

Modified Sensor Deployment Algorithm for Hole Detection and Healing using NS2

Vaishnavi Sukumaran, T.P. Saravanabava

M.E, College Of Engineering, Guindy, Chennai, India, 2014

Ph. D, Associate Professor, College of Engineering, Guindy, Chennai, India, 2014

Abstract

One of the fundamental services provided by a Wireless Sensor Network (WSN) is the monitoring of a specified Region of Interest (RoI). Considering the fact that emergence of holes in the RoI is unavoidable due to the inner nature of WSNs that includes random deployment, environmental factors, and external attacks. Assuring that the RoI is completely and continuously covered is very important. This paper seeks to address the problem of these holes in the RoI and healing them. We identify four key elements that are critical for ensuring effective coverage in mobile WSNs: (i) determining the boundary of the RoI, (ii) detecting coverage holes, (iii) determining the best target locations to relocate mobile nodes to repair holes, and (iv) dispatches mobile nodes to the target locations while minimizing the moving cost. We propose a lightweight and comprehensive solution, called Modified Hole Detection, which addresses all of the aforementioned aspects. MHD is an evenly distributed and localized algorithm that operates to (i) identify the boundary nodes and discovers holes within the RoI and (ii) identify the holes with respect to the RoI boundaries. Finally a distributed virtual forces-based local healing approach where only the nodes located at an appropriate distance from the hole will be involved in the healing process.

I. INTRODUCTION

Wireless Sensor Network (WSN) comprises small sensor motes with limited processing and communication power. By design, sensors are quite vulnerable to various forms of failure, caused due to sudden shock during their deployment (air dropping) or depletion of their limited energy resources. Several anomalies can occur in WSNs that impair their desired functionalities resulting in the formation of different kinds of holes, namely: coverage holes, routing holes, jamming holes, and worm holes Ref [2]. In this work we are interested in bounded holes, i.e., holes that are circumscribed by sensor nodes. In this case, *coverage holes*, i.e., areas not covered by any node, and *communication holes*, i.e., areas devoid of any nodes, both considered to be equivalent and will be referred to as holes in common from now on in this paper.

One of the fundamental services provided by a WSN is the monitoring of a specified Region of Interest (RoI), where the main duty is sensing the environment and communicating the information to the sink. Assuring that the RoI is completely covered at all time is very important Ref [3]. However, when the environment is unknown or hostile such as remote harsh fields, disaster areas and toxic urban regions, sensor deployment cannot be performed manually. To scatter sensors by aircraft is one possible solution. However, using this technique, the actual landing position cannot be controlled due to the existence of

wind and obstacles such as trees and buildings Ref [4]. Thus, an event occurring within these holes is neither detected nor reported and, therefore, the main task of the network will not be completed. Thus, it is primordial to provide a self-organizing mechanism to detect and recover coverage & communication.

This paper seeks to address the problem of hole detection in a different perspective and healing them using one of the available methods. Most of the existing solutions use global operations to calculate the size of a big hole and then relocate a group of mobile sensors to heal the hole. While some existing localized solutions requires strong assumptions or even unrealistic ones. The inspiration from the previous works motivates there search presented here. We propose a comprehensive solution, called Modified Hole Detection Mechanism (MHD) that has a very low complexity and avoid some drawbacks noticed in previous works. MHD is a distributed and localized algorithm that operates to identify holes within the RoI. The first phase includes hole identification and border detection. MHD algorithm operates on two types of nodes (i) Nodes circumscribing the hole and (ii) Nodes located at the RoI Extremes. MHD deals with holes of various forms and sizes despite the nodes distribution and density.

