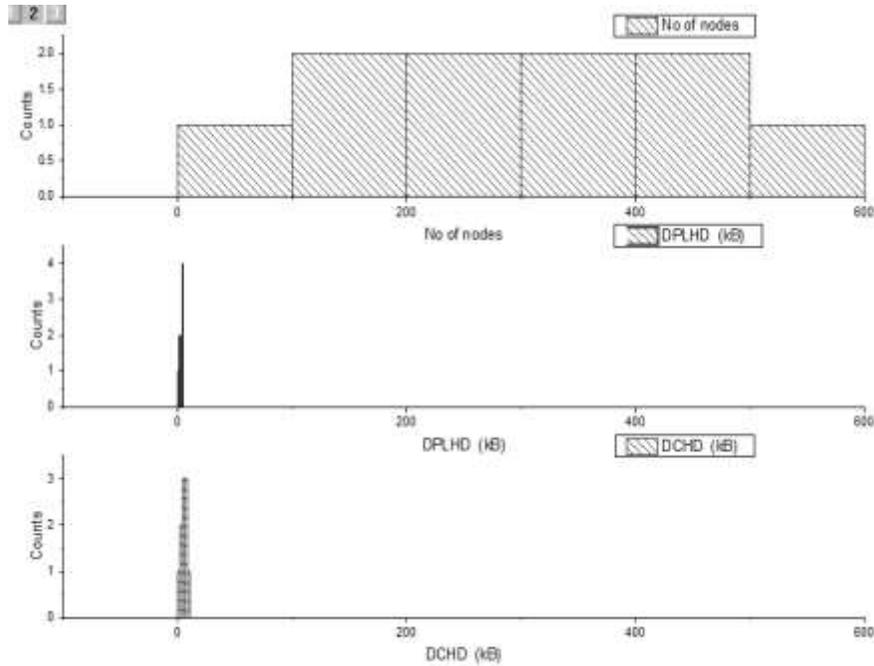


Fig XI Comparison of fit curves for control packet overhead at a same timeline

5.3 Analysis of Energy Consumption

The analysis of the average energy consumption for different numbers of nodes with different deployed sensors in proposed model is presented in figure XII. Here, the energy consumption is analyzed for a simulation run of different number of nodes. The total energy consumption is defined as the energy used to transmit and collect data. Different numbers of deployed sensors can be found to influence the average energy consumption. This test is conducted for cloud based and edge based simulations.

Table IV Average energy consumption vs Number of Nodes

Energy consumption(mJ)		
Number of nodes	DPLHD	DCHD
0	100	130
10	100	120
15	90	100
20	95	130
25	90	100

30	75	120
35	75	100
40	90	100
45	85	130
50	75	90
55	90	150
60	70	120
65	80	140
70	85	200
80	60	150
100	60	140

The energy consumption data are described by descriptive statistics. The sample and measurements are easily summarized. We form the basis for nearly all quantitative data processing along with basic graphic processing.

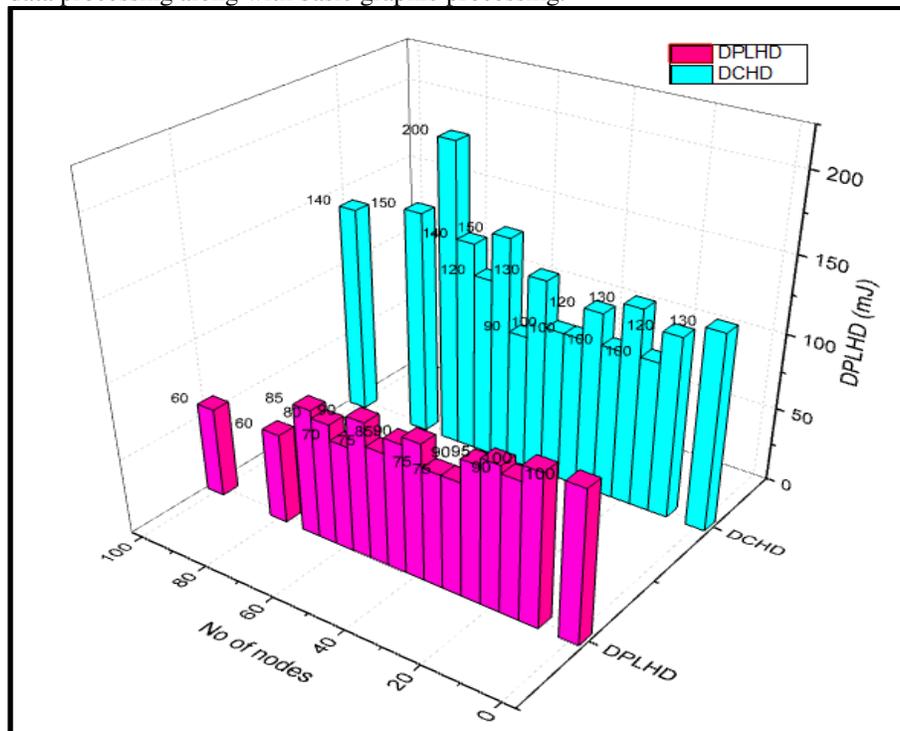


Fig XII Average energy consumption VS Number of nodes

[Type here]

