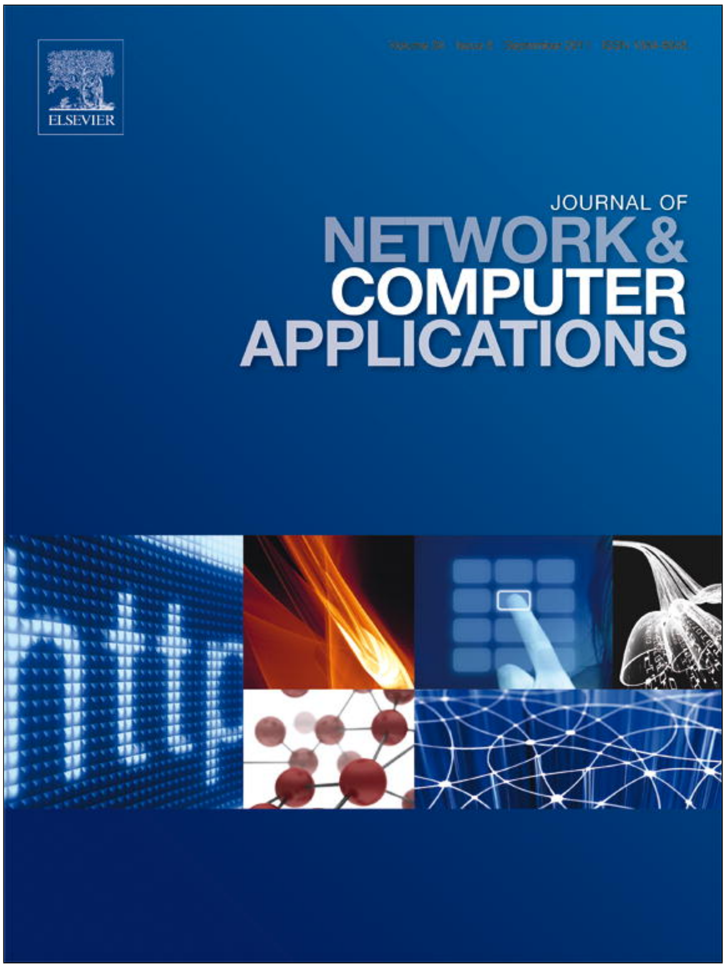
ELSEVIER
JOURNAL OF
NETWORK &
COMPUTER
APPLICATIONS

Journal of Network and Computer Applications 34 (2011) 1743–1756

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks

Hwa-Chun Ma [a,c], Prasan Kumar Sahoo [b,*], Yen-Wen Chen [a]

[a] Department of Communication Engineering, National Central University, Jhongli 32001, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, Chang Gung University, Kwei-Shan 33302, Taiwan, ROC
[c] Department of Information Management, China University of Technology, Taipei 11695, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 October 2010
Received in revised form
8 May 2011
Accepted 12 June 2011
Available online 29 June 2011

Keywords:
Wireless sensor networks
Coverage hole
Hole detection
Distributed

ABSTRACT

In wireless sensor networks, the purpose of surveillance cannot be fulfilled if coverage holes are generated due to accidental death of the nodes caused due to technical failures, explosions and malicious activities or power exhaustion. Since, sensors are normally deployed randomly over the dense forests and harsh terrains, it is not possible to find out the coverage holes manually. Hence, in this work a computational geometry approach based distributed hole detection protocol is designed to find out the coverage holes in a post deployment scenario. An efficient geometric method with proper theoretical basis is used to detect the coverage holes of the wireless sensor network, where communication and sensing range of the nodes are same. Performance evaluation of our protocol shows that the hole detection time and energy consumption due to hole detection outperforms over similar hole detection protocols.

1. Introduction

In wireless sensor networks (WSNs), sensors are distributed randomly with the help of helicopters or rocket lunchers to form the network dynamically without help of any infrastructure. The deployed nodes are normally inexpensive, tiny and battery-powered sensing devices with ability of sensing, processing, and communicating data and are expected to transmit the data to a remote base station. In order to monitor the region of interest, sensor nodes are deployed appropriately with sufficient number of sensors to ensure a certain degree of redundancy. However, because of the constraints of lightweight and low-capability sensor nodes with limited processing power, memory space, battery life, radio ranges, and communication bandwidth, the task of monitoring the network becomes more demanding. Hence, the design of wireless sensor networks can be greatly affected by the geometric distribution of the sensors deployed in the underlying environment.

In WSNs, distributions of sensors are not usually uniform due to random aerial deployment, presence of obstructions, and node failures caused by power depletion. In the post deployment scenarios, nodes deployed over certain region may be destroyed due to intrusion, explosion or environmental factors like heat, vibration, failure of electronic components or software bugs. In another scenario, power sources of the nodes may lead death of the nodes, thereby affecting the coverage of the original network. Hence, holes are hardly avoided in wireless sensor networks. On the other hand, holes are important indicators of the general health of a sensor network. The presence of holes in the underlying geometric environment could have important consequences on the performance (Ahmed et al., 2005) of the sensor network at many levels.

For perception applications such as object tracking, environmental monitoring, and military surveillance, the networks require sufficient coverage over the region of interest (Watfa and Commuri, 2006b). Besides, understanding the global geometry and topology of the sensor field could have important implications for the design of several basic network functionalities such as routing and data gathering mechanisms. For instance, the presence of holes changes the topology of the networks and create a communication void that have adverse effect on routing algorithms. Hence, ignoring detection of holes in the network can affect the efficiency of the geographic routing and excessive energy consumption of hole boundary nodes. Additionally, for information flow, the hole could also affect the overall capacity of the network. Therefore, detection of coverage holes in the wireless sensor networks is primarily important as its presence often have physical correspondence and may map to one of the special events that are being monitored by the sensor networks.

Algorithms for detecting various coverage holes in WSNs can generally be classified into computational geometry approach, statistical approach, and topological approach. The computational geometry method (Ganeriwal et al., 2004) uses the coordinates of

---

[*] Corresponding author.
E-mail addresses: richma@cute.edu.tw (H.-C. Ma), pksahooind@yahoo.com (P. Kumar Sahoo), ywchen@ce.ncu.edu.tw (Y.-W. Chen).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.06.007

the sensor nodes and standard geometric tools to determine the coverage characteristics of the network. One feature of this approach is that the precise geometry of the domain and exact location of the nodes must be available. In the statistical approach (Liu et al., 2005), it assumes a randomly and uniformly distributed collection of sensor nodes. In this approach, the nodes that encircle the holes should have much lower average degrees than that of other nodes in the interior of the networks. That is, with sufficient high density, it usually exhibits bi-modal behavior and thus can be used to detect the holes. The drawback of these probabilistic approaches is the need for dense and uniformly distribution of sensor nodes.

In this paper, a computational geometry approach based distributed coverage hole detection protocol is designed for the self organized WSN to detect the coverage holes remotely. The current coverage hole detection is mainly based on the calculation and not on prediction in which presence of holes can be known locally. The rest of the paper is organized as follows. Motivations and related works of the coverage hole detection problems are reviewed in Section 2. The basic concepts of our proposed protocol with few definitions are formulated in Section 3. The hole detection protocol with theoretical analysis is presented in Section 4. Performance evaluation of the proposed algorithm is done in Section 5 and concluding remarks are made in Section 6.

2. Related work

In wireless sensor networks, determination of hole is of prime importance and several authors have proposed different algorithms too detect the coverage holes. The authors (Wang et al., 2004) discuss the detection of coverage holes based on the Voronoi graph, and develop algorithms to make the sensor networks as uniform as possible. A grid-based system model (Wang et al., 2005) is proposed to detect the coverage hole, where the grid head has to judge the coverage holes. The authors (Wan and Yi, 2006), analyze the probability of coverage degree with the number or sensing radius of the sensors, which are deployed randomly. The authors (Silva and Ghrist, 2007; Ghrist and Muhammad, 2005), introduce a homological method for detecting the coverage holes. Though, the method is useful for detecting holes, there is no analysis for the computational complexity and the method is centralized one. The authors develop a hole detection algorithm (Funke, 2005) based on the topology of the communication graph. They introduce simple algorithms with connectivity information of the nodes to construct iso-counters based on hop count from a root node and to identify where the contours are broken.

The output from this algorithm can find the sensors beside the holes under the unit-disk graph assumption and sufficient sensor density. The hole detection methods proposed in Meguerdichian et al. (2001) evaluate if the monitoring areas are well or poorly covered. The main feature of the topological methods (Funke and Klen, 2006; Wang et al., 2006) is based on the network topology or connectivity information of the nodes to identify the holes. These methods are attractive particularly for a large scale of sensor network in which the location information is not available. Further, node is marked as a corresponding sensor or boundary node depending on the distance between actual geometry boundary (Funke and Klen, 2006). Though the algorithm is simple, it cannot show how nodes are connected in a meaningful way. A simple and distributed so called TBR algorithm (Wang et al., 2006) is introduced to detect the nodes besides coverage holes that further connects them into meaningful boundary cycles. They construct the tree structure and further form the shortest path. However, the cost of their method is higher as a large number of control packets are needed to execute the algorithm. The authors propose algorithms (Hsieh and Sheu, 2009) that identify the boundary nodes surrounding the coverage holes of wireless sensor networks. However, their algorithm cannot detect the presence of holes in the WSN.