The second phase involves the hole healing. It includes area calculation and relocation of mobile nodes. We implement a virtual forces-based local

healing approach based on the Hole Dimension Ref [1], in which the forces will be effective. This allows a local healing where only the few nodes located at an appropriate distance from the hole will be involved in the healing process. The main contribution of this work is the design and evaluation of Holes. This can estimate and effectively overcome the coverage problem in mobile WSNs. This paper makes the following specific contributions. First, a collaborative mechanism, called Hole Detection Algorithm (HDA) and Boundary Detection Algorithm (BDA) is proposed to identify the holes inscribed and boundary holes within the RoI. Second, we use a virtual forces-based Hole Healing Algorithm Ref [1]. This algorithm relocates only the adequate nodes within the shortest range. Experimental results for this approach provides a cost-effective and an accurate solution for hole healing in mobile WSNs.

II. RELATED WORKS

A. HOLE IDENTIFICATION AND BOUNDARY DETECTION

Many related works based on hole identification and border detection has been done. Some of them are as follows: Ref [5] Author as proposed a distributed scheme that is based on the communication topology graph. A node decides whether it is on the boundary of a hole by comparing its degree with the average degree of its 2-hop neighbors. Not all boundary nodes can be identified correctly by this algorithm. Indeed, for a large WSN with a few holes this method is not efficient. Ref [6] presented a coordinate-less method to identify hole boundaries in WSNs. They assume a uniform node distribution in non-hole areas. In a recent paper Ref [7], the same author has proposed a deterministic approach for boundary recognition that does not rely on a uniform node distribution but requires a high node density. Ref [8] in this it described a distributed algorithm to find the boundary nodes by using only connectivity information. They exploit a special structure of the shortest path tree to detect the existence of holes. The authors did not provide a complexity analysis but the proposed algorithm relies heavily on a repetitive network flooding. Ref [9] presented a hole boundary detection algorithm assuming the relative geographic information of 2-hop neighbors. The proposed algorithm uses a best-effort approach in order to maintain synchronization among nodes in the network.

B. HOLE HEALING RELATED WORKS

Ref [10 & 11], Mobile WSN are equipped with mobile platform and are capable of moving around after initial deployment. Underlying reason behind this mobility is to achieve max coverage of

the RoI. Ex: Robomote [19] & iMouse. In the process of hole healing Ref [12 - 15] enhances coverage by coverage pattern-based movement; the target location for the node movement was calculated based on predefined coverage patterns, while mobile nodes are likened as electromagnetic particles in virtual forces based movement. Ref [16] The RoI is partitioned into many small grid cells, and the number of nodes in each grid cell is considered as the load of the cell Ref [16]. In Ref [17], authors consider the point coverage problem with novel evaluation metric, coverage radius. Ref [18] proposed three different deployment protocols that relocate mobile sensors once coverage holes are detected using Voronoi diagrams.

III. PROBLEM STATEMENT

Anything can occur in wireless sensor networks that impair their desired functionalities in real time i.e., sensing and communication. Different kinds of holes can form in such networks creating geographically correlated problem areas such as coverage holes, routing holes, jamming holes, sink/black holes and worm holes, etc. The target field that is supposed to be 100% covered by the densely deployed nodes may have coverage holes, areas not covered by any node, due to random aerial deployment creating voids, presence of obstructions, and more likely, node failures etc., these nodes may not be able to communicate correctly if routing holes, areas devoid of any nodes, exist in the deployed topology. Thus the network fails to achieve its objectives. Some of these anomalies may be deliberately created by adversaries that are trying to avoid the sensor network. These malicious nodes can jam the communication to form jamming holes or they can overwhelm regions in the sensor network by denial of service attacks such as sink/black/worm holes Ref [2] to hinder their operation normally based on trust.

We are interested in large bounded holes, i.e., large holes that are circumscribed by sensor nodes. In this case, coverage holes, i.e., areas not covered by any node, and communication holes, i.e., areas devoid of any nodes, become equivalent and will be referred as hole in common here then. The problem is to design a mechanism for detecting and recovering holes by exploiting only the nodes mobility. Also to make sure that the holes inside and near the boundaries of RoI are considered. But the holes on the border that are the resultant of the initial deployment are not addressed for the healing process.

