Sensors & Transducers, Vol. 179, Issue 9, September 2014, pp. 240-250

Sensors & Transducers

http://www.sensorsportal.com

Remote Holes Detection Algorithms for Wireless Sensor Networks

[1, 2] Kai-Biao Lin, [2, 3] K. Robert Lai, [2] Chiao-Yun Liu
[1] Department of Computer Science & Technology, Xiamen University of Technology, Xiamen 361024, P. R. China
[2] Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32026, Taiwan, ROC
[3] Innovation Center for Big data and Digital Convergence, Yuan Ze University, Taoyuan 32003, Taiwan
Tel.: +86-592-6291518, fax: +86-592-6291390
E-mail: kblin@xmut.edu.cn, krlai@cs.yzu.edu.tw

Received: 29 August 2014 /Accepted: 29 August 2014 /Published: 30 September 2014

Abstract: This paper develop a practical and energy-efficient distributed algorithm for the detection of coverage holes in a wireless sensor network. It assumes that the location of sensor nodes is available. We do this in two phases. First we want to identify a set of nodes of which encircle a coverage hole. This is done by introducing a concept of mono-covered arc, which represents the circumference of a non-overlapping region of the sensing disc of a sensor node. That is, a coverage hole or insufficient coverage region will be along the right-hand side of these directed mono-covered arcs. Based on this notion, a graph-theoretic distributed algorithm can then be developed to identify every simple boundary and interior hole locally. Furthermore, this graph-theoretic information can also be forwarded to the sink or base station to recognize the geometric characteristics among these coverage holes existing in the network fabric remotely. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Coverage hole, Wireless sensor network, Distributed algorithm.

1. Introduction

Wireless sensor network (WSN) is composed of inexpensive and battery-powered sensing devices having ability of collecting, storing, processing, and communicating data with each other [1, 2]. These tiny sensing devices are often deployed in the target field in large numbers and then collaborate to form a wireless ad-hoc network capable of reporting the data, phenomenon, or event to a remote data collection point, namely sink or base station [3]. Popular and successful applications of wireless sensor networks can be seen in such domains as object tracking, surveillance, environmental and structural monitoring, health related monitoring, traffic control, factory automation and inventory management etc. [4-6].

But these myriad of applications also present various challenges for the task of designing scalable, self-organizing, and energy-efficient sensor networks. The task becomes even more demanding if we consider the constraints of lightweight and low-capability sensor nodes with limited processing power, memory space, battery life, radio ranges, and communication bandwidth [7-9]. Additionally, the design of wireless sensor networks can also be greatly affected by the geometric distribution of the sensors deployed in the underlying environment. In practice, distributions of sensor nodes in a wireless sensor network are usually far from being uniform due to random aerial deployment, terrain variation, presence of obstructions, position changing, and node failures caused by power depletion or external forces. That is, the real sensor networks usually have coverage holes indicating the regions without enough working sensors.

The presence of holes in the underlying geometric environment could have important consequences on the performance of the sensor network at many levels [10, 11]. For perception applications such as object tracking, environmental monitoring, and military surveillance, the networks require sufficient coverage over the region of interest [12]. The shape or topology of the sensor field often indicates important features of the underlying environment. Thus, the identification of the holes in sensor networks is of primary interest because its presence often has physical correspondence and may also map to one of the special events that are being monitored by the sensor networks.

On the other hand, holes are also important indicators of the general health of a sensor network. Especially, understanding the global geometry and topology of the sensor field could have important implications for the design of several basic networking functionalities such as routing and data gathering mechanisms. For instance, the presence of such holes changes the topology of the networks and creates a communication void that has adverse effect on routing algorithms. Additionally, for information flow, the hole could also affect the overall capacity of the network.

Depending on the application environments and level of information constraints, algorithms for identifying various coverage holes in sensor networks can be generally classified into three categories: computational geometry approach, statistical approach, and topological method. The computational geometry method [13-15] uses the coordinates of the sensor nodes and standard geometric tools to determine the coverage characteristics.

One feature of this approach is that the precise geometry of the domain and exact location of the nodes must be available. In statistical method [16], it assumes a randomly and uniformly distributed collection of sensor nodes. The main idea is 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. For topological methods [17-19], the main feature was based on the network topology or connectivity information to identify the holes. These methods are attractive particularly for a large scale of sensor network in which the location information is not available.

2. System Model

2.1. Assumptions

Consider a set of n sensors denoted by $S = \{S_1, S_2, \ldots, S_{n-1}, S_n\}$ and are deployed over a rectangular monitoring region R. Let, $s_i = (x_i, y_i)$ be the position of i-th sensor, for i=1~n. It is to be assumed that T1~T4 are the vertices of the rectangular region R as shown in Fig. 1, whose coordinates are known and are given by $T_i = (X_{Ti}, Y_{Ti})$, for i=1~4. Each sensor has its location information. We consider a homogeneous sensor network, where each sensor has uniform sensing (r_s) and communication (r_c) range and $r_c = 2r_s$. Throughout our work, the term “boundary” means the boundary of the monitoring region R.

2.2. Definitions

Definition 1 [Sensing Disc] Sensing disc is the disc with radius of r_s and centered at an active sensor. Any object within the sensing disc can be perfectly detected by a sensor. Throughout the paper, radius of the sensing disc is referred to as sensing range (r_s).

Definition 2 [Boundary/Non-Boundary Sensor] If sensing disc of any sensor intersects with the boundary of the monitoring region, the sensor is a boundary sensor. Otherwise, it is a non-boundary sensor. For example, as shown in Fig. 1, sensors S_7, S_{11}, S_{13}, S_{14} and S_{15} are boundary sensors and the rest are non-boundary sensors.