In order to maximize the WSN lifetime, the authors propose a hole detection protocol (Soreanu et al., 2008) that considers heuristics methods inspired from the jammer's possible movements. However, detection of holes using mobile nodes will degrade the lifetime of the sensor networks, as mobility of nodes can consume more energy. An anchor node based virtual modeling of holes (Yu et al., 2008) is proposed to solve the hole problems faced due to geographic routing. The authors design virtual circles, which can exactly cover a hole geometrically and information about the virtual circle is disseminated to all hole boundary nodes. However, in this protocol the control packets overhead is higher and nature of the holes cannot be predicted. A deterministic method for boundary node detection (Zhang et al., 2006) based on localized Voronoi polygons is proposed that uses the technique from the computational geometry. The proposed algorithm is localized one and requires one-hop neighbors information to detect the holes. Though the scheme uses localized Voronoi polygon to detect the boundary nodes, it cannot compute the nature of the holes and presence of holes beyond a single sensor.

An energy-efficient approach so called distributed hole coverage (DHC) algorithm (Watfa and Commuri, 2006a) is introduced to detect the coverage holes using coverage of the sensing circles and by identifying nodes besides the coverage holes. However, the method cannot detect all sensors which are besides the coverage holes. A distributed so called path density (PS) algorithm (Corke et al., 2007) is developed to detect the coverage holes in WSNs. It uses the path density to detect the holes by the neighbors of a dead sensor. The PS algorithm can detect coverage holes remotely, but requires more time and power consumption for detecting holes in practice. The hole detection algorithms (Watfa and Commuri, 2006a; Corke et al., 2007); require fully connectivity of the nodes in the network. Their work fails if connectivity is interrupted or lost somewhere. Besides, their hole detection method cannot use the communication graph or connectivity information to detect the coverage holes. PS algorithm uses the density of each path from the same node to decide which path leads to the coverage hole, and detects the hole based on this information. This algorithm uses two times flooding over the whole network to define the coverage hole.

A k-coverage verification scheme (Bejerano, 2008) for a target field is proposed, which requires a predefined value of k. According to this scheme, each node has only localized distance information of the distance between adjacent nodes in its vicinity and their sensing radius. Besides, an upper-bounded sensing radius and lower-bounded transmission radius of each node is considered. As per the proposed scheme, this could be achieved by configuring the nodes before placing them or by walking along the target field boundary with a hand held device. However, it is quite difficult due to random deployment nature of the sensors and geographical condition of the monitoring region. In this paper, we propose a computational geometry based distributed coverage hole detection protocol, in which only two-hop neighbors of a node can decide if any hole is present within their periphery and hole detection can be done without help of the sink. We compare our algorithm with DHC algorithm (Watfa and Commuri, 2006a), and path density (PS) algorithm (Corke et al., 2007), as they propose distributed hole detection methods.

2.1. Motivations

Normally, wireless sensors are deployed for the surveillance of a network and the mission of deployment cannot be accomplished if any coverage hole exists in the network. Hence, it is highly essential to detect the coverage holes time to time as sensors may fail due to technical snags or energy exhaustion of the nodes and, therefore, may create holes within the network. Though, several works propose the coverage and connectivity maintenance protocols, to the best of our knowledge very few algorithms have proposed to detect the coverage holes. Moreover, most of those protocols consider a regular monitoring region, which is not a real life scenario. Since, sensors are deployed randomly, getting an irregular deployment region is very practical. Hence, we propose the hole detection protocol for an irregular monitoring region. Besides, most of the state-of-art protocols consider that the communication range of a sensor is twice or greater than twice of its sensing range. However, communication is normally the main source of energy consumption and reduction in communication range can improve the overall network lifetime. Therefore, the coverage hole detection protocol proposed in this work intends not only to find out the presence of holes in the network, but also to minimize the energy consumption of each node by considering the communication range of a node is equal to its sensing range. Besides, the proposed protocol takes help of two-hop neighbors of a node to detect the hole around it and, therefore, requires less computation time.

3. Problem formulation

Consider a wireless sensor networks, where nodes are deployed randomly over the monitoring region such that some part of the network has sufficient coverage due to the presence of several redundant nodes, whereas other parts have coverage holes due to absence of any sensor. As soon as the network is formed, each node knows its location information and collects its one and two-hop neighbors list. The system model is considered to be a multi-hop wireless network with nodes having omni-directional transceivers. The sensing range (R_s) is equal to the communication range (R_c) and each node knows its location information via GPS or any location information system. Each node collects its two-hop neighbor's location information as soon as the deployment is over, i.e. each node knows the location of its neighbors within the range of $2R_c$.

3.1. Definitions

In this subsection definition of few related terms are given that are used in the hole detection algorithms.

Definition 1. *Sensing range*: Sensing range of a node is the circular disk of radius R_s, which is centered at its location. As shown in Fig. 1(a), sensing range of node A is represented by the circumference of the circle centered at A. Any object present within the sensing range is perfectly detected by the sensor.

Definition 2. *Communication range*: Communication range of a node is the circular disk of radius R_c, which is centered at its location. As shown in Fig. 1(a), communication range of node A is represented by the circumference of the circle centered at A. Throughput our work, $R_c = R_s$ is considered.

Definition 3. *Reference node* (*RN*): A source node that initiates to execute the hole detection algorithm is called a reference node (RN). It is to be noted that a reference node first collects its neighbors information located within $2R_c$ and executes the hole detection algorithm. For example, as shown in Fig. 1(a), if A is a reference node, first it initiates the hole detection procedure.

Definition 4. *Neighbor*: If A and B are any two nodes such that distance between them, i.e. $(d_{AB}) \le R_c$, then A and B are one-hop neighbors to each other. However, if $R_c < d_{AB} \le 2R_c$, A and B are two-hop neighbors to each other. Throughout our work, either a one-hop or two-hop neighbor is referred to as a neighbor of the reference node. As shown in Fig. 1(a), if A is a reference node, node B is the one-hop neighbor of A as it is located within R_c of A, whereas C, D and E are two-hop neighbors of A as they are located within $2R_c$ of A.

Definition 5. *Circum radius* (R): Radius of the circum circle formed by location of any three sensors as the vertices of a triangle is called circum radius R. As shown in Fig. 1(b), $\overline{AZ}$ is the circum radius of the triangle ABC, which is denoted by R. If a, b and c are length of three sides of the triangle ABC and Δ is area of that triangle, circum radius $(R) = abc/4\Delta$. Since each node knows its location information, length of each sides a, b, c and area of the triangle Δ can be found out.

Definition 6. *Circum center* (Z): Center of the circum circle formed by location of any three sensors as the vertices of a triangle is called circum center. As shown in Fig. 1(b), Z represents the circum center of the triangle ABC formed by the sensors located at A, B and C. If (x_1,y_1), (x_2,y_2) and (x_3,y_3) are locations of sensors A, B and C, respectively, then (x_0,y_0), coordinate of Z could be found out by solving the linear equations $x_0(x_2-x_1)+y_0(y_2-y_1)+k_1 = 0$ and $x_0(x_3-x_2)+y_0(y_3-y_1)+k_2 = 0$, where k_1 and k_2 are constants of the linear equations.

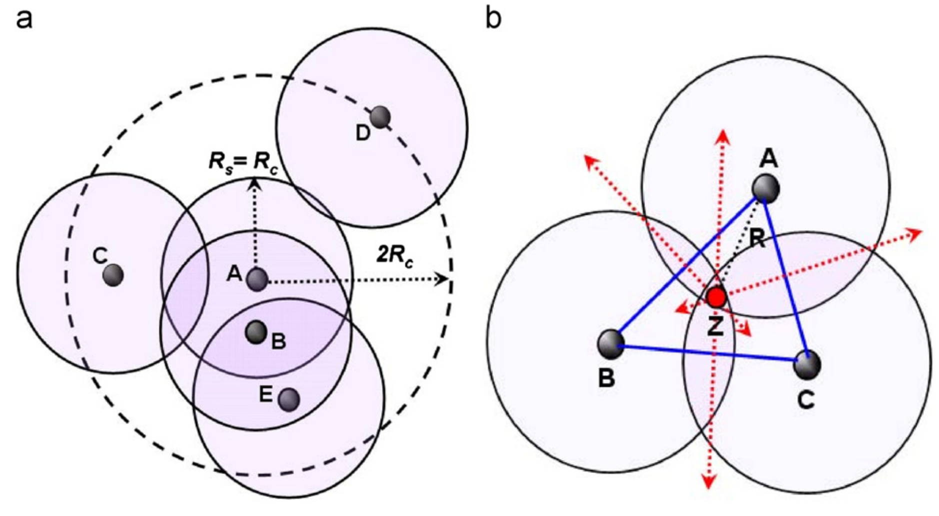

Fig. 1. Example to demonstrate various definitions. (a) R_c, R_s, reference node and neighbors. (b) Circum radius (R) and circum center (Z).

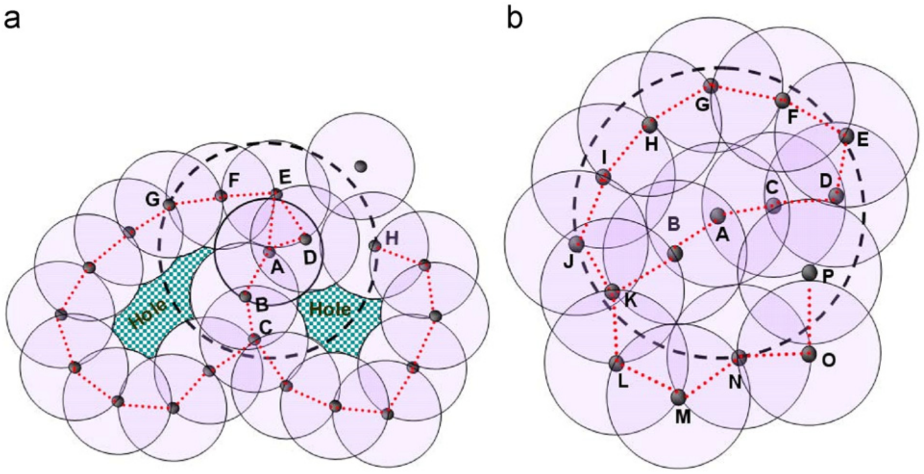

Fig. 2. Example of connectivity with or without holes in the network. (a) With presence of holes. (b) Without presence of holes.

