Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 1, pp. 557–563, DOI 10.12694/scpe.v25i1.2397

AN IMPROVED COVERAGE HOLE FINDING SYSTEM FOR CRITICAL APPLICATIONS BASED ON COMPUTATIONAL GEOMETRIC TECHNIQUES

ANITHA CHRISTY ANGELIN* AND SALAJA SILAS†

Abstract. Wireless Sensor Networks (WSNs) contain coverage holes caused by both random node sensor deployment and malfunctioning nodes. Because fixing the battery is challenging, collaborative discovery and assessment of coverage shortfalls, as well as getting rid of these holes, has been recognized as critical in WSNs. While placing nodes for sensors in a large-scale WSN is challenging. This research provides a cost-effective coverage hole detection approach based on collaborative distributed point placement. Create a polygon first by employing an angle estimate approach and neighbor data. Following that, a based on points hole identification technique is used to assess if a coverage issue appears in a large-scale WSN's supplied ROI. Furthermore, the region of the coverage hole is estimated using computational geometry-based polygonal triangulation methods. The accuracy of the method is tested here using statistical data. The results show that it outperforms earlier coverage hole-detecting algorithms. In particular, the method improves coverage rate by 75% when compared to conventional methodologies. It also lowers energy usage by 90%, adding to increased network lifetime. The quantitative favourable results demonstrate the effectiveness of the collaborative distributed point placement technique in detecting and successfully resolving coverage gaps in WSNs. In regards to coverage rate, energy consumption, and network longevity, the system being proposed beats previous coverage hole-detecting techniques.

Key words: Coverage hole; Computational Geometry; Visibility approximation; Triangulation; Energy efficiency.

1. Introduction. Coverage hole identification is required for essential WSN-based applications because of failures of nodes and haphazard implementation [1]. The operation of WSN is impacted by energy scarcity in terms of transmission range, regular or haphazard placement of sensors, localization, planning, coverage, as well as additional issues. Intruder identification in warfare, disaster assistance, medical, and workplace monitoring all employ sensor data [2-5]. Data will be lost, or propagation will be delayed if a node breaks while transmitting. Thus, enhancing coverage requires recognizing coverage holes [6-7]. Probabilistic and computational geometry-based algorithms were used for sensor network coverage hole detection depending on system boundaries and data limitations. By detecting holes in coverage hole detection using node location data. From various studies, It is evident that computational geometry-based algorithms function better than empirical ones [8-9]. This collaborative approach uses computational geometry to classify coverage holes in wireless sensor networks. This method is based on distributed energy-efficient point location-based coverage hole identification. A visibility approximation technique is used to establish where a coverage hole stops, and adjacent nodes begin. Considerations include crossover, sensor categories, and node locations [10]. The coverage hole is identified using a polygon triangulation technique and the one-hop neighbors of the coverage hole nodes. This approach is more energy-efficient.

2. Literature Review. Coverage hole detection strategies based on probabilistic and computational geometry methodologies are surveyed in the literature. Node density calculations using probabilistic approaches for great coverage do not sufficiently support hole identification techniques [11-12]. Each sensor node makes dispersed decisions. The probabilistic technique requires node density but less location-based data [13]. Coverage issues are now undetectable. Computational geometry finds coverage holes [14-15]. The tree-based coverage hole detection method for detecting, shaping, and sizing coverage holes [16]. Coverage holes, on the other hand, are recognized by the relative location of surrounding nodes. The implementation of a sensor network border

*Research Scholar, Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India. (anithachristyangelin1p@gmail.com)
†Professor, Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India. (blessingsalaja@gmail.com)

node structure technique, although higher-density nodes are needed to guess coverage holes [17]. The Distributed Boundary Detection based on the Connected Independent Set (BDCIS) approach precisely recognizes borders and holes. This strategy proposes that nodes aggregate their one-hop neighbor's connection details and create distinct data categories [18-19].