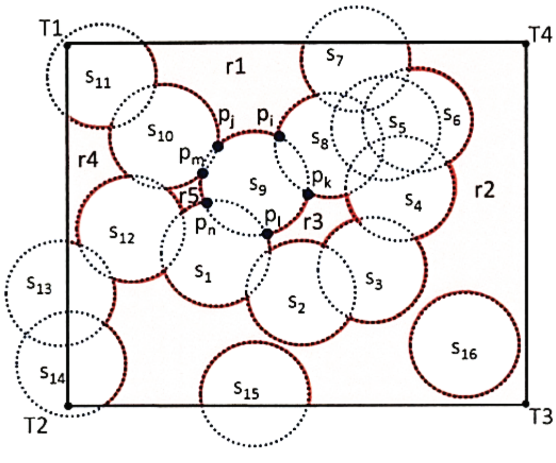


Fig.1. Example of a monitoring region R where 16 sensors deployed.

Definition 3 [Island/Archipelago Sensor] If a non-boundary sensor has no neighbor, the sensor is an island sensor. If a non-boundary sensor has connecting neighbors, it is an archipelago sensor. For example, as shown in Fig. 1, $S_1 \sim S_6$, S_8, S_9, S_{10} and $S_{12}$are archipelago sensors, and S_{16} is an island sensor. Through, S_{15}has no neighbor, it is considered as a boundary sensor, according to our definition.

Definition 4 [Coverage Hole] If any part of the monitoring region R is not covered by the sensing disc of a sensor, that part is termed as a coverage hole.

Definition 5 [Simple/Complex/Boundary/Non-boundary Coverage Hole] If a coverage hole is enclosed by single set of connected sensors (boundary/non-boundary), the coverage hole is simple. If the coverage hole is enclosed by sensors (boundary and non-boundary) and boundary of the monitoring of the monitoring region R, the coverage hole is a boundary coverage hole. Otherwise, the coverage hole is a non-boundary one. As shown in Fig. 1, r3 and r5 are simple and non-boundary coverage holes, as no boundary sensor encloses the holes. r1 and r4 are simple and boundary coverage holes, as boundary of the monitoring region encloses the holes. r2 is a complex and boundary coverage hole as it is enclosed by three sets of sensors, $\{S_1, S_2, S_3, S_4, S_6, S_7, S_{12}, S_{13}, S_{14}\}$, $\{S_{15}\}$, and $\{S_{16}\}$, and is enclosed by the boundary of the monitoring region.

3. Remote Hole Detection Algorithms

The hole detection algorithm comprises two phases. In the first phase, we introduce a concept, Mono-covered Arc, which implies a region on the outside of certain portions of the sensing disk of a sensor that may remain uncovered, and then, a set of connected mono-covered arcs collectively can define the perimeter of a coverage hole. Algorithm 1 is developed to determine the mono-covered arcs of each sensor node. Then, in the second phase, we propose a graph-theoretic hole detection method. Based on the mono-covered arcs information, Algorithm 2 proceeds to build a directed graph which is used to detect every simple hole existing in the network. Additionally, these graph-theoretic information are ultimately send back to the sink or base station to recognize the extent of coverage holes in the network fabric and also to identify the presence of complex holes which are enclosed by more than one set of connected mono-covered arcs.

3.1. Mono-covered Arc

In the random distribution of sensor nodes, it is obvious that the sensing disc of a sensor may be fully or partially overlapping (or disjoint) with other sensors. In our work, the arc pertaining to a non-overlapping region of the sensing disc of a sensor is termed as a mono-covered arc, as shown in Fig. 1, defined by three sets of end points $\{(p_i, p_j)$, (p_m, p_n), and $(p_l, p_k)\}$, measured in counterclockwise direction. Besides, S_5 has no mono-covered arc as its sensing disc is fully covered by other sensors. Since, S_{16} is an island sensor, its whole sensing disc is considered as the mono-covered arc per the definition.

Note that the end points of the mono-covered arc are represented by an ordered pair of points and the mono-covered arc is measured in counterclockwise direction taking those pair of pints. We develop an algorithm to determine the mono-covered arc of each sensor as described in the following subsection. Prior to execution of this algorithm, each sensor broadcasts packet with its location information to its one hop neighbors. Upon receiving location information from neighbors, each sensor constructs a neighbor table and executes the following schemes: End Points Estimation and Mono-covered Arc Selection.

3.1.1. End Points Estimation

The sensor who receives location information from the neighbors has to determine the end points and the neighbors associated with those end points. Note that island sensors do not have neighbors and therefore they do not estimate any end points. Each boundary or archipelago sensor chooses one neighbor and calculates the points of intersection of its sensing disc with its neighbors. If that point of intersection is inside R and is not covered by any other neighbors, those intersection points become member of the End Point Set P and the corresponding neighbors associated with those points become member of the Associated Neighbors Set N.

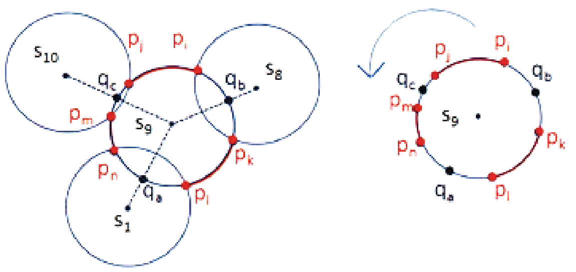
Let, any sensor $s_i \notin$ island sensor has to estimate the end points. First, s_i chooses one neighbor s_j from its neighbor set and calculates the intersection points p_i, p_j. If p_i or p_j is inside R and is not covered by any other neighbors of s_i, p_i or p_j become elements of the end points set P. Besides, s_j is selected as the member of the associated neighbors set N. This process continues for all neighbors of s_i. For example, as shown in Fig. 1, if we consider sensor s_9, its end points set can be given as $P(s_9) = \{p_j, p_m, p_n, p_l, p_k, p_i\}$. Besides, its associated neighbors set can be given as $N(s_9) = \{s_{10}, s_1, s_8\}$.