3.2. System model

In our system model, it is assumed that there are multiple coverage holes in the monitoring region and the network is connected. Therefore, the one-hop and two-hop neighbors of a node must be connected with it through one and multi-hops, respectively. For example, as shown in Fig. 2(a), if *A* is considered as a reference node, it is connected with other nodes of the network with help of its one and two-hop neighbors though coverage hole exists in the network. In fact, *B* and *D* are one-hop neighbors of *A*. Hence, *A* is connected to *C* and *E* through its one-hop neighbors *B* and *D*, respectively. Besides, *C*, *E*, *F* and *G* are two-hop neighbors of *A* as they are within its $2R_c$. Though, *A* is not connected with *G* and *F* through its one-hop neighbors *B* and *D*, it is connected with them through its two-hop neighbor *E*, which is connected to *D*. Hence, the whole network is connected though there are coverage holes in it.

As shown in Fig. 2(b), if *A* is considered as a reference node, no coverage hole is found around it, which is connected with its one-hop neighbors *B* and *C*. Besides, some of its two-hop neighbors are connected with *A* through its three-hop neighbors. For example, node *D* is a two-hop neighbor of *A* and is connected through its one-hop neighbor *B* with help of its two and three-hop neighbors. Hence, in our system model, the whole network is fully connected though coverage hole may exist in some part of the network. However, it is assumed that no isolated sensor is seen in the network as the whole network is fully connected. The dotted circle in both figures indicates twice of the communication range. Since, $R_c = R_s$ is considered in our protocol, radius of the dotted circle is equal to $2R_c$, within which only one and two-hop neighbors of *A* are located. As per our proposed system model each node must have at least one 1-hop neighbor, so that it can know its one and two-hop neighbors information. As shown in Fig. 2(a), node *H* is two-hop neighbor of *A*. *A* finds information about *H* through its immediate one-hop neighbor *B*, which is connected to *H* through *C* and some intermediate nodes.

4. Distributed hole detection protocol

In this section, we propose a self organized hole detection protocol that detects the coverage holes irrespective of any shape or size of the monitoring region. In this proposed hole detection protocol, it is assumed that each sensor knows location information of its one and two-hop neighbors as soon as the deployment procedure is over. The hole detection algorithm is executed by any sensor, which is termed as a reference node. Prior to executing the hole detection algorithm each node undergoes the neighbor discovery procedure as described in the following subsection.

4.1. Neighbor discovery phase

In this phase, a reference node is selected randomly from any part of the deployed region, which undergoes the neighbor discovery phase. The reference node broadcasts $HELLO_1$ message that contains its location information. Upon receiving the hello message, a node calculates the distance d_1 between the reference node and itself. If $d_1 \leq R_c$, the node sets itself as one-hop neighbor of the reference node *X* and unicasts its location information and ID to the reference node *X*. In the next step, each one-hop neighbors of the reference node *X*, broadcasts the $HELLO_2$ message that contains the location information of the reference node *X*. Upon receiving the $HELLO_2$ message, each receiver calculates its physical distance d_2 from the reference node *X*. If $d_2 \leqslant 2R_c$, the sensor sets itself as a two-hop neighbor of reference node *X* and unicasts its location information and ID to the reference node *X* through its sender. Ultimately, reference node *X* records the location information and ID of each of its one and two-hop neighbors. This procedure is executed by each node of the network in a distributed manner. Eventually, at the end of the neighbor discovery phase each node knows its one and two-hops neighbors list.

For example, as shown in Fig. 3, if node *X* executes the neighbor discovery phase, it broadcasts $HELLO_1$ message, which is received by the nodes *A*, *B*, *C* and *D*. Since, the distance between them and $X \leq R_c$, *A*, *B*, *C* and *D* set themselves as one-hop neighbor of *X*, which is later informed to *X*. Then, *A*, *B*, *C* and *D* broadcast the $HELLO_2$ message that contains the location information of node *X*. Upon receiving the $HELLO_2$ message, nodes *P*, *Q*, *R*, *T*, *M* and *N* set themselves as the two-hop neighbors of *X* as their distance from *X* is $\leq 2R_c$. Finally, *P*, *Q*, *R*, *T*, *M* and *N* broadcast the $HELLO_2$ message that contains location information of node *X*. Since, the distance of node *S* and *O* from node *X* is $\leq 2R_c$, they set themselves as the two-hop neighbors of *X*. Thus, *X* can find its one-hop neighbor's set that contains *A*, *B*, *C* and *D* and two-hop neighbor's set that contains *P*, *Q*, *R*, *S*, *T*, *M*, *N* and *O*.

4.2. Hole detection phase

As soon as each node gets its neighbor list, the hole detection phase is executed distributively. Any node can initiate the hole

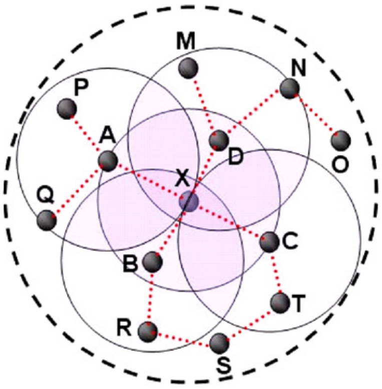

Fig. 3. Example of neighbor discovery procedure by a reference node X.

detection procedure, which can be referred to as a reference node. At the time of execution of hole detection phase, it is assumed that the circum radius (R) and circum center (Z) of a triangle are calculated from the location information of the reference node and its one pair of neighbors, which are self organized by the nodes. The hole detection phase is presented in Algorithm 1 and its correctness is verified analytically in the next subsection.

Algorithm 1. Hole Detection Algorithm

```
Notation:
$R_s$: Sensing range of a node;
$X$: Reference node that initiates the hole detection algorithm;
$N$: Set of neighbors (one and two hop) of $X$;
$(\alpha,\beta)$: Location of the reference node $X$;
$N_u$: Set of neighbors whose y-coordinate $\geq \beta$;
$N_d$: Set of neighbors whose y-coordinate $< \beta$;
Input:
1. Location of $X$ and any two of its neighbors;
Let it be $(x_1,y_1)$, $(x_2,y_2)$ and $(x_3,y_3)$;
2. Length of each sides of the triangle $a$, $b$ and $c$;
3. Area of the triangle $\triangle$;
4. Circum radius $R=abc/4\triangle$;
5. Circum center $Z$;
6. Each angle of the triangle $ABC$;
Let it be $A=\arcsin(a/2R)$, $B=\arcsin(b/2R)$, and
$C=\arcsin(c/2R)$;
Algorithm:
Step 1: Select any node $X$ randomly as a reference node;
Step 2: Find one and two-hop neighbors of $X$;
Assign those nodes to set $N$;
Step 3: Select nodes from set $N$ whose y-coordinate $\geq \beta$;
Assign those nodes to set $N_u$;
Step 4: Arrange nodes of $N_u$ with their x-coordinate in
ascending order and put them in a new set $N_ux$,
such that $N_ux=\{A_i,A_j/\forall A_i,A_j \in N_ux$, x-coordinate of
$A_i <$ x-coordinate of $A_j\}$;
Step 5: Select nodes from set $N$ whose y-coordinate $< \beta$;
Assign those nodes to set $N_d$;
Step 6: Arrange nodes of $N_d$ with their x-coordinate in
descending order and put them in a new set $N_dx$,
such that $N_dx=\{A_i,A_j/\forall A_i,A_j \in N_dx$, x-coordinate of
$A_i \geq$ x-coordinate of $A_j\}$;
Step 7: Select 1st two nodes $A_i$ and $A_j$ from $N_ux$ such that
x-coordinate of $A_i < A_j$;
do
{
Step 8: Compute circum radius $R$ and circum center $Z$ of
triangle $XA_iA_j$;
Step 9: Verify if $XA_iA_j$ is an acute or obtuse triangle;
Step 10: If ($X$ forms an acute triangle with its neighbors $A_i$ and $A_j$)
{
If $(R \leq R_s)$
No hole exists around the reference node $X$;
else
There exists a hole around the reference node $X$;
}
Step 11: If ($X$ forms an obtuse triangle with its neighbors $A_i$ and $A_j$)
{
If $(R \leq R_s)$
No hole exists around the reference node $X$;
else
Check if circum center $Z$ is covered by any other sensor
Step 12: If $Z$ is covered by a sensor
No hole exists around the reference node $X$;
else
There exists a hole around the reference node $X$;
}
Step 13: Update $N_ux \leftarrow N_ux-\{A_i\}$;
} while($N_ux \neq 1$);
Step 14: Choose the 1st node $A_i$ of $N_dx$ and last balance node $A_j$
of $N_ux$;
Step 15: Execute the procedures from Step 8 through 12 for
the nodes of $N_dx$;
Step 16: Update $N_dx \leftarrow N_dx-\{A_i\}$;
Step 17: Continue the procedure until $N_dx \neq \phi$;
```

The explanation of the hole detection algorithm is shown in Fig. 4. As shown in Fig. 4(a), let X be a reference node, whose one and two-hop neighbors set $N=\{A,B,C,D,E,F,G,H,J,K,L,M\}$. As per step 4 of the algorithm, set $N_u=\{A,B,C,D,E\}$ and arranging those nodes in the ascending order of their x-coordinate, N_{ux} becomes $\{B,A,D,C,E\}$. Similarly, as per step 6 of the algorithm, set $N_d=\{F,G,H,J,K,L,M\}$ and arranging those nodes in the descending order of their x-coordinate, N_{dx} becomes $\{G,F,H,J,K,L,M\}$. According to step 7 of the algorithm, node B and A should be selected from the set N_{ux}, and X forms an obtuse triangle with node B and A. However, the circum radius of ΔXBA is $>$ R_s and its circum center must be covered either by A, C or D. Hence, there is no hole exist around nodes B and A. Then, node A and D are selected from the set N_{ux}, and X forms an obtuse triangle with node B and D. However, the circum center of ΔXBD must be covered either by B, C or D. Hence, no hole exists around B and D. Continuing this process with each pair of nodes of N_{ux} and forming a triangle with X, it can be verified that no hole exists within sensors A –E.