Most of the hole detecting approaches involves computational geometry tools, such as Voronoi Diagram to identify the hole Ref [3, 12, 18]. With the help of these algorithms, each node

irrespective of each other tries to detect and repair the holes in its Voronoi Polygon Ref [18].

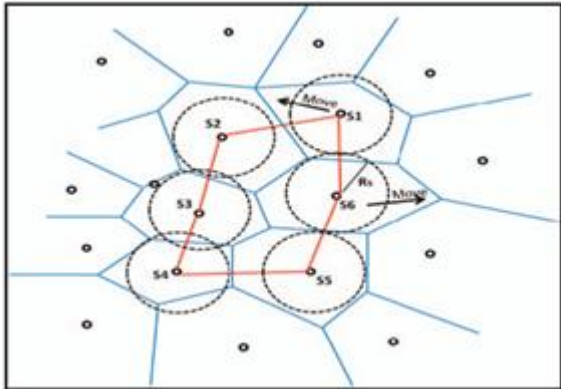


Fig: 1. Ineffectiveness of VOR in treating large holes. Ref [18]

This approach is not suitable for large holes because moving of nodes without the knowledge of the other node movements may lead to creation of new holes or will not help to recover the hole problem completely and efficiently.

Before discussing our proposal, we make the following assumptions:

1. A dense mobile WSN is deployed in an obstacle free RoI.
2. Deployment can be deterministic or random.
3. All the deployed nodes are homogeneous (processing power, communication & energy consumption).
4. Location information of the each node is known by using some localization scheme with respect to RoI. Ref [20].
5. Each node is aware of the boundary range information of the RoI.
6. We consider the isotropic sensing model (uniformity in all orientations). We assume that $R_c \geq 2R_s$ to ensure a common standard.

IV. PROPOSED MECHANISM

In our algorithm we propose a Modified Hole Detection mechanism (MHD) to detect the hole within the RoI and boundaries of the RoI then heal the holes. MHD includes the (i) Hole Detection Algorithm and (ii) Boundary Detection Algorithm. In this approach, holes that are circumscribed by nodes are taken into account and preceded for healing of the same. The mobility feature of the nodes is exploited to heal the holes. Also, only the optimum number of nodes located around the hole takes part in the healing process. This paper covers identification of holes in the boundaries whereas healing of such boundary holes is maintained as future work.

Before stepping into the detection of hole, we need to look into few things like how to detect,

calculate and relocate the mobile nodes? In the following sections we shall address all these issues.

In our Hole Detection Algorithm (HDA), the existence of hole in the RoI is determined and its boundary is computed. Later which the Hole Healing Algorithm (HHA) is followed Ref [1]; simple healing process is invoked such that only selective nodes located within the certain range will be involved in hole healing. Such selected mobile nodes are mutually attracted towards the hole center maintaining a fixed distance to achieve a minimum overlapping distance between them to avoid congestion. In the Boundary Detection Algorithm (BDA), we discover the existence of improperly covered RoI boundaries.

A. MODIFIED HOLE DETECTION APPROACH

In geographical greedy forwarding, a source node knows the location of the destination node, either by acquiring it from a location service or by computing it using a hash function in a data-centric storage scheme. A packet is forwarded to a 1-hop neighbor in the direction towards the destination from the current node. This process is repeated until the packet reaches the destination or the packet is stuck at a node whose all 1-hop neighbors are far away from the direction of destination. The node where a packet may get stuck is called a local minimum or stuck node Ref [9]. But on the other hand, if there is no sink or destination available for the deployed nodes then the possibility of identifying a stuck node becomes impossible.

To overcome this issue, we propose an algorithm in which, each node is capable of identifying itself whether it is a stuck node or not, irrespective of the presence of a sink or storage data center. In our algorithm QUAD Rule, we ensure that each individual node is capable of communicating in 360° with respect to its communication range. The QUAD rule specifies that a node is not a stuck node if where there exists at least one 1-hop neighbor within the range of angle spanned by itself which is less than $\pi/4$.