3.1.2. Mono-covered Arc Selection

In this phase, a sensor determines the mono covered arc elements taking end points set and the associated neighbors set into consideration. Each sensor s_i calculates another points set $Q = q \in Q \mid$ q is a point on the line $s_i s_j$ and $| s_i q | = r_s, \forall s_j \in N(s_i)\}$. In other words, if s_j is an associated neighbor of s_i, and q is point on the line joining the location of $s_i s_j$, then $| s_i q | < r_s$.

After finding the points of set Q, all points of P and Q are placed on the circumference of the sensing disc of the sensor. The pair of points of P that does not contain any point of Q is considered as a mono covered arc. For example, as shown in Fig. 2(a), s_{10}, s_1 and s_8 are associated neighbors of s_9 and p_j, p_m, p_n, p_l p_k and p_i are end points set of s_9.

As per the algorithm described above, q_a, q_b and q_c become the elements of set Q. Hence, as shown in Fig. 2(b), the end points $\{p_i, p_j, p_k, p_l, p_m, p_n\}$ and points $\{q_a, q_b, q_c\}$ are placed on the circumference of the sensing disc of s_9. Considering in counterclockwise direction, any pair of end points are selected such that neither q_a nor q_b, nor q_c lies in between any pair of end points. Thus, (p_i, p_j), (p_m, p_n) and (p_m, p_n) become the mono covered arc.



(a) (b)

Fig.2. Example of Mono Covered Arc. (a) Estimation of end points set and points belong to set Q. (b) Decision of mono covered arc.

The complete algorithm for estimating the end points set, associated neighbors set and finally selecting the mono covered arcs is given in Fig. 3.

Algorithm 1: Mono-covered Arc Selection of node i

Notation:
Nr: *Nr*={s| s is the neighbor of i};
N: the associated neighbors set;
P: the end points set of *i*;
Q: q points set of *i;*
MC_{Arc} *(i)*: the mono-covered arcs set of node *i*.
End Point Estimation (*Nr*), return P,N
Set *P*=*N*=∅;
Foreach sensor j∈*Nr* **do**
ComputeIntersectionPoints(i, j)
return intersection points set p[];
for (k=0~(Size(p[])-1))
if (CheckIsNotCoveredAndInsideR(p[k]));
P=P∪p[k];
N=N∪j;
if (i∈ boundary sensor);
ComputeIntersectionBounaryPoints(i),
return intersection with boundary points set b[];
for (k=0~(Size(b[])-1)))
if (CheckIsNotCoveredAndInsideR(b[k]));
P=P∪b[k];
Mono-Covered Arc(P,N), return MC_{Arc}(i)
Set Q=∅
if (P ≠∅)
foreach sensor j∈N do
compute q points, Q=Q∪q
foreach p_i, p_j∈P, and (p_i, p_j)
is one sequential pair of counterclockwise order do
If (CheckNoQBetweenAndInsideR(p_i, p_j))
MC_{Arc}(i) = MC_{Arc}(i) ∪ (p_i, p_j);
else set MC_{Arc}(i)=∅

Fig.3. Algorithm for Determining Mono-covered Arc.

3.2. Graph-theoretic Hole Detection

This part describes how to use the concept of mono-covered arc information to build a directed graph for identifying the coverage holes. Note that each boundary and archipelago sensors have their own mono-covered arc information and the corresponding end points set. However, the island sensor only knows their location information. It is assumed that each sensor, upon getting the mono-covered arc information, forms the directed graphs which represent the geometric characteristics of coverage holes in the network. And locally this information can be used to detect the holes, which exist along the right hand side of each directed graph. To further identify the complex holes and their extent at a remote location, the graph-theoretic information is forwarded to the sink.

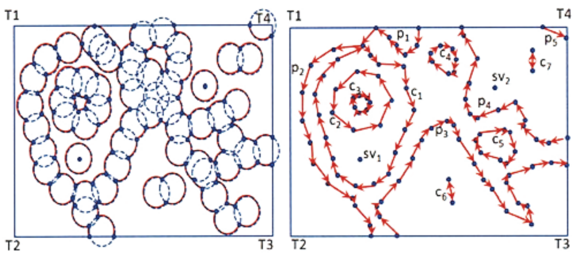
3.2.1. Construction of Graph

Let, S be the set of sensors deployed over the monitoring region R, and MC_{Arc} be the set of mono-covered arcs of all nodes of the network, i.e. $MC_{Arc} = \{(p_i, p_j) \mid (p_i, p_j) \in MC_{Arc}(s), \forall s \in S\}$. Let, $G(V, E)$ be a non-empty directed graph of a pair of disjoint sets (V, E). $\exists$ a directed edge e from vertex p_i to p_j, $\forall e \in E$, p_i and $p_j \in V$, such that initial vertex $init(e) = p_i$ and terminal vertex $ter(e) = p_j$, where $(p_i, p_j) \in MC_{Arc}$. Based on this assumption, first, we construct different directed graphs $G(V, E)$ as per the following definitions.

Definition 6 [Single Vertex] Let, G (V, E) be a directed graph of non-empty alternating sequence $v_0, v_1, \ldots, v_n \in V$, and $e_i = (v_{i-1}, v_i) \in E$, for all i<n. In $G(V, E)$, $\forall v_i \in V$, v_i is a Single Vertex, if and only if no directed edge is initiated from or terminated at v_i, and v_i is formed by an island sensor. In other word, single vertex v_i is an isolated point. As shown in Fig. 4(b), SV_1 and SV_2 are a single vertices, which are formed due to presence of an island sensor.

Definition 7 [Path] If $v_0 e_1 v_1 e_2 v_2 e_3 \ldots e_n, v_n$ are the alternating sequence of vertices and edges of directed graph $G(V, E)$, such that v_0 and v_1 are on the boundary of the monitoring region R, the graph formed by the directed edges $e_i = (v_{i-1}, v_i) \in E$, for all $i < n$ is called a Path. As shown in Fig. 4(b), p_1, p_2, p_3, p_4 and p_5 are different path formed from the original topology given in Fig. 4(a).