Now, based on step 15 of the algorithm, node E from set N_{ux} and node G from set N_{dx} should be taken to form a triangle with reference node X. Thus, an acute triangle XEG is formed, whose circum radius $R \leq R_s$. Hence, as per our algorithm, no hole exists around sensors E and G. Thus, continuing steps 16 and 17 of the algorithm, it could be verified that no coverage hole exists within them. Similarly, as shown in Fig. 4(b), let X be a reference node, whose one and two-hop neighbors set $N=\{A,B,E,G,J,L,M\}$. As per step 4 of the algorithm, set $N_u=\{B,E,A\}$ and arranging those nodes in ascending order of their x-coordinate, N_{ux} becomes $\{B,A,E\}$. It is to observed that triangle formed by XAB is obtuse one whose circum radius $R>R_s$. Since, its circum center Z may be covered either by node B or X, there is no coverage hole within them. However, the circum center Z of ΔXBE is not covered by any node and, therefore, a coverage hole exists between node B and E. Based on step 15 and the subsequent steps of the algorithm, it can be verified that no coverage hole exists within them, as the triangle formed with X is either acute or its circum center Z is covered by either of the nodes present in the network.

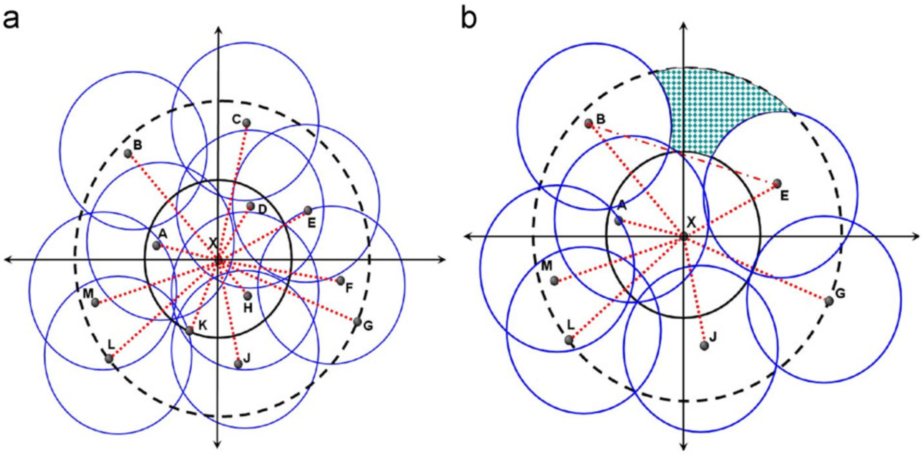

Fig. 4. Example showing coverage hole detection by a reference node X. (a) When no hole is existed. (b) When hole is existed.

It is to be noted that each node of the network is a reference node and, therefore, each node executes the hole detection algorithm with its one and two-hop neighbors distributively. Hence, the presence or absence of coverage holes can be detected with help of any pair of neighbors of the reference node. Besides, this proposed algorithm can detect the coverage hole irrespective of any shape or size of the monitoring region.

4.3. Theoretical analysis

As described in the hole detection algorithm, each sensor forms a triangle either with one pair of its one-hop or two-hop neighbors or with one node from its one-hop and another node from its two-hop neighbors. As given in the hole detection algorithm, the presence or absence of the hole depends on the nature of the triangle. In order to justify the correctness of the algorithm analytically, some lemmas are proposed in this subsection as follows.

Axiom 1. The triangle (acute, right or obtuse) formed by a reference node with any pair of its neighbors (one-hop or two-hop) must be enclosed within the effective sensing range of those three nodes.

Axiom 2. The circum radius (R) of the triangle (acute, right or obtuse) formed by a reference node with any pair of its neighbors (one-hop or two-hop) must be either $\leq R_s$ or $> R_s$.

Axiom 3. The circum center (Z) of the triangle (acute, right or obtuse) formed by a reference node with any pair of its neighbors (one-hop or two-hop) must be located inside or outside the sensing range of those three sensors.

Lemma 1. *If an acute triangle is formed by a reference node with its one-hop neighbors, then no coverage hole exists within those three sensors.*

Proof. Let an acute triangle be formed by a reference node A with its one pair of one-hop neighbors as shown in Fig. 5.

The maximum acute angle of that triangle must be $\leq \pi/2$.

$\Rightarrow$ Circum center Z must be located at most on the one side of that triangle.
$\Rightarrow$ Circum radius $R < R_s$ and circum center Z must be covered by those nodes.
$\Rightarrow$ There exists common sensing region as $R < R_s$
$\Rightarrow$ No coverage hole exists within those nodes.

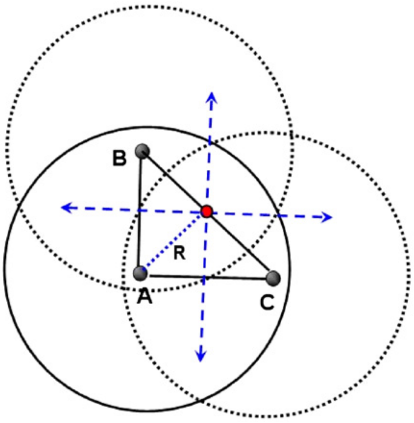

Fig. 5. Figure demonstrating Lemma 1: Acute triangle is formed by reference node A with its one-hop neighbors B and C.

Lemma 2. *If an obtuse triangle is formed by a reference node with its one-hop neighbors such that its circum radius $R \leq R_s$, then no hole exists within those sensors.*

Proof. Let an obtuse triangle be formed a reference node A with its one pair of one-hop neighbors as shown in Fig. 6 and $R \leq R_s$

$\Rightarrow$ Circum radius R must be within sensing disk of any sensor and circum center Z must be covered by those nodes.
$\Rightarrow$ No coverage hole exists within those nodes.

Lemma 3. *If an obtuse triangle is formed by a reference node with its one-hop neighbors such that its circum radius $R > R_s$, and circum center (Z) is not covered by any of its neighbors, then there must be a hole besides those sensors.*

Proof. Let an obtuse triangle be formed by a reference node A with its one pair of one-hop neighbors as shown in Fig. 7 and $R > R_s$

$\Rightarrow$ Circum radius R and circum center Z must be outside the sensing disk of those sensors.
$\Rightarrow$ Hole must be existed around those sensors, if Z is not covered by any other sensor.

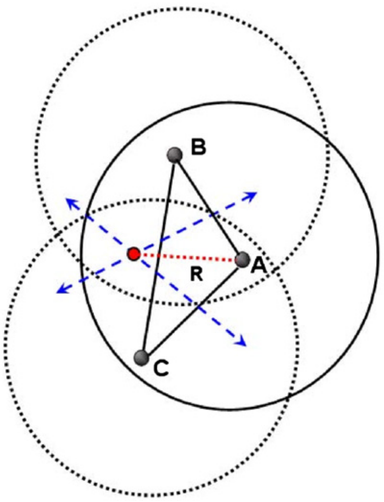

Fig. 6. Figure demonstrating Lemma 2: Obtuse triangle is formed by reference node *A* with its one-hop neighbors *B* and *C*.

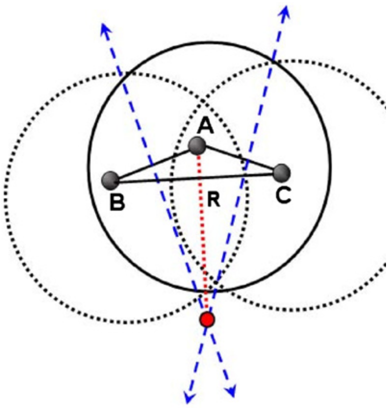

Fig. 7. Figure demonstrating Lemma 3: Obtuse triangle is formed by reference node *A* with its one-hop neighbors *B* and *C* and $R > R_s$.

Lemma 4. *If an acute triangle is formed by a reference node with its two-hop neighbors and its circum radius* $R > R_s$, *then there must be a coverage hole within those sensors, otherwise no coverage hole exists.*

Proof. Let an acute triangle be formed by a reference node *A* with its one pair of two-hop neighbors as shown in Fig. 8.

By the method of contradiction, let us assume that circum radius $R \le R_s$

- ⇒ There must be a common sensing region within those three sensors.
- ⇒ No hole exists within those three sensors.