Table V Comparison of descriptive statistical parameters for energy consumption

Methods	DPLHD (mJ)	DCHD (mJ)
N total	16	16
Mean	82.5	126.25
Standard Deviation	12.52	27.29
Lower 95% CI of Mean	75.5	111.7
Upper 95% CI of Mean	89.5	140.7
Variance	156.5	745
Sum	1320	2020
Minimum	60	90
Median	85	25
Maximum	100	200

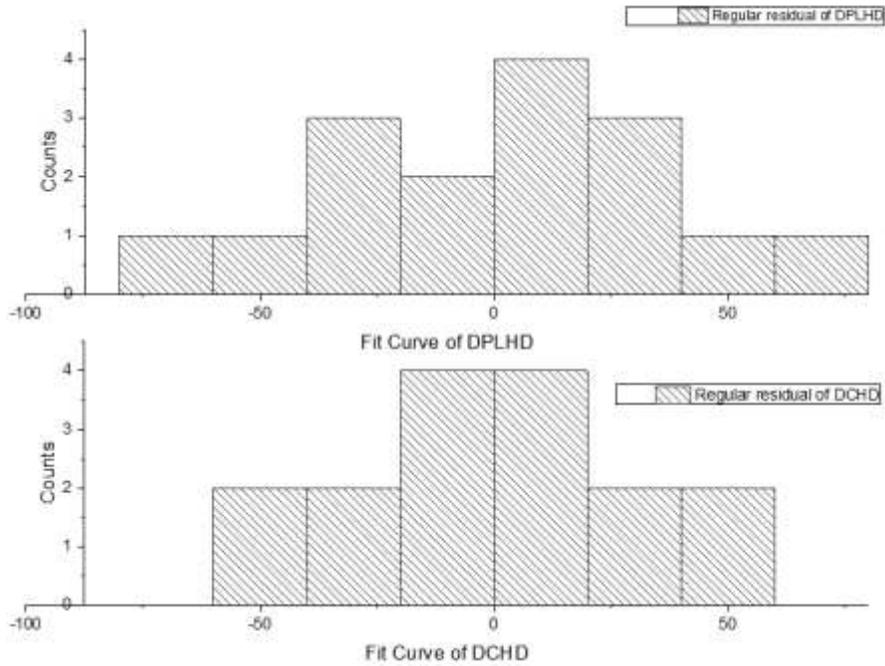


Fig XIII Comparison of fit curves for energy consumption

Complexity analysis

Below is the complexity of the calculation of each stage in DPLHD.

Table VI Computational Complexity

Phases	Complexity
Visibility estimation	$O(nh)$
Modified Partition	$O(n^2)$
Point location estimation	$O(\log n)$
Hole Detection	$O(1)$

The visibility estimate stage simply eliminates the identification of boundary nodes and edges. The overhead network from the transmission of message between sensor nodes is then computed by $O(nh)$. The hole detection step used locally to solve the lowest minimum point is just a step in the process to reduce the overhead network (1). Triangulation and point-location steps evaluate the output in an iterative manner depend on the node density, as the complexity increases slightly, which is shrunk by future techniques.

6 Application and Future work

In vital healthcare applications such as COVID identification and area monitoring, the analysis of coverage troughs is important. It is essential to effectively deal with difficult patient consultations and improving the success rate of rescue; and ensuring effective quality control. Occurrence of coverage hole can harness this features. Usage of IOT with WSNs with proper planning can solve these issues. The proposed DPLHD with IOT applications can enhance the distribution network so that sensors must communicate with services such as the Internet and wireless networks such that information can be relayed to remote health workers efficiently.

7 Conclusion

Coverage hole detection justifies that the need for mission critical software for coverage hole identification is now more crucial. Potential research can also concentrate on the reclamation and optimization of the recovery [37] of troughs following their appearance. Very few methods may be used for mobile sinks or multiple sinks to avoid hole formation [13]. Nodes submit their data to sink if the mobile sink approaches and thereby stop excessive wasted energy supply in multi-hop delivery [12]. To set up a flexible infrastructure, protocols built for these complex networks should be distributed. Geometrical approaches use more resources when detecting hole and retrieving data [12], as they are high in the number of nodes, compared with probabilistic approaches. However efficient hole healing techniques [36] can be applied to enhance the coverage hole detection problems.

[Type here]

Declarations**Availability of data and material**

Not applicable.

Competing interests

The authors declare that they have no competing interests

Funding

Not applicable.