Energy use is substantial while precision is low when compared to other approaches. The distributed virtual force-based hole identification and healing proposed by Zhou et al. [14] reduces unnecessary nodes and holes [20]. The WSNs coverage hole is discovered using a computational geometric analytical approach. The algorithm's network lifetime is shorter than that of others. Se Hang et al. [30] proposed FD-CT (Force-Directed Contour Tracing), a method that uses force-directed algorithms and contour tracing techniques to identify holes in wireless sensor networks without the use of location data or anchor points. The method, however, has problems with false hole detection. Tapas and colleagues [31] proposed using convex hull methods to calculate the circumference of both stationary and portable wireless sensor networks (WSNs). When a border node within a stationary WSN needs to be replaced, this method chooses a new neighbouring node to maximize total coverage along the network's boundary.

Khedr et al. [22] propose classification-distributed Distributed Coverage Hole Detection (DCHD). Unbounded coverage gaps were contained using the WSN algorithm. The approach includes node placement, sensing coverage overlap, and non-overlapping range. Its downside is the high computational overhead. Z. Kang et al. developed a distributed technique that is connectivity-based and coordinate-free [23]. BCPs are used in the technique, which can be done on a single node through neighbor verification. This algorithm is highly difficult and computationally complex. Computational Geometric techniques let neighbors understand their respective positions need localization, the number of nodes, the total number of holes [24-25], hole size, the number of network messages exchanged, the median node degrees, and other factors that influence boundaries identification time, consumption of energy, the precision of detection, and communication costs [26-27]. For such massive networks, distributed protocols assure reliability. Existing methods have several drawbacks, the most notable of which are the high number of messages exchanged the length of time it takes to make a selection and the fact that numerous internal nodes are incorrectly identified as boundary nodes [28-29]. The distributed method is connectivity-based and coordinate-free. BCPs are used in the technique, which can be done on a single node through neighbor verification. The algorithm is sophisticated. In geometric approaches, neighboring nodes can share information to understand the relative coordinates of neighboring nodes.

3. Methodologies. The proposed framework allows the network's nodes that sense to be placed on a 2-D grid. Here, the main nodes inside the goal region are consistently positioned, while border nodes are evenly scattered beyond the intended area's exterior boundaries. Every node lacks precise location data, and the node could be classified as a node within the system or doesn't rely on a starting context. Considering the following circumstances: R_s is the sensing range's radius; R_c does a sensor node's communication range have a radius that allows for $R_c = 2R_s$.

3.1. System model. Every node in the network's hierarchy has a binary sensing design that uses a unique ID that is unique to that node. In Figure 3.1, the Point location-oriented coverage hole detection architecture comprises a Geometrical Visualisation Unit (GVU) for visible estimation to recognize the neighbor node and a minimal cost triangulation approach for finding the polygon's border nodes [18]. The Hole Detection Unit (HDU) is made up of an exact position-based hole detection method that detects the position of the failing node that caused the amount of coverage hole. The hole-related data that the hole detection system evaluates is saved in the hole data repository for use later on.

3.2. Visibility approximation algorithm.

Procedure.

Input: Set of points S enclosing the sensor nodes.

Output: Polygon P with boundary points.

1. Initialize an empty set S enclosing to store the sensor nodes defining the boundary.
2. Initialize a variable m to 0.
   Repeat the following steps:
3. Set E[m] to the current position.

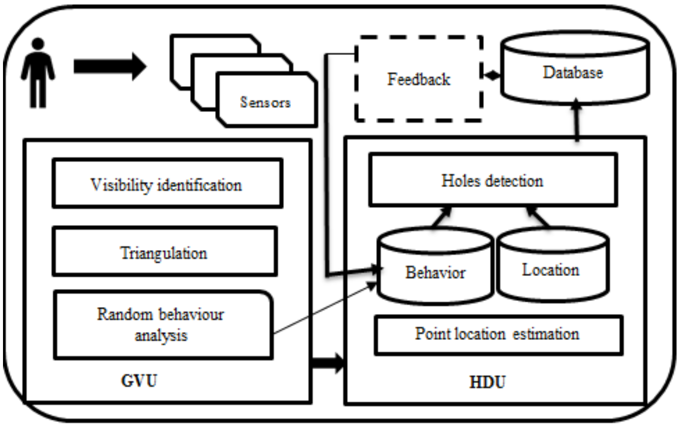

Fig. 3.1: DPHD framework

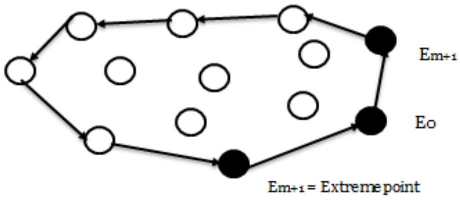

Fig. 3.2: Visibility approximation algorithm

4. Initialize a variable ending position to the position of the first sensor node in set S.
5. For each sensor node S[n] in set S:If ending position is equal to the position of S[n], update ending position to the position of S[n].
6. Increment m by 1.
7. Update the current position to ending position.
8. Repeat the loop until ending position is equal to E[0].