The process of discovering a hole is first initiated by the identification of stuck nodes. Each node executes the QUAD rule to check whether the node itself is a stuck node or not.

B. QUAD RULE & RIGHT HAND RULE

QUAD Rule: The communication range of each node (say 24m in radius & 360° in direction) is split into 4 quadrants of 90° each and checks for the existence of at least one 1-hop neighbor within the quadrant. If and only if this condition is satisfied for all the 4 quadrants the node is not a stuck node else it

identifies itself as a stuck node. In other word, if there is no node for it to communicate in at least 1 quadrant then it is identified as stuck node and the corresponding quadrant is known as the stuck quadrant.

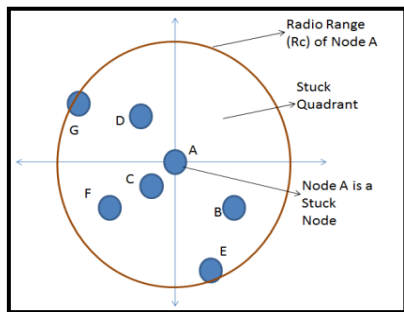


Fig: 2. Identification of Stuck Node and Stuck Quadrant

This rule is executed by each node to identify whether it is a Stuck Node or not. If yes, then it generates a discovery packet and forwards it accordingly.

Right Hand Rule: Once a node identifies itself as a stuck node it generates a HD packet and transmits it to other nodes in order to collect the information of similar nodes in the network. This is done by forwarding the HD packet in a particular direction so that it will return back to itself; in here we follow the right hand rule.

The node to which the HD packet has to be sent is determined by the following condition, each stuck node will now be aware in which quadrant they have identified themselves as a stuck node. The adjacent quadrant to the stuck quadrant along the right hand side will be chosen for the selection of next hop neighbor, even in this quadrant there may be more than one 1-hop neighbor, in such a case the node which is closest to the stuck quadrant will be identified as the next hop neighbor for forwarding the HD packet. By this way the traveling of the HD packet will be faster and identification of hole boundary will be more precise.

An HD packet is constructed with stuck node information and forwarded to the selected node, the node that receives the packet may or may not be a stuck node itself, if it is not a stuck node then it forwards the packet as per the RH rule from which it has received the HD packet, if it happens to be a stuck node then it fills in its Node-ID and location information and then forwards.

C. HOLE DETECTION ALGORITHM

All the nodes that are marked as stuck nodes by the QUAD rule will check its location information against the available boundary range of RoI. As the result they differentiate themselves from the

boundary nodes of the RoI and proceed with the process of determining hole boundary. Once a node identifies itself as a stuck node it generates a new Hole Discovery (HD) packet, marked with its ID b_i and forwards it to the next boundary node b_{i+1} where node b_{i+1} is chosen based on Stuck Quadrant-Right Hand Rule. If node b_{i+1} is a stuck node then inserts its location information into the received HD packet and forwards it to the node b_{i+2} in the same way creating a cascading effect. This process is repeated until the HD packet has traveled around the hole and eventually been received by the initiator node b_i .

The returning HD packets are expected to be received at the chosen node (1-hop neighbor closest to the left side of the stuck quadrant). Once the HD packet reaches the node from where it has been initiated in this case Node b_i . Node b_i extracts the locations of the boundary nodes $\{b_2, b_3, \dots, b_n\}$ from the received HD packet. Then it selects two nodes b_m and b_n so that the distance between them is the longest between any two nodes in the set of boundary nodes to determine hole center.

$$Distance(b_m, b_n) = \max \left\{ \frac{Distance(b_j, b_k)}{b_j, b_k} \right\} \in \{b_2, b_3, \dots, b_n\}$$