Authors' contributions

Both coauthors contributed significantly to the research and this paper, and the first author is the main contributor.

Acknowledgements

Not applicable.

References

- [1] M. Verma and S. Sharma, "A Greedy Approach for Coverage Hole Detection and Restoration in Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 101, no. 1, 2018, doi: 10.1007/s11277-018-5668-7.
- [2] A. M. Khedr, W. Osamy, and A. Salim, "Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks," *Comput. Commun.*, vol. 124, 2018, doi: 10.1016/j.comcom.2018.04.002.
- [3] S. M. Koriem and M. A. Bayoumi, "Detecting and measuring holes in Wireless Sensor Network," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.08.001.
- [4] C. Qiu, H. Shen, and K. Chen, "An Energy-Efficient and Distributed Cooperation Mechanism for k-Coverage Hole Detection and Healing in WSNs," *IEEE Trans. Mob. Comput.*, vol. 17, no. 6, 2018, doi: 10.1109/TMC.2017.2767048.
- [5] T. Amgoth and P. K. Jana, "Coverage hole detection and restoration algorithm for wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 1, 2017, doi: 10.1007/s12083-015-0407-2.
- [6] X. Feng, X. Zhang, J. Zhang, and A. A. Muhdhar, "A coverage hole detection and repair algorithm in wireless sensor networks," *Cluster Comput.*, vol. 22, 2019, doi: 10.1007/s10586-017-1665-y.
- [7] B. Khalifa, Z. Al Aghbari, A. M. Khedr, and J. H. Abawajy, "Coverage Hole Repair in WSNs Using Cascaded Neighbor Intervention," *IEEE Sens. J.*, vol. 17, no. 21, 2017, doi: 10.1109/JSEN.2017.2755122.
- [8] J. Wang, C. Ju, H. jin Kim, R. S. Sherratt, and S. Lee, "A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs," *Cluster Comput.*, vol. 22, 2019, doi: 10.1007/s10586-017-1586-9.
- [9] W. Li and Y. Wu, "Tree-based coverage hole detection and healing method in wireless sensor networks," *Comput. Networks*, vol. 103, 2016, doi: 10.1016/j.comnet.2016.04.005.
- [10] R. Beghdad and A. Lamraoui, "Boundary and holes recognition in wireless sensor networks," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 1, 2016, doi: 10.1016/j.jides.2016.04.001.
- [11] W. Li and W. Zhang, "Coverage hole and boundary nodes detection in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 48, 2015, doi: 10.1016/j.jnca.2014.10.011.
- [12] F. Kiani, E. Amiri, M. Zamani, T. Khodadadi, and A. Abdul Manaf, "Efficient intelligent energy routing protocol in wireless sensor networks," *Int. J. Distrib. Sens. Networks*, vol. 2015, 2015, doi: 10.1155/2015/618072.
- [13] A. Sinha and D. K. Lobiyal, "Prediction Models for Energy Efficient Data Aggregation in Wireless Sensor Network," *Wirel. Pers. Commun.*, vol. 84, no. 2, 2015, doi: 10.1007/s11277-015-2690-x.
- [14] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors (Switzerland)*, vol. 14, no. 2, 2014, doi: 10.3390/s140202822.
- [15] L. Malathi, R. K. Gnanamurthy, and K. Chandrasekaran, "Energy efficient data collection through hybrid unequal clustering for wireless sensor networks," *Comput. Electr. Eng.*, vol. 48, 2015, doi: 10.1016/j.compeleceng.2015.06.019.
- [16] P. Chanak, I. Banerjee, and H. Rahaman, "Load management scheme for energy holes reduction in wireless sensor networks," *Comput. Electr. Eng.*, vol. 48, 2015, doi: 10.1016/j.compeleceng.2015.05.013.
- [17] H. P. Gupta, S. V. Rao, A. K. Yadav, and T. Dutta, "Geographic routing in clustered wireless sensor networks among obstacles," *IEEE Sens. J.*, vol. 15, no. 5, 2015, doi: 10.1109/JSEN.2014.2385734.
- [18] S. Tanwar, N. Kumar, and J. J. P. C. P. C. Rodrigues, "A systematic review on heterogeneous routing protocols for wireless sensor network," *Journal of Network and Computer Applications*, vol. 53, 2015, doi: 10.1016/j.jnca.2015.03.004.
- [19] J. F. Huang, G. Y. Chang, and G. H. Chen, "A historical-beacon-aided localization algorithm for mobile sensor networks," *IEEE Trans. Mob. Comput.*, vol. 14, no. 6, 2015, doi: 10.1109/TMC.2014.2346777.
- [20] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, no. PA, 2015, doi: 10.1016/j.adhoc.2014.07.012.