However, if no common sensing region exists

- ⇒ $R > R_s$
- ⇒ Circum center *Z* must be located outside the sensing region of those three sensors.
- ⇒ There exists a hole within those three sensors.

Lemma 5. *If an obtuse triangle is formed by a reference node with its two-hop neighbors such that the angle subtended at the reference node is acute and* $R \le R_s$, *then no hole exists within them, otherwise a hole must exists if circum center Z is not covered by any other sensor.*

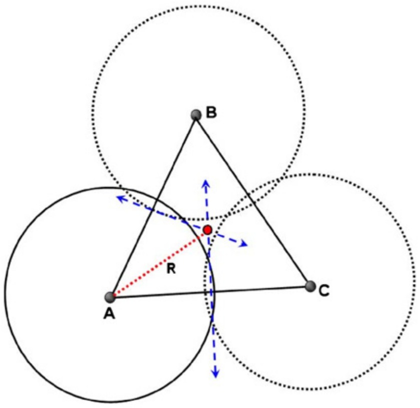

Fig. 8. Figure demonstrating Lemma 4: Acute triangle is formed by reference node *A* with its two-hop neighbors *B* and *C*.

Proof. Let an obtuse triangle be formed by a reference node *A* with its one pair of two-hop neighbors as shown in Fig. 9(a) and (b), where angle at the reference node is acute.

If $R \le R_s$ is true,

- ⇒ Circum center *Z* must be located inside the sensing region of those three sensors.
- ⇒ No hole exists within those three sensors.

Otherwise, if $R > R_s$ is true,

- ⇒ Circum center *Z* must be located outside the sensing region of those three sensors.
- ⇒ Coverage hole exists if *Z* is not covered by any other neighbor of the reference node.

Lemma 6. *If an obtuse triangle is formed by a reference node with its two-hop neighbors and the angle subtended at the reference node is obtuse, coverage hole exists in between those two-hop neighbors.*

Proof. Let an obtuse triangle be formed by a reference node *A* with its two-hop neighbors as shown in Fig. 10.

Since the triangle is formed between a pair of immediate two-hop neighbors which are in ascending order of their *x*-coordinate,

- ⇒ No other node exists in between those two-hop neighbors of the reference node.

Besides, since the triangle is obtuse,

- ⇒ $R > R_s$
- ⇒ Coverage hole exists in between those two-hop neighbors.

Lemma 7. *If an acute triangle is formed by a reference node with one of its one-hop neighbor and another one with its two-hop neighbors, no hole exists if* $R \le R_s$, *otherwise, coverage hole exists within them.*

Proof. Let an acute triangle be formed by a reference node *A* with its one and two-hop neighbors as shown in Fig. 11.

If $R \le R_s$,

- ⇒ Circum center *Z* must be located inside the sensing region of those three sensors.

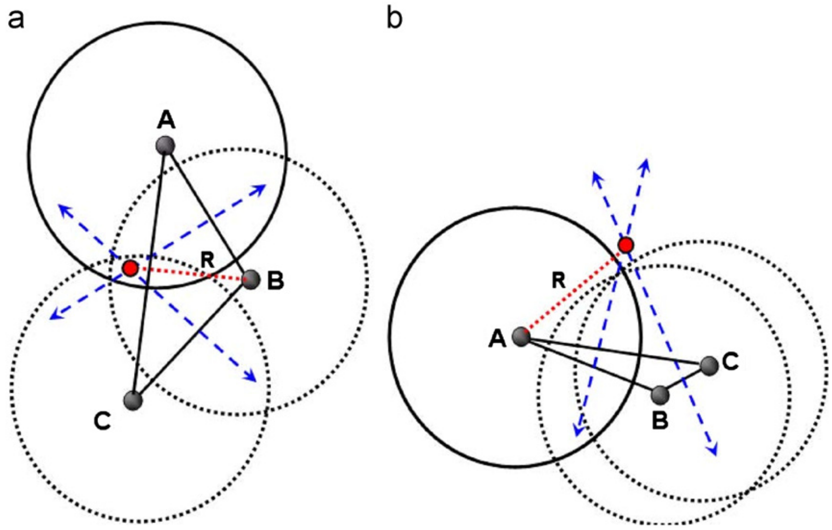

Fig. 9. Figure demonstrating Lemma 5: Obtuse triangle is formed by reference node *A* with its two-hop neighbors, but angle subtended at *A* is acute. (a) Neighbors when $R \leqslant R_s$. (b) Neighbors when $R > R_s$.

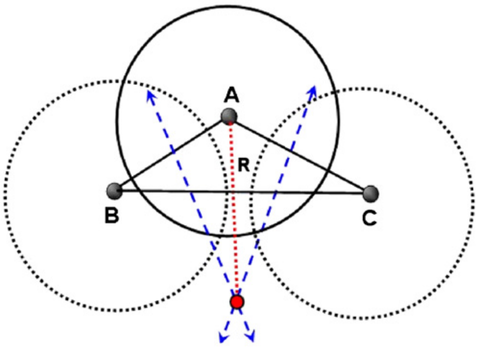

Fig. 10. Figure demonstrating Lemma 6: Obtuse triangle is formed by reference node *A* with its two-hop neighbors and the angle subtended at *A* is obtuse.

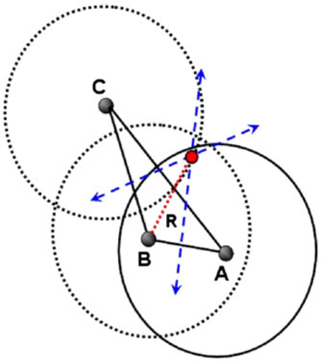

Fig. 12. Figure demonstrating Lemma 8: Obtuse triangle is formed by a reference node *A* with its one-hop neighbor *B* and its two-hop neighbor *C*.

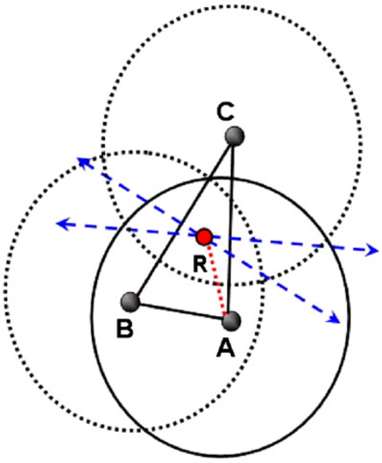

Fig. 11. Figure demonstrating Lemma 7: Acute triangle be formed by reference node *A* with its one-hop neighbor *B* and its two-hop neighbor *C*.

⇒ No hole exists within those three sensors.

However, if $R > R_s$

⇒ Coverage hole exists in between those neighbors, if circum center *Z* is not covered by any other neighbor of the reference node.

Lemma 8. *If an obtuse triangle is formed by a reference node with one of its one-hop neighbor and another one with its two-hop neighbors, such that angle subtended at the reference node is acute, no hole exists within them if* $R \leq R_s$*, otherwise a hole must exist if circum center Z is not covered by any other sensor.*

Proof. Let an obtuse triangle be formed by a reference node *A* with its one and two-hop neighbors such that angle subtended at the reference node is acute as shown in Fig. 12.

Proof of this lemma is same as Lemma 5.

Lemma 9. *If the triangle formed by a reference node with one of its one-hop neighbor and another one with its two-hop neighbors subtends an obtuse angle at the reference node, hole exists within those neighbors.*

Proof. Let an obtuse triangle be formed by a reference node *A* with its one and two-hop neighbors such that angle subtended at the reference node is obtuse as shown in Fig. 13.

Proof of this lemma is same as Lemma 6.

Theorem 1. *Coverage hole may or may not exist in the network, if an acute triangle is formed by a reference node with its neighbors.*

Proof. As proved in Lemma 1, no coverage hole exists if an acute triangle is formed by a reference node with its one-hop neighbors.

As proved in Lemma 4 and 7, no coverage hole exists if acute triangle is formed with its 2-hop neighbors or with one of its one-hop neighbor and another with its 2-hop neighbors, when $R \leq R_S$.

However, as proved in second part of Lemma 4 and 7, coverage hole exists, if acute triangle is formed by a reference node with its 2-hop neighbors, when $R > R_S$. From the Proofs of Lemma 1, 4 and 7, Theorem 1 can be proved easily.

Theorem 2. *Coverage hole may or may not exist in the network, if an obtuse triangle is formed by a reference node with its neighbors.*

Proof. When $R \leq R_S$, no hole exists in the network if an obtuse triangle is formed by a reference node with its one-hop neighbors, as proved in Lemma 2.

Similarly, no coverage hole exists if obtuse triangle is formed with one of its one-hop neighbor and another with its 2-hop neighbors, as shown in Lemma 8.

However, if $R > R_S$ and obtuse triangle is formed by a reference node with its 2-hop neighbors, coverage hole exists in the network as proved in Lemma 3. From the Proofs of Lemma 2, 3 and 8, Proof of Theorem 2 can be concluded.

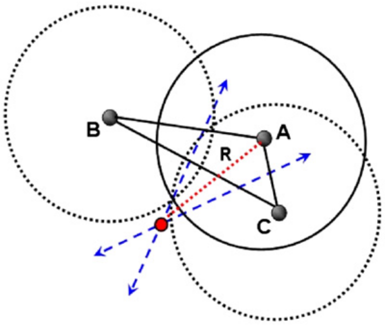

Fig. 13. Figure demonstrating Lemma 9: Obtuse angle subtended at the reference node *A*, which forms a triangle with its one-hop neighbor *C* and its two-hop neighbor *B*.