The determined hole center is the midpoint v of segment b_m, b_n .

$$X_v = \frac{X_{b_m} + X_{b_n}}{2} \quad Y_v = \frac{Y_{b_m} + Y_{b_n}}{2}$$

In Ref [1] a mechanism to prevent redundancy in the discovery process is performed with a basic idea to remove redundant HD packets as soon as possible. The criterion for judging whether a HD packet is redundant is as follows: at each node, if a HD packet arrives and discovers that the packet has a Hole-ID greater than a Hole-ID carried by a packet already passed, the packet will be considered redundant and it will be deleted.

This paper addresses the above mentioned problem as the HD packets are sent only to the chosen 1-hop neighbor and expected from the chosen another 1-hop neighbor; the probability of receiving duplicate HD packets is negligible. Also, the computational complexity and processing overhead for detecting duplicate HD packet as been completely reduced.

Finally, the node that has the smallest Node-ID removes the HD packet and names itself as Hole Manager (HM). It will be responsible for the hole-healing announcement. This is in contrast to the Hole Manager Selection strategy where one can pick the node with the largest residual energy among all the boundary nodes. Ref [1]

D. BOUNDARY DETECTION ALGORITHM

All the nodes that are marked as stuck nodes by the QUAD rule compares its location with the available boundary range of the RoI in-order to identify whether it belongs to the extremities of the RoI. As the result of comparison if a node exists closer to the boundary range exactly at the stuck quadrant then the node is identified itself as the boundary node of the RoI.

$$2 \leq nSQ < 4, \text{ where } n \text{ is number of Stuck Quadrants}$$

E. HOLE HEALING ALGORITHM

At the end of the hole detection phase, the healing phase is executed following Ref [1]. The nodes locomotion facility is exploit to heal the holes. The relocation algorithm is completely distributed and it is based on the concept of virtual forces. To heal the discovered hole we define an attractive force that acts from the hole center and attracts the nodes towards this center. Similarly, a repulsive force is defined among nodes to minimize the overlapping in between. It is defined as the Hole Healing Algorithm (HHA) in which the forces will be effective. This allows a local healing where only the nodes located at an appropriate distance from the hole will be involved in the healing process.

In this phase, the HM node plays the role of determining the hole dimension and informing nodes on their movement. The HM node has information about the size of the hole and boundary nodes.

The hole HHA constitutes the basic idea of the algorithm; the determination of this area will determine the number of nodes that must be relocated to ensure a local repair of the hole. After the identification of the hole by the MHD algorithm, the HM node calculates the center and the size of the hole. As mentioned before, the hole is approximated by a circle whose radius is the longest distance between two hole boundary nodes.

Therefore, to determine the HHA, we need to determine the radius of the circle that defines the HHA. To find the appropriate radius, we have used an iterative approach based on the following formula:

$$R = r * (1 + \beta), \beta \in R^+$$

where, r is the hole radius β is a positive constant, which depends on the nodes density and the sensing range R_s .

For $\beta = 0$, we start with a radius equals to the estimated hole radius r due to the fact that this area may contain a sufficient nodes to heal the hole. The area defined by this circle (denoted HHA-0) is equal

to πr^2 . the number of nodes necessary to cover the area HHA-0 is equal to:

$$\frac{\pi r^2}{\pi R_s^2} = \frac{r^2}{R_s^2}$$

The HM node calculates the number of nodes present in the HHA-0. It solicits its direct neighbors to calculate the number of their neighbors in this area. It sends a Hole Healing Area Determination packet containing information about the hole. This communication is done over the GG to reduce the number of exchanged messages. If the number of nodes found by the HM node is less than the required number to heal the hole, the movement of these nodes will create new holes. To avoid this scenario the HM node starts a new round by increasing β value, and then it repeats this process until it finds a sufficient number of nodes to recover the hole. Ref [1]

As far as we have seen, it is a simplest case if there is only one hole in the network, but most of the time that will not be the real time scenario. In most of the case there would be more than one hole, in this case the algorithm must be adaptive for healing all the holes.