Definition 8 [Cycle] If $v_0 e_1 v_1 e_2 v_2 e_3 \ldots e_n, v_n$ are the alternating sequence of vertices and edges of directed graph $G(V, E)$, such that $v_0 = v_n$, the graph formed by the directed edges $e_i = (v_{i-1}, v_i) \in E$, for all $i < n$ is called a Cycle. As shown in Fig. 4(b), c_1, c_2, c_3, c_4, c_5, c_6 and c_7 are different Cycles formed from the original topology given in Fig. 4(a).



(a) (b)

Fig. 4. (a) Original topology of the sensor network with coverage holes. (b) Construction of directed graphs G(V,E) as per our definitions.

Note that each sensor knows its end points set and the associated neighbors set to form its mono-covered arcs. Prior to detecting the holes, each sensor has to construct the types of graph with respect to its associated neighbors, which is ultimately forwarded to the sink to analyze the nature and presence of the holes. In our work, it is assumed that any sensor s_i initiates the graph construction phase by unicasting its own mono-covered arcs which the initial vertex is on the boundary as the initiator. Each sensor appends the mono-covered information of its associated neighbor to its own mono-covered arcs data and reconstructs the graph. The process of contrasting graph is terminated until it reaches at the boundary vertex (in this case a Path is formed), or the process reaches at the initiator vertex (in this case a Cycle is formed). In case of island sensor, only Single Vertex is formed and therefore only location information of the sensor is forwarded to the sink using higher power level. The complete algorithm of the graph construction is given in Fig.5, and example of formation of different types of graphs is given in Fig. 3. In this algorithm, we think the EdgeTraversal function can be implementing in each sensor easily by means of message passing, and for the end point, where sensor can determine the graph form. Note that, we use P, C and V_s to represent the set of paths, cycles, and single vertices, respectively in our algorithm. And the island sensors represent their location as one set of the same points.

Algorithm 2: Construction of Graph

Notation:

MC_{Arc}: $MC_{Arc} = \{(p_i, p_j) | (p_i, p_j) \in MC_{Arc}(s), \forall s \in S\}$;
β: $\beta = \{p | \forall (p,*)$ or $(*,p) \in MC_{Arc}$ and p is on the boundary$\}$;
g: subset of edges. $g \subseteq MC_{Arc}$, $g = \{(u_k, v_k) | k=0 \sim n\}$
InE(): function to choose an initial edge, which is initiated at the boundary point has highest priority;
NEE (): function to choose the next edge which is connected to the previous one;
Construction of Graph(MC_{Arc}), return P,C
While($MC_{Arc} \neq \varnothing$)
g=EdgeTraversal(MC_{Arc});
if ($u_0 \in \beta$and $v_n \in \beta$)
$P = P \cup g$;
Else if ($u_0 == v_n$)
$C = C \cup g$;
$MC_{Arc} = MC_{Arc} - g$;
EdgeTraversal(MC_{Arc}), return g
$g = \varnothing$;
(u_0, v_0)=InE(MC_{Arc});
$g = g \cup (u_0, v_0)$;
set k=0;
while ($v_k \notin \beta$ and $u_0 \neq v_k$)
(u_{k+1}, v_{k+1})=NEE(MC_{Arc});
k=k+1;
$g = g \cup (u_k, v_k)$;
return g;

Fig.5. Graph Construction Algorithm.

Corollary 1 In a random deployment of nodes, any form of graph does not exist, other than a single vertex, path, or cycle.

Proof. Per our definitions and as shown in Fig. 4(b), single vertex is formed due to an island sensor. It can be observed every boundary vertex has only one directed edge either started from or terminated at.

This implies any graph that is initiated at a boundary vertex must be ended at another boundary vertex and forms a path. Additionally, it also can be observed every interior vertex is associated with two directed edges of which one is started from and, at the same time, another is terminated to. In other words, any graph that is initiated at an interior vertex must be terminated at itself and thus forms a cycle.

This proof is also consistent with a touch point, which indicates a sensing disk is intersected with that of neighboring nodes or the boundary at a single point. For an interior touch point vertex, it has a total of four directed edges associated with. Of these four edges, two are going out and the other two are coming from neighboring vertices. For a boundary touch point vertex, there are a pair of directed edges associated with. Of these two edges, one is starting from and the other is entering to the vertex.

3.2.2. Hole Detection Schemes

In the following subsection, we would prove certain lemmas which are used to detect the coverage holes.

Corollary 2 $\forall$ directed edge $e \in E$, there exists a hole along right side of that edge e.

Proof. $\forall e, e \in E$, suppose $e = (p_i, p_j)$.

=> There is a mono-covered $arc(p_i, p_j)$. Assume $(p_i, p_j) \in MC_{Arc}(s)$. By the definition, mono-covered arc is a section of sensing circle of sensor s, and is inside R and is not covered by any other sensors.

=> There is a hole besides (p_i, p_j). Since, mono-covered $arc(p_i, p_j)$ has recorded in counterclockwise order; sensing disk of s is totally in the left side of the $arc(p_i, p_j)$.

=> The hole must be in the right side of (p_i, p_j). Since, (p_i, p_j) represents a directed edge e in $G(V, E)$, the hole is still in the right of e.

Lemma 1 $\forall$ path $p \in P$, there exists a boundary coverage hole along right side of that path p.

Proof. Let, $p = v_0 e_0 v_1 e_1 \ldots v_{n-1} e_{n-1} v_n$ be a path. Hence, v_0 and v_n must be on the boundary and $\exists e_i \in p \ s.t. e_i = init(v_i) ter(v_{i+1})$.

=> From Corollary 3.2, $\exists$ a hole along right side of each e_i, as $p = \text{union of } e_i, i = 1 \sim n$.

=> $\exists$ A coverage hole enclosed by boundary points v_0 and v_n

=> $\exists$ a boundary coverage hole along right side of path p.