Theorem 3. *Presence or absence of coverage hole in the network depends on the nature of angle formed by a reference node with its neighbors.*

Proof. If an acute angle is subtended at the reference node and $R \leqslant R_S$, no hole exists within those sensors, as proved in Lemma 5.

However, if an obtuse angle is subtended at the reference node either with the 2-hop neighbors or with one of the one-hop and another with 2-hop neighbor of a reference node, coverage hole exists within those neighbors, which is proved in Lemma 6 and 9. Hence, Proof of Theorem 3 can be completed.

5. Performance evaluation

In this section, performance of our hole detection protocol is evaluated by simulating the proposed algorithm for different number of nodes. Besides, in order to justify the contribution of our work in terms of simulation results, our algorithm is compared with similar hole detection protocols as given in the subsequent subsections.

5.1. Simulation setup

In order to find the performance evaluation of the proposed hole detection protocol, it is simulated using NS-2.33 for different number of nodes that are deployed randomly over an area of 500 m × 500 m. The number of deployed nodes varies from 200 to 2000. In order to get the accurate evaluation, each simulation is run for 30 rounds to get the average of each data. IEEE 802.15.4 MAC is considered as the channel access mechanism. For each sensor node, a fixed amount of 100 J initial reserved energy is assumed. The sensing range varies from 10 to 30 m with communication range is equal to the sensing range, i.e. $R_c = R_s$ and a homogenous network environment is considered in the simulation. Each sensor, initially broadcasts message limited within $2R_c$ to get the two-hop neighbor's information, which is compatible with our algorithm.

In the simulation, holes are generated randomly among the multi-hop and fully connected nodes so that they can form

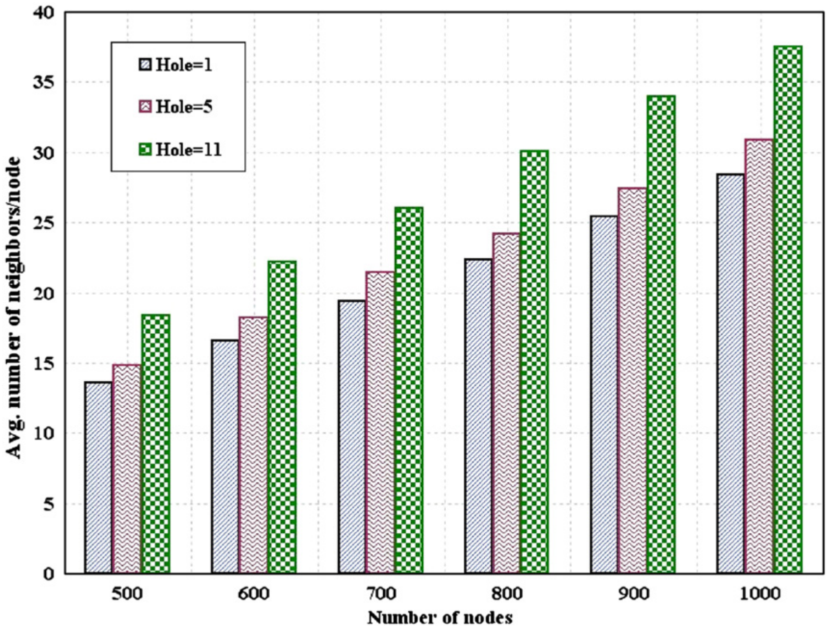

Fig. 14. Average number of neighbors per node for different number of deployed nodes.

different groups of nodes. In order to implement the neighbor discovery phase, all nodes are given location information to calculate the distance between itself and the nodes around it by uni-casting the control packets. Taking $R_c = 10$ m, the one and two-hop neighbor set of each node is found out. The traffic data rate is kept at 250 Kbps and control packets are sent in every 2 s to detect the neighbors, which is continued till 20 s to get the final list of the neighbors (one and two hops) of each node. The detail simulation results are given in the next subsection.

5.2. Simulation results

The simulation results for detecting coverage holes for different number of nodes and energy consumption due to hole detection is evaluated in this section. Besides, comparison of similar hole detection protocols with our proposed algorithm is also made in this section. In order to get practical insight of our hole detection schemes, we have simulated our protocol for different number of holes to find the average number of neighbors of each reference node, as shown in Fig. 14. It is found that more number of neighbors have to participate to execute the hole detection algorithm, if number of coverage holes are increased.

We consider two criteria to evaluate the performance of our algorithms. They are the average hole detection time and average power consumption for detecting the holes taking different number of nodes, average number of holes and average density of nodes, as shown in Figs. 15–18. As shown in Fig. 15, the average hole detection time increases with increase in number of nodes exponentially. Besides, the hole detection time is also increased with increase in average number of neighbors of each reference node. This exponential increase in hole detection time is due to formation of more triangles with more pair of nodes and thereby increase in the computation time. The simulation result of average hole detection time for different number of nodes with different values of communication range is presented in Fig. 16. It is observed that the hole detection time increases exponentially with increase in the communication range. Since, the communication range increases, a reference node can have more number of neighbors that increases the computation time. Hence, the overall hole detection time is increased due to increase in communication range R_c. From this observation, we can infer that the proposed algorithm is favorable for hole detection.

The average hole detection time for average number of holes present in the network is shown in Figs. 17 and 18. As shown in

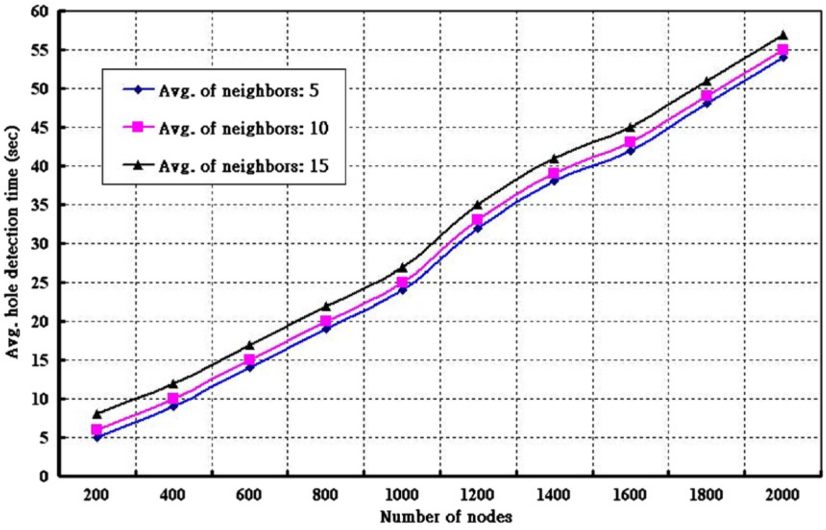

Fig. 15. Average hole detection time for different number of deployed nodes.

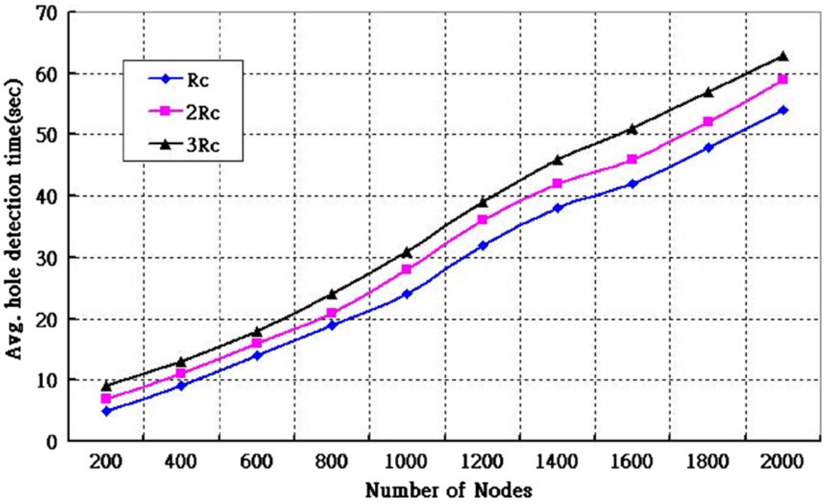

Fig. 16. Average hole detection time for different number of nodes with different sensing range.

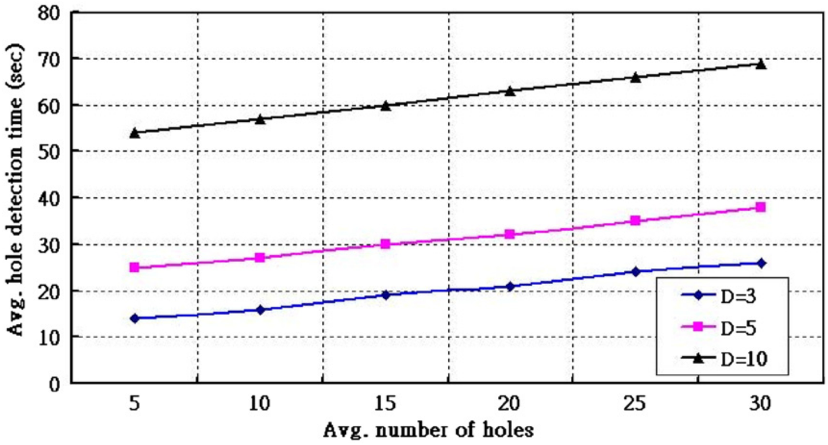

Fig. 17. Average hole detection time for different number of holes with different node density.