The readapting of the algorithm is as follows, when the holes and the hole boundaries are all identified, the Hole Manager with the lowest Node ID will proceed with the Hole Healing, once it is done, this will be followed by the next lowest Node ID. Ref [1]

V. EXPERIMENTS

For experimental purpose, we have implemented Bi-Phase Mechanism in the NS2 simulator. This is done by varying the size and position of the hole created within the RoI.

In the first one, we vary the radius of the hole created within the RoI. We use a deterministic deployment strategy with holes of different radii.

Experiment 1

PARAMETERS	SCENARIO I
Max (X,Y) in (m,m)	(200,200)
Deployment	Deterministic
$R_s(m), R_c(m)$	12,24
Simulation Time	200
Max Speed (m/s)	10
Number of Nodes	169
Routing Protocol	AODV

With Deterministic Sensor Deployment

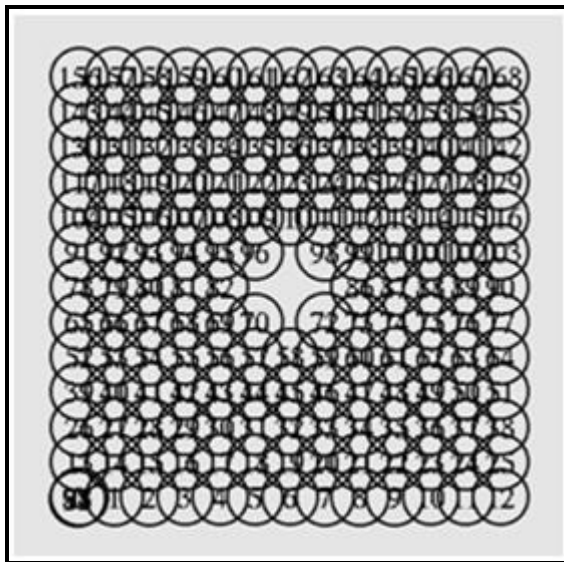


Fig. 3. Original RoI with emergence of holes

The first test is performed with a network where sensor deployment is deterministic and the existence of hole is small in area.

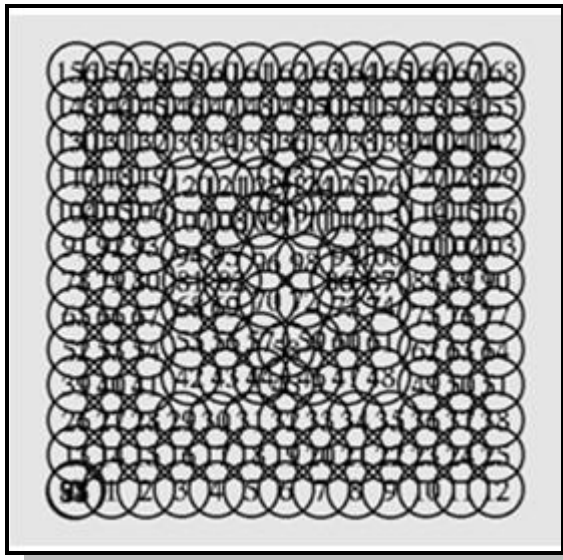


Fig. 4. Repaired RoI

The repaired Region of Interest is fully covered and hole-free as depicted in Fig. 4

Statistics - Deterministic deployment

PARAMETERS	DIMENSIONS
Hole Centre (X,Y) in (m,m)	(91,91)
Hole Area in m ²	11304
Healed Sensor Area in m ²	452.15
Max. no. of nodes selected for hole healing	44
No. of nodes used for healing	25