Lemma 2 $\forall$ cycle $c \in C$, if c encloses any region r in a clockwise direction, there exists a non-boundary coverage hole along cycle c.

Proof. Let $c = v_0 e_0 v_1 e_1 \ldots v_{n-1} e_{n-1} v_n e_n v_0$ is a cycle.

=> from Corollary 3.2, $\exists$ a hole along right side of each e_i, since $c = \text{union of } e_i, i = 1 \sim n$, and c encloses r in the clockwise direction.

=> r is bounded by c and along right side of it.

=> $\exists$ a non-boundary coverage hole along c in r.

Lemma 3 $\forall$ cycle $c \in C$, if c encloses any region r in a counterclockwise direction, there exists a coverage hole along c in $R - r$.

Proof. Let, $c = v_0 e_0 v_1 e_1 \ldots v_{n-1} e_{n-1} v_n e_n v_0$ be a cycle.

=> From Corollary 3.2, $\exists$ a hole along right side of each e_i, since $c = \text{union of } e_i, i = 1 \sim n$, and c encloses r in the counterclockwise direction.

=> r is bounded by c and in the left side of c.

=> $R - r$ is in the right side of c.

=> $\exists$ a coverage hole along c.

Lemma 4 All coverage holes exist in R are along at least one path or cycle in $G(V, E)$, if there exists more than one coverage holes in R.

Proof. We prove it by contradiction. Suppose there are n coverage holes in R, i.e. $r_1, r_2, \ldots, r_n, n \geq 2$, and each coverage holes are isolated. Let, there exists a coverage hole r_i enclosed by a single vertex and a loop, except for the boundary.

=> Since, single vertex cannot separate R into isolated parts, r_i cannot be isolated from other coverage holes.

$(\rightarrow\leftarrow)$ If there exists more than one coverage holes.

=>All coverage holes must be along with at least one path or cycle.

Lemma 5 If there exists only one path p in $G(V,E)$, without existence of any other forms of graph, there exists only one simple boundary coverage hole r along right side of p.

Proof. (Existence and boundary part) From lemma 3.1, there is a coverage hole with boundary in the right side of p.

(Only one part) prove by contradiction. Suppose $\exists$ another coverage hole r', $r' \neq r$.

=>from lemma 3.4, r' is along one path or cycle.

=>If r' is beside a cycle, s.t. $|C| \neq (\rightarrow\leftarrow)$.

=> If r' is beside a path p', and $p' \neq p(\rightarrow\leftarrow)$.

=> If r' is beside a path p', and $p' \neq p$, s.t. r' is in the right side of p, then $r' = r(\rightarrow\leftarrow)$.

(Simple part) prove by contradiction. Suppose r is complex.

=>except for p, $\exists$ another single vertex, or path, or cycle $(\rightarrow\leftarrow)$.

Lemma 6 If there is only one clockwise cycle c in $G(V,E)$, without existence of any other forms of graph, and c close a region r, there exist only one simple non-boundary coverage hole r.

Proof. (Existence and non-boundary part) From Lemma 3.2, there is a non-boundary coverage hole which is bounded by c along c in r.

(Only one part) prove by contradiction. Suppose $\exists$ another coverage hole $r', r \neq r$.

=>from lemma 3.4, r' is along one path or cycle.

=>If r' is beside a path p, s.t. $|P| \neq 0(\rightarrow\leftarrow)$.

=> If r' is beside a cycle c', and $c' \neq c(\rightarrow\leftarrow)$.

=> If r' is beside a cycle c', and $c' = c$, s.t. $r' = r(\rightarrow\leftarrow)$.

(Simple part) prove by contradiction. Suppose r is complex.

=>except for c, $\exists$ another single vertex, or path, or cycle $(\rightarrow\leftarrow)$.

Lemma 7 If there is only one counterclockwise cycle c in $G(V,E)$, without existence of any other forms of graph, and in c close a region in r, there exist only one simple boundary coverage hole in $R-r$.

Proof. (Existence part) From Lemma 3, there is a coverage hole which is along c in $R-r$.

(Only one part) prove by contradiction. Suppose $\exists$ another coverage hole r', $r' \neq R-r$.

=>from lemma 3.4, r' is beside one path or cycle.

=>If r' is beside a path p, s.t. $|P| \neq 0(\rightarrow\leftarrow)$.

=> If r' is beside a cycle c', and $c' \neq c(\rightarrow\leftarrow)$.

=> If r' is beside a cycle c', and $c' = c$, s.t. $r' = R - r(\rightarrow\leftarrow)$.

(Simple part) prove by contradiction. Suppose r is complex.

=>except for c, there exist another single vertex, or path, or cycle $(\rightarrow\leftarrow)$.

(Boundary part) Because $R-r$ include the boundary, therefore $R-r$ is a boundary coverage hole.

Lemma 8 If there is no path and cycle in $G(V,E)$, and $|V| \neq 0$, there is only one boundary coverage hole.

Proof. (Existence part) (1) if $|E| = 0$, s.t. all vertices $\in VS$.

=>all vertices are the island sensors.

=>coverage hole existing (2) if $|E| \neq 0$, s.t. there are mono-covered arcs.

=> Coverage hole existing

(Only one part) because $|P| = 0$ and $|C| = 0$, there R is not divided into different isolated parts, and the whole R is seen as one coverage hole.

(Boundary part) prove by contradiction if the boundary is covered totally, s.t. there will be a cycle $(\rightarrow\leftarrow)$.

=>the boundary is not covered totally.

=>the hole is a boundary coverage hole.