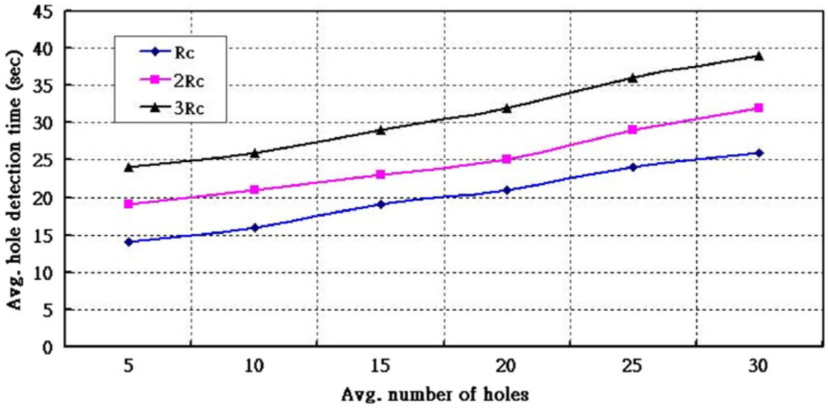

Fig. 18. Average hole detection time for different number of holes with different communication range.

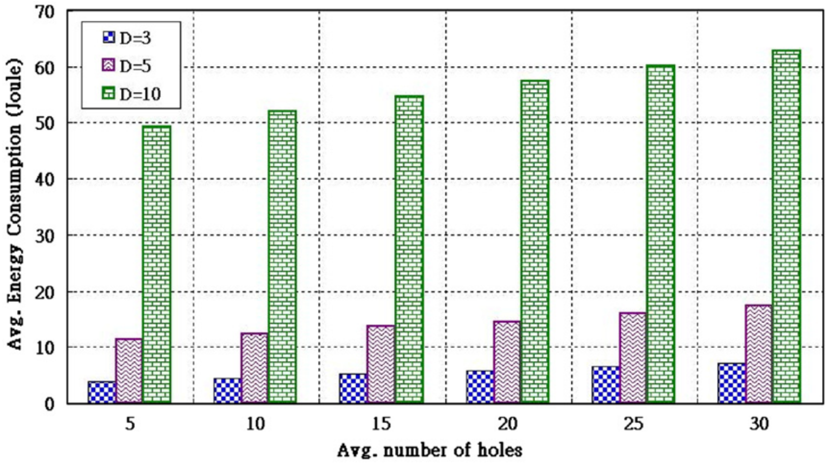

Fig. 19. Average energy consumption for different number of holes with different node density.

Fig. 17, average hole detection time is simulated for different density of nodes per square meters. It is observed that the average hole detection time increases linearly with increase in different number of holes as more holes requires more time to detect them. Besides, the hole detection time is also increased with increase in average node density of the network. This is due to exchange of more control packets among the densely deployed nodes. If there are n sensors in the network, where m is the average number of neighbors of each sensor and T represents average packet delivery time, then time cost for delivering each packet can be estimated as $(n \cdot m \cdot T)$. The average hole detection time for different number of holes with different values of communication range is simulated and the result is displayed in Fig. 18. Since, more nodes are attached to each reference node if communication range is

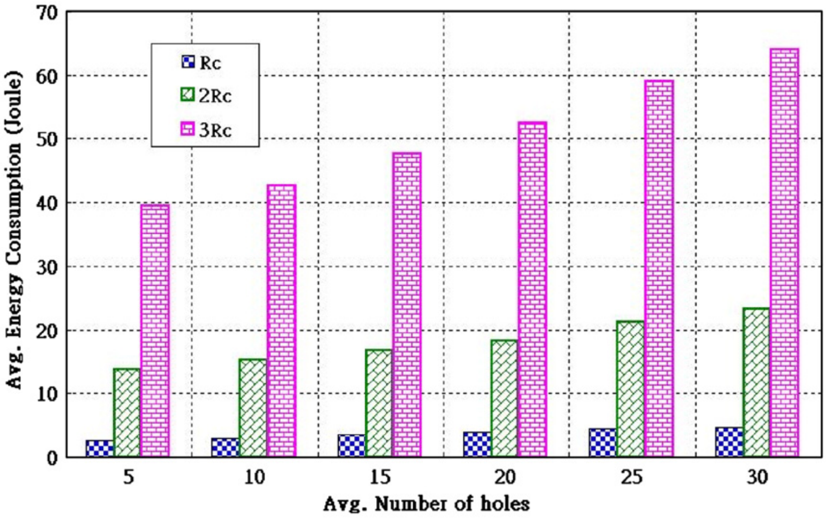

Fig. 20. Average energy consumption for different number of holes with different communication range.

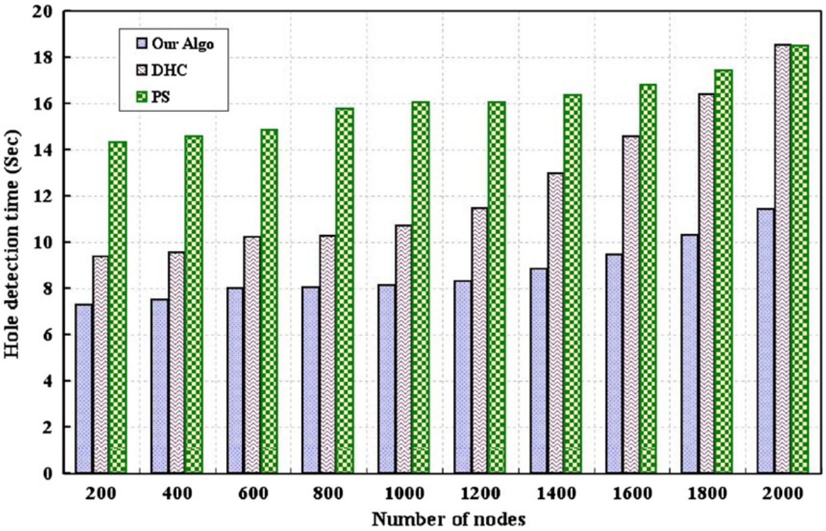

Fig. 21. Comparison of different protocols for average hole detection time.

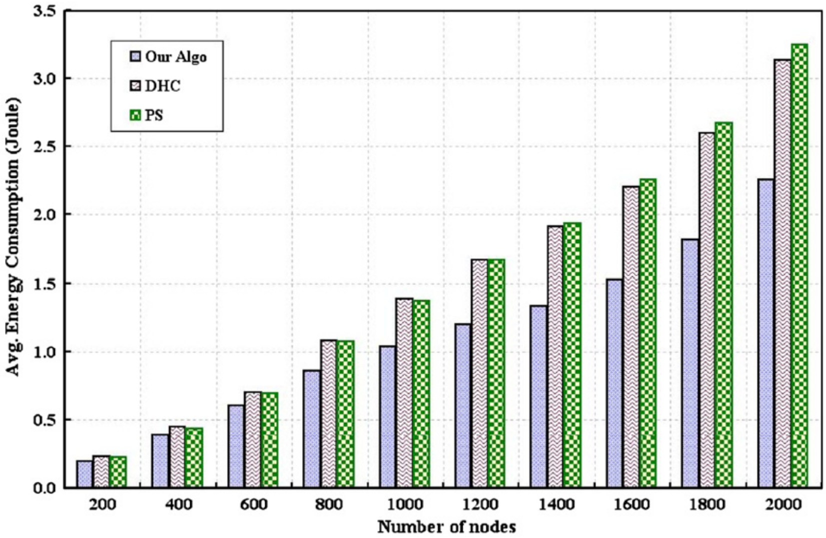

Fig. 22. Comparison of different protocols for average energy consumption.

increased, the hole detection time is also increased linearly. More neighbors coupled with more number of coverage holes obviously increase the average hole detection time as shown in Fig. 18.

The average energy consumption for different number of coverage holes with different density of nodes is given in Fig. 19. If P represents average power consumption for delivering each packet, P_{idle} is the average power consumption during idle period of the whole process and m is the average number neighbors of each sensor, then average power consumption for each sensor can be estimated as $(m \cdot P)+P_{idle}$. From Fig. 19, it is found that average power consumption for detecting the holes is increased with increase in number of holes. The average power consumption is also higher, if density of the nodes over the network is increased. Besides, average energy consumption for detecting different number of holes with different ranges of communication range is evaluated as shown in Fig. 20. It is observed that average energy consumption per node increases if communication range is increased. The energy consumption increases as communication requires more energy to forward the control packets in detecting holes and more neighbors of a reference node have to execute the hole detection procedure.

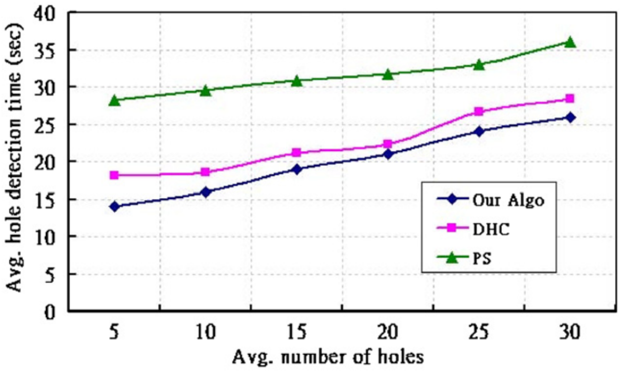

Fig. 23. Comparison of different protocols for average hole detection time with number of holes.