Experiment 2

PARAMETERS	SCENARIO II
(X,Y) in (m,m)	(200,200)
Deployment	Random
Rs(m), Rc(m)	12, 24
Simulation Time	200, 250, 300
Max Speed (m/s)	10
Number of Nodes	169, 200, 300
Routing Protocol	AODV

With Random Sensor Deployment

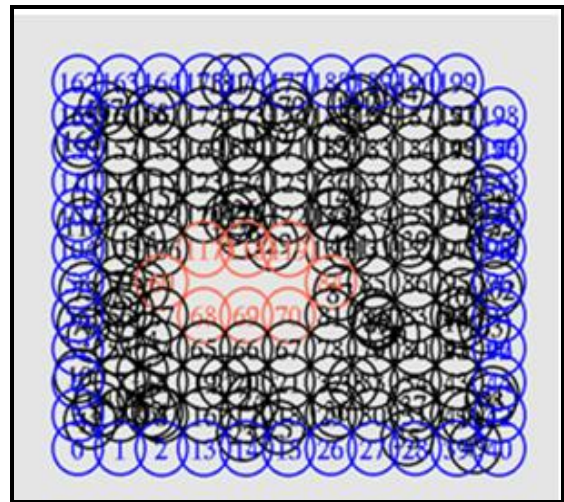


Fig. 5. Phase 1 – Hole and Border detection completed in original RoI

The second test was performed with Random Sensor Deployment. Fig. 5 highlights the completion of phase 1 - Hole and Border determination where the next step involves Hole Healing

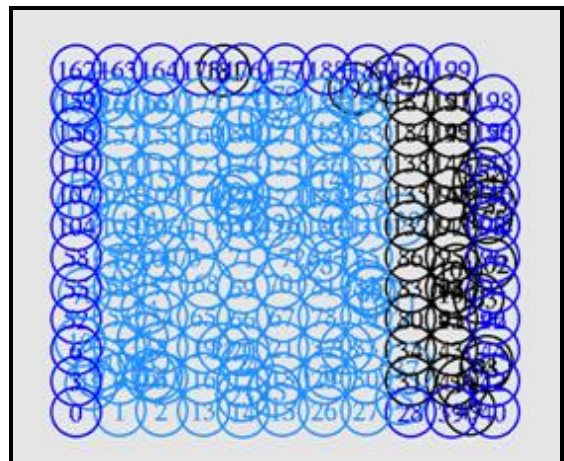


Fig. 6. Phase 2 – Hole Healing Process completed

Fig. 6 depicts the completion of phase 2 – Hole Healing process after which HD packet exchange takes place to check the presence of holes and RoI boundary.

Statistics - Random deployment

PARAMETERS	DIMENSIONS
Hole Centre (X,Y) in (m,m)	(80,75)
Hole Area in m ²	20096
Healed Sensor Area in m ²	452.15
Max. no. of nodes selected for hole healing	122
No. of nodes used for healing	44

VI. RESULTS

Finally, the obtained results allow us to assert that the detection and healing of holes has been performed effectively for the improvement of network coverage.

VII. CONCLUSION

This paper has proposed and implemented a lightweight and comprehensive Modified Hole Detection. The protocol uses Hole Detection Algorithm (HDA) and Boundary Detection Algorithm (BDA) to detect holes in the network. Compared to the existing schemes, MHD has a very low complexity and deals with holes of various forms and sizes despite the nodes distribution and density. By exploiting the virtual forces concept, it relocates only the adequate nodes within the shortest time and at the lowest cost. Through the performance evaluation, we validated using different criteria and showed that it detects and heals the holes despite their number or size with less mobility in various situations. The evaluation results demonstrate that it provides a cost-effective and an accurate solution for hole detection and healing in mobile WSNs.

In the future, we plan to investigate on multiple holes and work on holes located at the network boundary.