4. Performance Analysis

4.1. Simulation Setups

We considered a convex monitoring region of size 200*200 m^2 and have simulated our algorithms using ns-2.33. About 300 to 1000 sensors are distributed randomly over the monitoring region. The simulation parameters are setup according to IEEE 802.15.4 MAC/PHY specification and radio characteristics of IEEE 802.15.4 compliant product CC2420 [20] along with AODV routing protocol and TwoRayGround propagation model. Initially, each node is assumed to have fixed amount of 50J reserved energy and energy cost due to forwarding of each control packet is taken as 0.3J. Based on the variable number of deployed nodes, density of the nodes is considered as 1, 3, and 5 nodes/m^2. Throughout our simulation, sensing range and communication ranges are set to be 10 m and 20 m, respectively. The traffic data rate is kept as 250 00Kbps and control packets are sent in every 0.2 seconds to detect the neighbors, which is continued to get the final list of one-hop neighbors of each node.

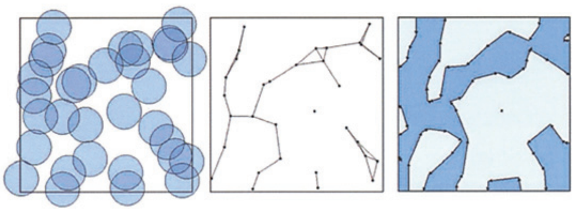
In the simulation, holes are generated randomly among multi-hop and fully connected nodes such that they can form different groups of disjoint set of nodes. All types of situations such as simple with boundary, complex with boundary and simple with non-boundary types of holes are consider getting the real time hole detection environment. In order to implement the mono-covered arc and graph construction algorithms, all nodes are given location information to estimate the endpoint set and associate neighbors of each node. The mono-covered arc information of each node is unicast to the one-hop neighbors to construct the graph.

4.2. Simulation Results

In this part we discuss the simulation results of our hole detection schemes. We separate our simulation results into two different parts such as hole information topology and performance evaluation of our work. In the performance evaluation part, we compare our hole detect methods with some known hole detection algorithms. Details of our simulation results are given as follow.

4.2.1. Topological Hole Formation

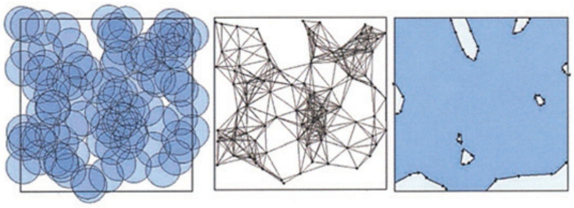
In order to get real picture of our simulations based on our hole information schemes, we generated different types of holes as shown in Fig. 6, Fig. 7 and Fig. 8.



(a) (b) (c)

Fig.6. Original topology, connectivity figure, and the construction of graphs from our algorithms with average 1 node/ m^2.

We have simulated for different densities of nodes with (a) original topology of the network, (b) connectivity among nodes, and (c) construction of graphs based on our algorithms. As shown in Fig. 6, density of the network is considered as 1 node/m^2, which implies that each unit square area of the monitoring region is covered by one node on average. There are 9 paths and one island, where 3 paths and one island enclose the one coverage hole. Besides, there are 7 boundary coverage holes, out of which 6 are simple, and 1 is complex. And for Fig. 7, density of the network is considered as 3 node/m^2, there are 6 paths and 3 cycles, where 6 simple boundary coverage holes, and 3 simple non-boundary coverage holes. For Fig. 8, density of the network is considered as 5 node/m^2, there are 2 paths and 1 cycle, where 2 simple boundary coverage holes, and 1 simple non-boundary coverage hole. We find that our work can show the coverage holes exactly and even if the connectivity is interrupted. Thus we think our work is complete, and no need to assume the density of network or the connectivity.



(a) (b) (c)

Fig.7. Original topology, connectivity figure, and the construction of graphs from our algorithms with average 3 node/ m^2.

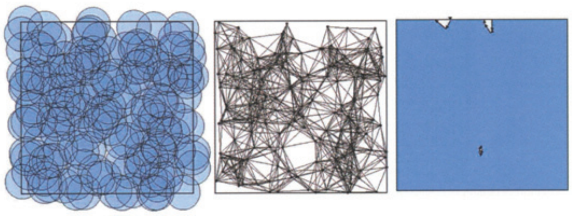


Fig.8. Original topology, connectivity figure, and the construction of graphs from our algorithms with average 5 node/m^2.

4.2.2. Performance Evaluation

Here, we analyze the performance and efficiency of our algorithms. If n is the total number of deployed sensors and M is the maximum number of neighbors of any sensor, each sensor has to calculate the intersection point sets with its neighbors. Hence, each node requires M units of time for calculation. Each sensor checks if the intersection points are covered by other neighbors or not, for which it needs at most $(M-1)$ unit times of calculation. Hence, computational complexity of our hole detection algorithm can be estimated as $0(n.M^2)$.

In order to get the practical insight of our schemes, we simulate our algorithms to find average number of neighbors for different number of holes, as shown in Fig. 9. It is found that more number of neighbors have to participate to execute the hole formation algorithm, if number of coverage holes are increased. We consider two criteria to evaluate performance of our algorithms. They are the average hole detection time and average power consumption for detecting the holes. We use the number of coverage holes and average density of the nodes to evaluate those parameters, as shown in Fig. 10, Fig. 11 and Fig. 12.

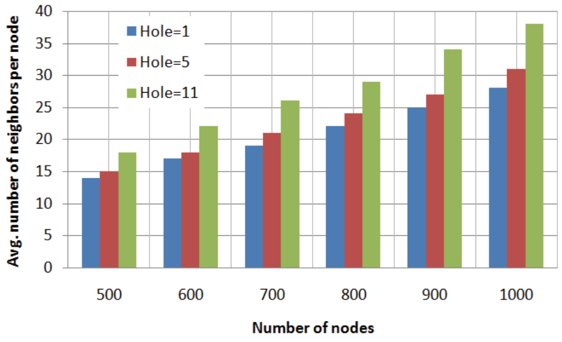


Fig.9. Relationship between average numbers of neighbors for different number of coverage holes.