This method processes points in the order they were received. As shown in Figure 3.2, the approximation of visibility begins with m=0 and a polygon point E=0. Take the left-most location and select E_{m+1} so that every point is to the opposite side of the path E_{m+1}. Following that, as illustrated in Figure 5.1, modify the polygon feature set and continue till they hit the left edge. In O(n^1) steps, all polar angles define this point in space as E_m in the polar coordinate centre. The loop within the loop keeps track of every point in the set S, while the outermost loop keeps track of each position in the polygon. As a result, the total run time is O(nh).

3.3. Point location estimation algorithm. Figure 3.2 shows the point location estimation algorithm used to locate the failure node from the monotonous triangle T. Slabs divide monotonous polygons. A slab's S-side is between two successive segments. Find the zone that contains a certain location whenever non-cross segments cross the surface from left to right. Determine whichever vertical block has a sensor node as the horizontal surface splits into vertical slabs to facilitate point localization. Determine the extent of a sensor location when the surface is divided into non-intersecting portions which run from left to right. The method allows for point localization in exponential time. It is simple to implement since each slab may cover a significant portion of its sections. The area required to build the slabs and the region inside the slabs may be as huge as O(n2). The polygon construction process is critical in identifying the coverage hole borders. The time complexity O(n log n) shows that it scales well with the total number of sensor nodes. The memory need for maintaining intermediate information during polygon building is illustrated by the space complexity of O(n).

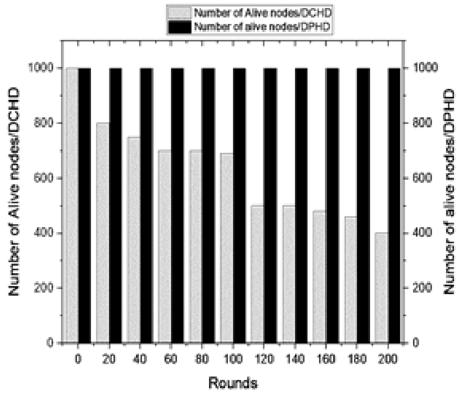

Figure 3 Rounds Vs. Number of alive nodes

Table 1 Number of Alive nodes in DPHD Vs. DCHD

Rounds	Number of Alive nodes /DCHD	Number of alive nodes /DPHD
0	1000	1000
20	800	1000
40	750	1000
60	700	1000
80	700	1000
100	690	1000
120	500	1000
140	500	1000
160	480	1000
180	460	1000
200	400	1000

Fig. 5.1: Modify the polygon feature set

3.4. Coverage hole identification. The hole regions are computed by choosing the nodes in the sensor network that are nearest to the sector dimensions and determining the exact location of the idle sensor node inside the specified segment. The hole size is estimated employing a hole detection technique in which the node power is contrasted to the threshold value obtained through a probabilistic methodology. Nodes having a lower energy consumption below the acceptable value are detected as failing nodes that cause coverage gaps. The hole detection step entails determining whether or not coverage holes exist in the region of interest. The algorithm's time complexity of $O(n^2)$ indicates that the time required for execution rises exponentially with a given number of sensor nodes. The memory overhead during hole detection is shown by the space complexity of O(n).

4. Experimental Setup. As it is expected that the transmitter and receiver distance is twice that provided by the sensors, a 40-meter range is employed in the resultant experimental case. The specs of the model are based on standard IEEE 802.15.4 MAC and PHY criteria. The baseline energy budget for the sensor node is 50 J, and 0.14 J of energy is lost due to the transmission of each control packet. To facilitate the formation of unique clusters of unconnected nodes, a hole is arbitrarily generated between the one-hop and entirely attached node. Every node receives a set of parameters to locate the sensor array of its failing neighbor to obtain information about the other node's choices via its one-hop neighbors.