REFERENCE

[1] Mustapha Reda SENOUCI, Abdelhamid MELLOUK, Senior Member, IEEE and Khalid ASSNOUNE, Localized Movement-Assisted Sensor Deployment Algorithm for Hole Detection and Healing, in Mobile Computer Science from USTHB in 2009
 [2] Ahmed, N., Kanhere, S. S., Jha, S. The holes problem in wireless sensor networks: a survey. SIGMOBILE Mob.Comput. Commun. Rev 2005; 9(2), pp. 4-18.
 [3] Wang, B. Coverage Control in Sensor Networks. Springer, NY, 2010.

[4] Guiling Wang, Guohong Cao, and Tom La Porta, The Pennsylvania State university Movement-Assisted Sensor Deployment, Department of computer Science & Engineering, 2004
 [5] Kun, B., Kun T., Naijie G., Wan L. D. and Xiaohu L. Topological Hole Detection in Sensor Networks with Cooperative Neighbors. ICSNC'06, Los Alamitos, CA, USA, 2006, pp. 31.
 [6] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In the Proc. ALGOSENSORS, Springer LNCS vol. 3121, 2004, pp. 123-136.
 [7] S. P. Fekete, M. Kaufmann, A. Kröller, and N. Lehmann. A new approach for boundary recognition in geometric sensor networks. In Proc. 17th Canadian Conference on Compu. Geom., 2005, pp. 82-85.
 [8] Y. Wang, J. Gao, and S.B. Mitchell. Boundary recognition in sensor networks by topological methods. MobiCom '06, ACM, NY, USA, 2006, pp. 122-133.
 [9] Shirsat Amit and Bhargava Bharat. Local geometric algorithm for hole boundary detection in sensor networks. Security and Communication Networks, 2011, 4(9), pp. 1003–1012.
 [10] Sibley, G.T., Rahimi, M.H. and Sukhatme, G.S. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11-15 May, 2002, pp. 1143-1148.
 [11] Tseng, Y.C., Wang, Y.C., Cheng, K.Y. and Hsieh, Y.Y. iMouse: An integrated mobile surveillance and wireless sensor system. IEEE Computer, 2007, 40(6), pp. 76–82.
 [12] Younis, M. and Akkaya, K. Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey. Elsevier Ad Hoc Network Journal, 2008, 6(4), pp. 621-655.
 [13] Wang, B., Lim, H. B. and Di, M. A survey of movement strategies for improving network coverage in wireless sensor networks. Computer Communications, Elsevier, 2009, 32, pp. 1427-1436.
 [14] Yong, Z. and Li, W. A Sensor Deployment Algorithm for Mobile Wireless Sensor Networks. CCDC'09, 17-19 June, 2009, Guilin, China, pp. 4642-4647.
 [15] Zou, Y. and Chakrabarty, K. Sensor Deployment and Target Localization in

- distributed sensor networks. ACM Transactions on Embedded Computing Systems, 2004, 3(1), pp. 61–91.
- [16] Yangy, S., Liz, M. and Wu, J. Scan-Based Movement-Assisted Sensor Deployment Methods in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(8), pp. 1108-1121.
- [17] X. Li, H. Frey, N. Santoro, and I. Stojmenovic. Strictly Localized Sensor Self-Deployment for Optimal Focused Coverage. IEEE Transactions on Mobile Computing, Nov. 2011, 10(11): 1520-1533.
- [18] Wang, G., Cao, G. and La Porta, T.F.L. Movement-Assisted Sensor Deployment. IEEE Transactions on Mobile Computing, 2006, 5(6), pp. 640-652.
- [19] Gabriel T. Sibley, Mohammad H. Rahimi and Gaurav S. Sukhatme, Robomote: A Tiny Mobile Robot Platform for Large-scale Ad-hoc Sensor Networks, Robotics Research Laboratory, Department of Computer Science, University of Southern California, Los Angeles, CA 00080-0781, May 2002
- [20] Langendoen K and Reijers N. Distributed localization in wireless sensor networks: a quantitative comparison. Computer Networks, 2003, 43(4), pp. 499-518.