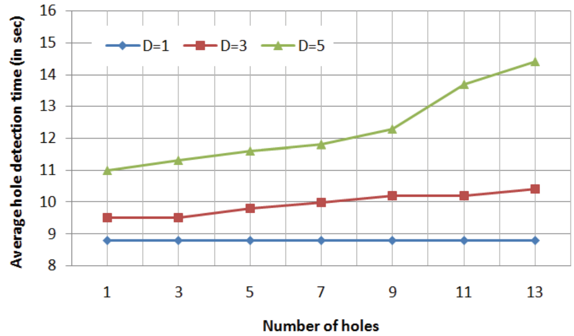


Fig.10. Average hole detection time for different number of holes with different node densities.

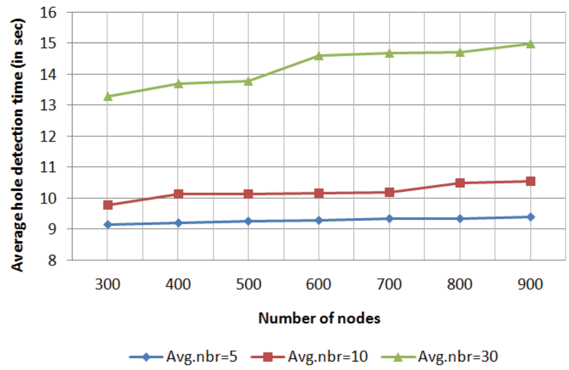


Fig.11. Average hole detection time for different number of average neighbors and different number of nodes.

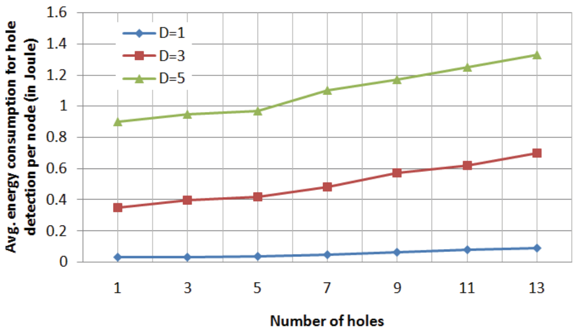


Fig.12. Average amount of power consumption for different densities of the nodes and different number of holes.

As shown in Fig. 10, it is observed that the average hole detection time increases with increase in different number of holes, which is quite obvious. Besides, the hole detection time also increases with increase in density of the nodes. 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 package delivery time, then time cost for delivering each packet can be estimated as $(n.m.T)$. As shown in Fig. 11, the average hole detection time increases with increase in different number of nodes. The hole detection time is also increased with increase in the average number of neighbors. It is due to estimation of more end points and associated neighbors to find the mono-covered arc. The average power consumption for detecting different number of holes is given in Fig. 12. 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 $(m.P) + P_{idle}$. From Fig.12, it is found that average power consumption for detecting the holes is increased with increase in the number of holes. The average power consumption is also higher if density of the nodes over the network is increased.

Here, we compare our work with similar hole detection algorithms [21-23]. The hole detection algorithms mentioned in [21, 22], require fully connectivity of somewhere. They can't use the communication graph or connectivity information to detect the coverage holes. The Distributed Hole Coverage (DHC) algorithm [23], introduces similar idea with ours. They use the notation of coverage of the sensing disc to detect the holes. However, as shown in Fig.13(a) is the result of DHC to detect all nodes which enclose the holes, and Fig.13(b) is our result, it is obviously that DHC is unable to detect the nodes which beside the boundary coverage holes. It can only detect the nodes beside non-boundary coverage holes, and therefore hole detection using DHC is totally limited. Compare to our work, we can find all nodes beside all coverage holes correctly.

It is to be noted that the Topological Boundary Recognition (TBR) algorithm uses topological method to detect the coverage hole. And another distributed algorithm Path Density (PS) algorithm use the density of each path from the same node to decide which one pass the coverage hole, and detect the coverage hole by means of these information. Both algorithm need to use flood the network twice to define the coverage hole. As shown in Fig.14, under the same situation which there are several number of nodes in R, TBR needs more time to construct the tree structure and to get shortest path. And for PS, it needs time to broadcast the density messages, and return the hole information by means of broadcast way. Although DHC need less time than our work in local part, that is because they do not form the meaningful graph to define the coverage hole. The same situation occurs when considering energy, as shown in Fig. 15, our method also consume less energy than TBR and PS. Although, the more energy we need than DHC, but not much and can make our method more distributed and define the coverage holes more definitely.

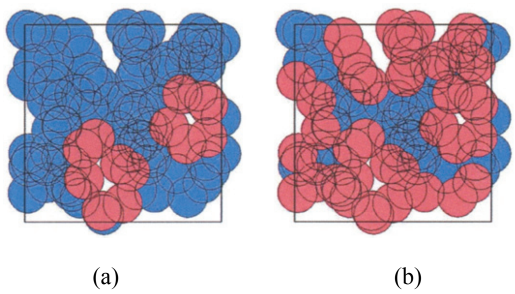


Fig.13. Comparison of hole detection methods of our algorithm with DHC.

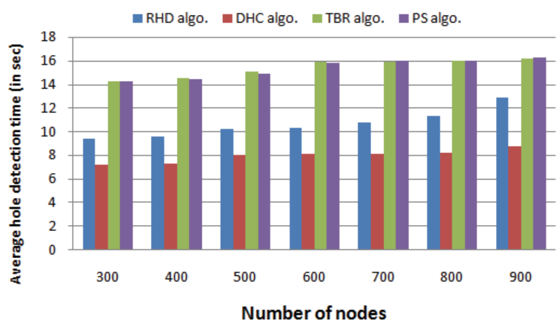


Fig.14. Average hole detection time for different number of sensors.