5. Results and Discussions. Simulation findings reveal that the proposed method for DPHD has a lower node mortality rate than traditional protocols. The proposed method DPHD attempts to automatically alter the status of the network nodes to maintain the entire network as operational as feasible. As a result, the average lifespan of nodes inside the designed algorithm is quite sensitive.

5.1. Number of alive nodes. Figure 5.1 depicts the evaluation of living nodes for multiple rounds of transmission of packets while using the coverage hole identification technique. According to the basic fitting study, the proposed approach fits perfectly when contrasted with the existing distributed coverage hole detection technique. Figure 5.2 depicts a basic fitting evaluation that describes the variation in the aggregate amount of living nodes.

The mean model's quantitative element is R-Squared. The closer R-squared is to one, the more accurately it depicts the range of responses that lie around the median. As a result, higher values for R-squared suggest better predictions. Adjusted R-squared will always be smaller than R-squared, however, it is generally fairly small when assessing inadequate test coefficients amid excessive noise. The R-Squared value for DPHD is one,

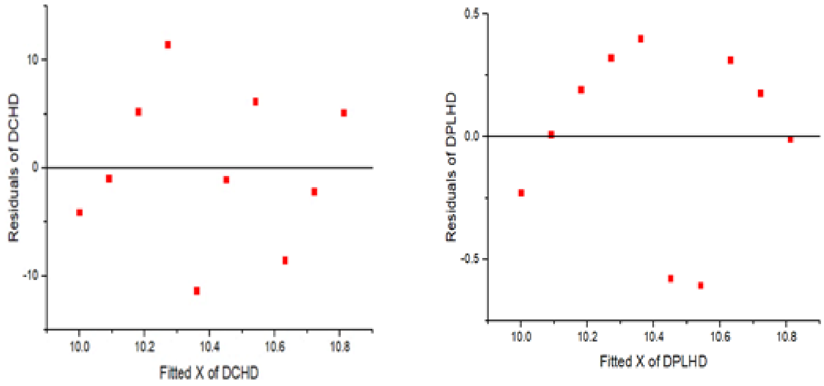

Fig. 5.2: Simple fitting analysis for Rounds Vs Number of alive nodes

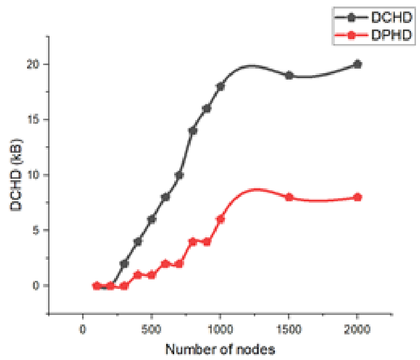

Table 2 Analysis of Control Packet Overhead for DCHD and DPHD

Plot	DCHD	DPHD
Number of Points	12	12
Degrees of Freedom	10	10
Residual Sum of Squares	105.855	8.659
R-Square	0.8282	0.9116
Adj. R-Square	0.81105	0.9028

Fig. 5.3: Control packet overhead VS number of nodes

which is larger than the conventional DCHD technique [25].

5.2. Control packet overhead. Figure 5.3 shows the control portion of the overlay packet, which defines the path between the sender and the recipient as well as the total number of application-specific information bytes transferred. DPHD control packet overhead is compared to DHCD for various node counts [25]. Our protocol outperforms DHCD with fewer availability holes. The simulation reveals no control packet overhead for sensors under 100. When the number of sensors exceeds 200, sensor density increases and control packets can lead to more redundant sensors on the network.

5.3. Average energy consumption. Figure 5.4 depicts a study of the median energy usage for numerous holes with multiple sensors. Let n be the count of neighbors of a sensor Si, and Et and Er denote the amount of energy expended by the sensor when sending and receiving data from its neighbors. Because node Si must broadcast its position data as well as collect location data from its k neighbors, a node's energy usage in the DPHD method may be E = Et + nEr. As seen in Figure 6, the overall energy usage is influenced by the variety of sensors put in the location. The corresponding values are shown in Table 3. This approach employs a huge number of sensor nodes with a variety of holes. Each additional pair of holes increases the amount of energy consumed. This condition is due to the increased number of holes between neighbors that include covering holes.