[Type here]

- [21] E. Garone, A. Gasparri, and F. Lamonaca, "Clock synchronization protocol for wireless sensor networks with bounded communication delays," *Automatica*, vol. 59, 2015, doi: 10.1016/j.automatica.2015.06.014.
- [22] J. Park, S. Lee, and S. Yoo, "Time slot assignment for convergecast in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 83, 2015, doi: 10.1016/j.jpdc.2015.05.004.
- [23] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage with line-based deployed mobile sensors," *Ad Hoc Networks*, vol. 11, no. 4, 2013, doi: 10.1016/j.adhoc.2010.10.002.
- [24] T. Alskaf, M. Guerrero Zapata, and B. Bellalta, "Game theory for energy efficiency in Wireless Sensor Networks: Latest trends," *Journal of Network and Computer Applications*, vol. 54, 2015, doi: 10.1016/j.jnca.2015.03.011.
- [25] S. M. Nazrul Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks with random node deployment," *Ad Hoc Networks*, vol. 34, 2015, doi: 10.1016/j.adhoc.2014.09.008.
- [26] C. M. Chao, M. W. Lu, and Y. C. Lin, "Energy-Efficient Multichannel MAC Protocol Design for Bursty Data Traffic in Underwater Sensor Networks," *IEEE J. Ocean. Eng.*, vol. 40, no. 2, 2015, doi: 10.1109/JOE.2014.2304311.
- [27] G. Hua, Y. X. Li, and X. M. Yan, "Research on the wireless sensor networks applied in the battlefield situation awareness system," in *Communications in Computer and Information Science*, 2011, vol. 144 CCIS, no. PART 2, doi: 10.1007/978-3-642-20370-1_73.
- [28] Di. Pant, S. Verma, and P. Dhuliya, "A study on disaster detection and management using WSN in Himalayan region of Uttarakhand," in *Proceedings - 2017 3rd International Conference on Advances in Computing, Communication and Automation (Fall), ICACCA 2017*, 2018, vol. 2018-January, doi: 10.1109/ICACCAF.2017.8344703.
- [29] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors*, vol. 12, no. 1, 2012, doi: 10.3390/s120100055.
- [30] Y. Zhang, X. Zhang, Z. Wang, and H. Liu, "Virtual edge based coverage hole detection algorithm in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2013, doi: 10.1109/WCNC.2013.6554783.
- [31] R. K. S. Dhingra, "Coverage Hole Detection in Wireless Sensor Networks," *Int. J. Sci. Res.*, vol. 4, no. 6, 2015.
- [32] P. Si, C. Wu, Y. Zhang, Z. Xia, and L. Cheng, "A hole detection and redeployment strategy in wireless sensor network," *J. Inf. Comput. Sci.*, vol. 8, no. 13, 2011.
- [33] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, "Neighborhood-based topology recognition in sensor networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3121, 2004, doi: 10.1007/978-3-540-27820-7_12.
- [34] S. Funke and C. Klein, "Hole detection or: 'How much geometry hides in connectivity?,'" in *Proceedings of the Annual Symposium on Computational Geometry*, 2006, vol. 2006.
- [35] X. X. Wang, S. Han, Y. Wu, and X. X. Wang, "Coverage and energy consumption control in mobile heterogeneous wireless sensor networks," *IEEE Trans. Automat. Contr.*, vol. 58, no. 4, 2013, doi: 10.1109/TAC.2012.2225511.
- [36] A. Soundarya and V. Santhi, "An efficient algorithm for coverage hole detection and healing in wireless sensor networks," in *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2017*, 2017, doi: 10.1109/IEMENTECH.2017.8076995.
- [37] Z. Kang, H. Yu, and Q. Xiong, "Detection and recovery of coverage holes in wireless sensor networks," *J. Networks*, vol. 8, no. 4, 2013, doi: 10.4304/jnw.8.4.822-828.
- [38] P. K. Sahoo, M. J. Chiang, and S. L. Wu, "An efficient distributed coverage hole detection protocol for wireless sensor networks," *Sensors (Switzerland)*, vol. 16, no. 3, 2016, doi: 10.3390/s16030386.
- [39] W. Abdellatif, H. Abdelkader, and M. Hadhoud, "An energy-efficient coverage hole detection technique for randomly deployed wireless sensor networks," in *Proceedings of 2016 11th International Conference on Computer Engineering and Systems, ICCES 2016*, 2017, doi: 10.1109/ICCES.2016.7822025.
- [40] P. Antil and A. Malik, "Hole detection for quantifying connectivity in wireless sensor networks: A survey," *Journal of Computer Networks and Communications*. 2014, doi: 10.1155/2014/969501.

- [41] J. Zhang, H. Chu, and X. Feng, "Efficient Coverage Hole Detection Algorithm Based on the Simplified Rips Complex in Wireless Sensor Networks," *J. Sensors*, 2020, doi: 10.1155/2020/3236970.