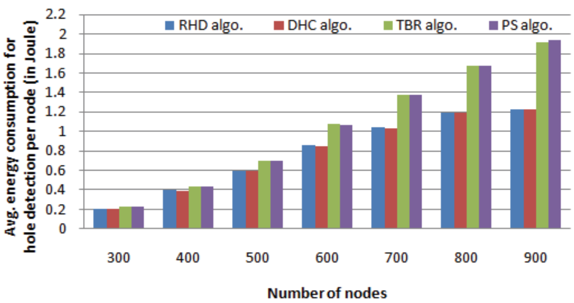


Fig.15. Average power consumption for different number of nodes.

5. Conclusions

In this paper, we propose a distributed hole detection scheme that can detect the coverage holes remotely. We propose algorithms to how get the mono-covered arc information of the nodes from the location information of the neighboring nodes. Using local mono-covered arc information, each node can construct the graphs, which is later forwarded to the sink. Finally, we propose several rules in form of lemmas so that sink can analyze the nature and position of the coverage holes remotely. Our remote hole detection schemes can any type of holes such as boundary or non-boundary holes. As compared to similar hole detection algorithms, our algorithm can detect holes efficiently with least hole detection time. Hence, implementation of our algorithms can give more beneficial results as compared to other centralized methods.

Acknowledgments

The work is partially supported by the Department of Housing and Urban-Rural Development Project under Grant No. 2013-K8-34, Xiamen University of Technology's International Cooperation and Exchange Project under Grant No. E201300200.

References

[1]. O. Banimelhem, M. Mowafi, and W. Aljoby, Genetic algorithm based node deployment in hybrid wireless sensor networks, *Communications and Network*, Vol. 5, Issue 4, 2013, pp. 273-279.

[2]. C. Zhu, C. L. Zheng, L. Shu, and G. J. Han, A survey on coverage and connectivity issues in wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 35, Issue 2, March 2012, pp. 619-632.

[3]. H. S. Rajat Bhardwaj, Aman Singh, Coverage hole removal in WSNs, *International Journal of Smart Sensors and Ad Hoc Networks*, Vol. 3, Issue 1, 2013, pp. 40-43.

[4]. S. Babaie and S. S. Pirahesh, Hole detection for increasing coverage in wireless sensor network using triangular structure, *IJCSI International Journal of Computer Science*, Vol. 9, Issue 1, No. 2, 2012, pp. 213-218.

[5]. B. Wang, H. B. Lim, and D. Ma, A survey of movement strategies for improving network coverage in wireless sensor networks, *Computer Communications*, Vol. 32, Issue 13-14, 2009, pp. 1427-1436.

[6]. H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors, *IEEE Transactions on Industrial Informatics*, Vol. 10, Issue 1, 2014, pp. 163-174.

[7]. C.-F. Hsin and M. Liu, Randomly duty-cycled wireless sensor networks: dynamics of coverage, *IEEE Transactions on Wireless Communications*, Vol. 5, Issue 11, 2006, pp. 3182-3192.

[8]. W. Peng-Jun and W. Chih, Coverage by randomly deployed wireless sensor networks, *IEEE Transactions on Information Theory*, Vol. 52, Issue 6, 2006, pp. 2658-2669.

[9]. I. H. Peng and Y.-W. Chen, Energy consumption bounds analysis and its applications for grid based wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 444-451.

[10]. S. S. K. N. Ahmed, and S. Jha, The holes problem in wireless sensor networks: a survey, *Mobile Computing and Communications Review*, Vol. 9, Issue 2,2005, pp. 4-18.

[11]. X. X. Nan Qu, Fang-Ming Shao, Coverage hole problem under sensing topology in wireless sensor network, in *Proceedings of the International Conference on Software Engineering and Computer Science (ICSECS2013)*, 2013, pp. 55-58.

[12]. S. Karmakar and A. Roy, Holes detection in wireless sensor networks: a survey, *International Journal of Modern Education and Computer Science*, Vol. 6, Issue 4, 2014, pp. 24-30.

[13]. Y.-C. T. Chi-Fu Huang, The coverage problem in a wireless sensor network, *Mobile Networks and Applications*, Vol. 10, Issue 4, 2005, pp. 519-528.

[14]. G. Wang, G. Cao, and T. F. La Porta, Movement-assisted sensor deployment, *IEEE Transactions on Mobile Computing*, Vol. 5, Issue 6,2006, pp. 640-652.

[15].H.-C. Ma, P. Kumar Sahoo, and Y.-W. Chen, Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 34, Issue 5, 2011, pp. 1743-1756.

[16]. P. B. Benyuan Liu, Olivier Dousse, Mobility improves coverage of sensor networks, in *Proceedings of the 6thACM International Symposium on Mobile ad hoc Networking and Computing*, 2005, pp. 300-308.

[17]. V. d. S. a. R. Ghrist, Homological Sensor Networks, *Notices of the American Mathematical Society*, Vol. 54, Issue 1, 2007, pp. 10-17.

[18]. D. Dong, X. Liao, K. Liu, Y. Liu, and W. Xu, Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches, *IEEE Transactions on Computers*, Vol. 61, Issue 10, 2012, pp. 1417-1428.

[19]. M. Vecchio, R. López-Valcarce, and F. Marcelloni, A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks, *Applied Soft Computing*, Vol. 12, Issue 7, 2012, pp. 1891-1901.

[20]. Data sheet for CC2420 2.4 GHz IEEE 802.15.4/ZigBee RF transceiver (http://www.ti.com/lit/ds/symlink/cc2420.pdf).

[21]. R. P. Peter Corke, Daniela Rus, 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, (http://cmc.cs.dartmouth.edu/cmc/papers/corke:holes.pdf).

[22]. Y. Wang, J. Gao, and J. S. B. Mitchell, Boundary recognition in sensor networks by topological methods, in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, 2006, p. 122.

[23]. M. K. Watfa and S. Commuri, Energy-efficient approaches to coverage holes detection in wireless sensor networks, in *Proceedings of the IEEE International Symposium on Intelligent Control,* 2006, pp. 137-142.