Descriptive statistics are used to characterize the energy usage data in Figure 5.3. The sample and observations are straightforward. The proposed framework is the foundation for almost every aspect of quantitative analysis as well as basic visual analysis. The corresponding values are shown in Table 2.

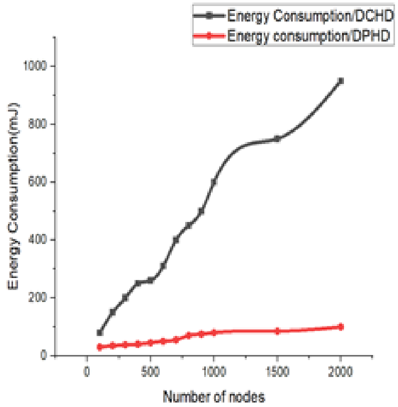

Table 3 Analysis of Energy Consumption for DCHD and DPHD

Equation	y = a + b*x	
Plot	Energy Consumption/DCHD	Energy consumption/DPHD
Weight	No Weighting	
Intercept	60.42289 ± 16.54572	28.99751 ± 3.44326
Slope	0.46388 ± 0.01803	0.03945 ± 0.00375
Residual Sum of Squares	10896.21891	471.89502
R-Square (COD)	0.98511	0.91699
Adj. R-Square	0.98362	0.90869

Fig. 5.4: Average energy usage against node count

5.4. Complexity analysis. The visibility estimation phase simply eliminates the ability to identify nodes and edges that are associated with boundaries. As a result, the maximum computing complexity of this phase is O (nh). To avoid network costs resulting from signal transport among sensor nodes, the phase of hole identification uses just locally stored data to compute the minimal crucial thresholds, hence the complexity of its computation is O(1). Both the triangulation and point position estimate stages continuously decide the result based on the number of nodes, however, a little increases the level of difficulty that can be lowered utilizing later approaches.

6. Conclusions. The detection of coverage holes supports the idea that there is a greater need for mission-critical software to locate coverage holes. The reclamation of holes and the optimization of the recovery following the appearance of holes is another possible area of research that might be conducted in the future. To prevent hole formation, movable sinks or numerous sinks can only make use of a very limited number of ways. If the mobile sink is nearby, nodes will transmit their data to it, preventing an excessive quantity of wasted energy supply during multi-hop delivery. Managing protocols that were developed specifically for these intricate networks is necessary to provide a flexible infrastructure. In comparison to probabilistic methods, geometrical methods require a greater amount of resources while finding holes and extracting data, because they have a greater number of nodes than probabilistic methods. Effective hole healing approaches on the other end of the spectrum, might be used to enhance coverage hole detection concerns.

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.
[2] J. Park, S. Lee, and S. Yoo, "Time slot assignment for converge cast in wireless sensor networks," J. Parallel Distrib. Comput., vol. 83, 2015.
[3] 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.
[4] 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.
[5] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," Sensors, vol. 12, no. 1. 2012.
[6] 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.
[7] W. Abdellatief, 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.
[8] 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.
[9] 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.
[10] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," Sensors (Switzerland), vol. 14, no. 2. 2014.
[11] P. Chanak, I. Banerjee, and H. Rahaman, "Load management scheme for energy holes reduction in wireless sensor networks," Comput. Electr. Eng., vol. 48, 2015.
[12] 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.
[13] 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.
[14] S. M. Koriem and M. A. Bayoumi, "Detecting and measuring holes in Wireless Sensor Network," J. King Saud Univ. - Comput. Inf. Sci., 2018.
[15] 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.
[16] W. Li and Y. Wu, "Tree-based coverage hole detection and healing method in wireless sensor networks," Comput. Networks, vol. 103, 2016.
[17] 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.
[18] R. Beghdad and A. Lamraoui, "Boundary and holes recognition in wireless sensor networks," J. Innov. Digit. Ecosyst., vol. 3, no. 1, 2016.
[19] 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.
[20] 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.
[21] 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.
[22] 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.
[23] Z. Kang, H. Yu, and Q. Xiong, "Detection and recovery of coverage holes in wireless sensor networks," J. Networks, vol. 8, no. 4, 2013.
[24] 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.
[25] 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.

Edited by: Mahesh T R
Special issue on: Scalable Dew Computing for Future Generation IoT Systems
Received: Jul 4, 2023
Accepted: Oct 2, 2023