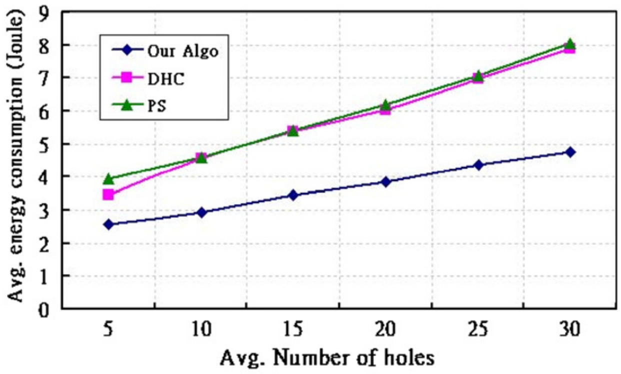

Fig. 24. Comparison of different protocols for average energy consumption with number of holes.

As shown in Fig. 21, the hole detection time of our algorithm outperforms over *DHC* and *PS*. As per *PS* algorithm, it uses broadcast mechanism to transmit the density messages, and returns the hole detection information through broadcasting. Hence, hole detection time of *PS* is worse than our algorithm. The distributed hole coverage (DHC) algorithm (Watfa and Commuri, 2006a) uses the notion of coverage of the sensing disk to detect the holes. However, we consider only three nodes to check the presence of a hole and can detect holes irrespective of shape or size of the monitoring region. Hence, our hole detection time is also better than *DHC*. The average energy consumption for detecting the holes for different number of nodes are presented in Fig. 22. It is observed that our algorithm outperforms over both *DHC* and *PS* for different number of nodes. For smaller number of nodes, though average energy consumption in our protocol is almost similar to *PS*, for higher number of nodes, our protocol outperforms over *DHC* and *PS*. Since, sensor network is suitable for the higher node density, we think it is quite reasonable and our protocol is suitable for WSN in saving energy and detecting holes.

The average hole detection time and average energy consumption for detecting the holes for different number of coverage holes are simulated and compared with distributed hole coverage (DHC) and path density (PS) algorithms as shown in Figs. 23 and 24. As presented in Fig. 23, it is observed that the hole detection time in our protocol is smaller than the DHC and PS as only boundary nodes that enclose the hole are considered to detect the presence of coverage holes. The average energy consumption in our protocol for detecting hole is very less as compared to the energy consumption in DHC and PS protocols. As shown in Fig. 24, though the energy consumption in our protocol increases with increase in the average number of coverage holes of the network, it is substantially less than the energy consumption of DHC and PS, which is due to the exchange of limited number of control packets to detect the holes. The characteristics of our protocol are compared with DHC and PS in Table 1. The network lifetime, which is defined as the duration of time from the initial deployment until the first node of the network is dead due to detection of coverage holes by executing our algorithm is simulated and is compared with DHC and PS as shown in Fig. 25. It is observed that the network lifetime in our protocol is better than DHC and PS as only one-hop neighbors of a node has to execute the algorithm to detect the presence of any hole. Besides, as contrary to DHC and PS, each node in our protocol uses shortest communication range to send and receive packets and, therefore, consume less power and maintains the longer network lifetime.

Table 1
Comparison of previous schemes with the proposed scheme.

Protocols	Avg. hole detection time/holes	Avg. energy consumption/holes	Control packets overhead	Constraints
PS	Longer	Higher Considers $R_c = 2R_s$	Large Uses flooding	Proper selection of routing algorithms
DHC	Smaller	High Considers $R_c = 2R_s$	Large Uses broadcasting	Can detect only bounded coverage holes
Our protocol	Smallest	Low Considers $R_c = R_s$	Small Uses unicasting	Cannot detect holes between sensors and boundary of the region

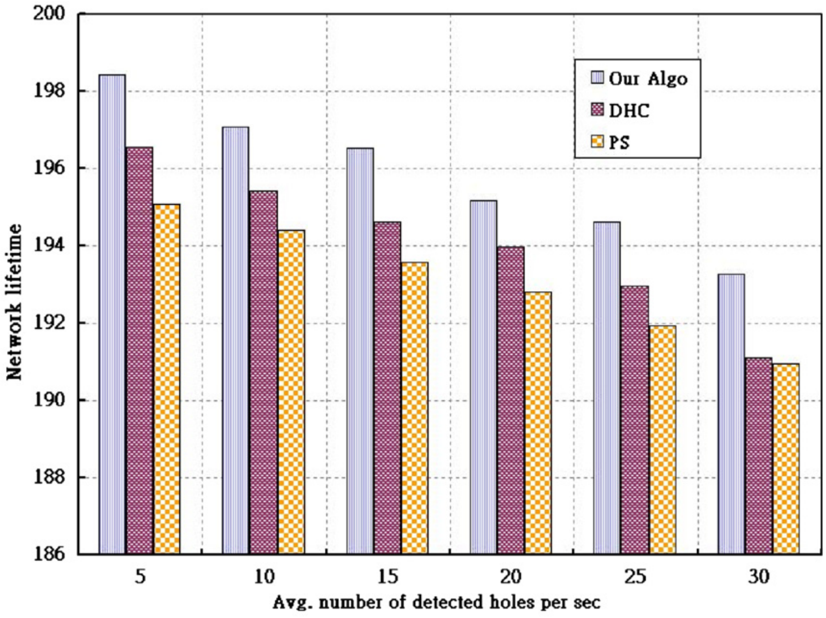

Fig. 25. Comparison of different protocols in terms of network lifetime with average number of detected holes per second.

6. Conclusions

In this work, a computational geometry approach based hole detection scheme is proposed, which can detect the coverage holes distributively. Detection of the holes are done using simple but efficient geometric methods taking only one and two-hop neighbors of each node. Global view of the coverage hole detection is proposed taking local information of the nodes. As contrary to the existing coverage hole detection protocols, communication range of each node is taken to be equal to the sensing range, by which more energy could be saved due to communication. Besides, the proposed protocol can detect coverage holes irrespective of any shape or size of the monitoring region, which is a different contribution in our work. Hence, implementation of our algorithms can be made in any form of monitoring region and, therefore, could be more useful and beneficial as compared to similar hole detection protocols.

References

Ahmed N, Kanhere SS, Jha S. The holes problem in wireless sensor networks: a survey. SIGMOBILE Mobile Computing and Communications Review 2005;9: 4–18.

Bejerano Y. Simple and efficient k-coverage verification without location information. In: Proceedings of IEEE 27th conference on computer communications (INFOCOM); 2008. p. 291–5.

Corke P, Peterson R, Rus D. Finding holes in sensor networks. In: Proceedings of the workshop on omniscient space: robot control architecture geared toward adapting to dynamic environments at ICRA; 2007.

Funke S. Topological hole detection in wireless sensor networks and its applications. In: Proceedings of the joint workshop on foundations of mobile computing; 2005. p. 44–53.

Funke S, Klen C. Hole detection or how much geometry hides in connectivity? In: Proceeding of the 22nd ACM symposium on computational geometry; 2006. p. 377–85.

Ganeriwal S, Kansal A, Srivastava S. Self aware actuation for fault repair in sensor networks. In: Proceedings of IEEE international conference on robotics and automation (ICRA); 2004.

Ghrist R, Muhammad A. Coverage and hole-detection in sensor networks via homology. In: Proceedings of the 4th international symposium on information processing in sensor networks; 2005. p. 254–60.

Hsieh K-Y, Sheu J-P. Hole detection and boundary recognition in wireless sensor networks. In: Proceedings of IEEE 20th international symposium on personal, indoor and mobile radio communications; 2009.

Liu B, Brass P, Dousse O, Nain P, Towsley D. Mobility improves coverage of sensor networks. In: Proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing; 2005.

Meguerdichian S, Koushanfar F, Potkonjak M, Srivastava MB. Coverage problems in wireless ad-hoc sensor networks. In: Proceeding of INFCOOM; 2001. p. 1380–7.

Silva VD, Ghrist R. Homological sensor networks. In: Notices of the American mathematical society, vol. 54(1), 2007. p. 10–7.

Soreanu P, Volkovich Z, Barzily Z. Energy-efficient predictive jamming holes detection protocol for wireless sensor networks. In: Proceedings of second international conference on sensor technologies and applications (SENSORCOMM); 2008. p. 306–11.

Wan PJ, Yi CW. Coverage by randomly deployed wireless sensor networks. In: IEEE transactions on information theory; 2006. p. 2658–69.

Wang G, Cao G, Porta L. Movement-assisted sensor deployment. In: Proceedings of twenty-third annual joint conference of the IEEE computer and communications societies (INFOCOM); 2004.

Wang G, Cao G, Porta L, Zhang W. Sensor relocation in mobile sensor networks. In: Proceedings of 24th annual joint conference of the IEEE computer and communications societies (INFOCOM); 2005.

Wang Y, Gao J, Mitchell JSB. Boundary recognition in sensor networks by topological methods. In: Proceedings of international conference on mobile computing and networking; 2006.

Watfa MK, Commuri S. Energy-efficient approaches to coverage holes detection in wireless sensor networks. In: Proceedings of IEEE international symposium on intelligent control; 2006a. p. 131–6.

Watfa MK, Commuri S. Power conservation approaches to the border coverage problem in wireless sensor networks. In: Proceedings of ICWN; 2006b.

Yu F, Choi Y, Park S, Lee E, Tian Y, Jin M, Kim S-H. Anchor node based virtual modeling of holes in wireless sensor networks. In: Proceedings of IEEE international conference on communications (ICC); 2008. p. 3120–4.

Zhang C, Zhang Y, Fang Y. Detecting coverage boundary nodes in wireless sensor networks. In: Proceedings of IEEE international conference on networking, sensing and control; 2006.