COVERAGE HOLES DISCOVERY ALGORITHM FOR WIRELESS SENSOR NETWORK

WANG QING-SHENG, GAO HAO

1 Assoc Prof, Department of Computer Science and technology, Taiyuan University of Technology, Taiyuan
2 Department of Computer Science and technology, Taiyuan University of Technology, Taiyuan

E-mail: wangqs1308@163.com, gaohao8912@163.com

ABSTRACT

Nonuniform random distribution and exhausted energy of sensor nodes may lead to coverage holes emerged in wireless sensor networks. Aiming at the problems of coverage holes in wireless sensor networks, a geometry-based distributed coverage holes discovery algorithm is proposed in this paper, and it is proved theoretically. The main idea of this algorithm is forming a triangle by sensor node and its two neighbor nodes, calculating circumradius and circumcenter of the triangle and judging existence of coverage holes under the guidance of relevant knowledge of geometric graphics. Simulation results on Matlab platform demonstrate that the algorithm not only can detect coverage holes and boundary nodes effectively, but also has a good performance in terms of discovery accuracy.

Keywords: Wireless Sensor Networks, Coverage Holes Discovery, Distributed Algorithm

1. INTRODUCTION

Wireless Sensor Networks (WSNs) have a wide range of applications in the field of environmental monitoring, traffic management and intelligent building. In these applications, coverage control is its primary concern [1]. In order to make WSNs complete the task of target monitoring and access to information, sensor nodes should effectively cover the monitored area or target. The integrity of the WSNs coverage is the important measure of quality of service [2]. However, with the continuous operation of the network, sensor nodes may die due to random deployment, depletion of sensor power or environment destruction, which makes the network coverage area missing and forms coverage holes. Furthermore, since the sensing radius and communication radius of each sensor node is fixed, random deployment of sensor nodes may lead to form coverage holes which affect the network performance seriously. Therefore, when coverage holes appear in WSNs, it should be detected and found immediately to keep perception of the wireless sensor network communication service quality.

The remaining parts of the paper are organized as follows. Section 2 presents recently related work of coverage holes detection. Section 3 introduces the definitions and assumptions. Section 4 describes a coverage holes discovery approach based on geometric graphics, related theorems and their proof are also presented in this section. Section 5 presents experiment results and analysis, followed by conclusions and future work in Section 6.

2. RELATED WORKS

There are already some methods about coverage holes detection in WSNs. Wang [3] discovers coverage holes by using Voronoi graph and patches coverage holes by mobile nodes. This algorithm is applied to randomly deployed WSNs, but mobile nodes need to consume a large amount of energy. Ghrist and Silva [4, 5] combine homology theory with coverage holes discovery algorithm. However, their method is centralized, which makes it impractical in large-scale sensor networks. When the number of nodes is n, time complexity of this algorithm is O (n5). Buchart [6] establishes the WSNs communication connection diagram, converts communication connection diagram into a planar simplified complex, and presents a coverage holes discovery algorithm based on this model. Kanno [7] establishes the WSNs communication connection diagram by using algebraic topology, this algorithm can be applied to the WSNs which sensor nodes coordinates are unknown. Xin [8] presents a boundary arc and boundary node discovery algorithm according to coverage holes surrounded by boundary arc, and then detects coverage holes. Corke [9] proposes a path density
algorithm to detect coverage holes. It uses the path
density to detect coverage holes by the neighbors of
a dead sensor node. The algorithm can detect
coverage holes remotely, but it requires more time
and power consumption for detecting coverage
holes in practice.

Based on researches mentioned above, we make
further research on coverage holes discovery. A
computational geometry based distributed coverage
holes discovery algorithm is proposed in this paper.
Since each sensor node has the capabilities of local
sensing and communication, it can execute this
algorithm concurrently and detect coverage holes
and boundary nodes of the whole network under the
guidance of relevant knowledge of geometric
graphics.

3. DEFINITIONS AND ASSUMPTIONS

3.1. Definitions

Definition 1 (Neighbor nodes). If sensing range
of A and B is intersecting, then A and B are
neighbor nodes to each other. Especially, if the
distance between node A and node B d(A,B)≤2Rc, A
and B are one-hop neighbor nodes to each other; if
Rc<d(A,B)<2Rc, A and B are two-hop neighbor
nodes to each other [10].

3.2. Assumptions

The algorithm in this paper is developed under
the following assumptions:

1. The network is connected. Each sensor node has
the same initial energy and the same ability of
computation and communication.

2. Sensor nodes can get its own position
information and have unique identification
number.

3. The communication radius Rc of a sensor node is
twice of its sensing radius Rc.

4. Boolean sensing model is adopted to each sensor

4. DISTRIBUTED COVERAGE HOLE
DISCOVERY ALGORITHM

4.1. Theoretical Foundation

Theorem 1. If triangle formed by a node with its
two neighbors is an acute triangle or right triangle,
and circumradius R≤Rc, there are not coverage
holes around the node; otherwise, there are
coverage holes around the node if circumcenter Z is
not covered by any other neighbor of the node.

Proof: Consider the limit of the paper. We only
describe the proof of the case of acute triangle. The
proof of right triangle is similar to the proof of
acute triangle.

Case 1: When the acute triangle is formed by the
node with its two one-hop neighbor, circumcenter Z
must be located inside the acute triangle,
circumradius R≤Rc. So there is common sensing
region around three sensors. It means that there are
not coverage holes around the node.

Case 2: When the acute triangle is formed by the
node with its two two-hop neighbor, if circumradius
R≤Rc, there is common sensing region around three
sensors. Hence there are not coverage holes around
the node. If circumradius R > Rc, there is not
common sensing region around three sensors,
circumcenter Z must be located outside the acute
triangle. It means that there are not coverage holes
around the node.

Case 3: When the acute triangle is formed by the
node with its one-hop neighbor and two-hop
neighbor, proof of case 3 is similar to proof of case
2.

Theorem 2. If triangle formed by a node with its
two neighbors is an obtuse triangle, and
circumradius R≤Rc, there are not coverage holes
around the node; otherwise, if angle subtended at
the node is acute, there are not coverage holes
around node X, if angle subtended at the node is
obtuse, there are coverage holes around the node if
circumcenter Z is not covered by any other
neighbor of the node.

Proof: Case 1: When the obtuse triangle is
formed by the node with its two one-hop neighbor,
if circumradius R≤Rc, then circumcenter Z must be
located inside the obtuse triangle, there is common
sensing region around three sensors. So there are
not coverage holes around the node. If circumradius
R > Rc, circumcenter Z must be located outside the
obtuse triangle, there are coverage holes around the
node if circumcenter Z is not covered by any other
neighbor of the node.

Case 2: When the obtuse triangle is formed by
the node with its two two-hop neighbor, if angle
subtended at the node is acute, there is common
sensing region around three sensors. So there are
not coverage holes around the node. If angle
subtended at the node is obtuse, and circumradius R≤Rc,
there is common sensing region around three
sensors, circumcenter Z must be located outside any
sensing region of three sensor
nodes, and there are coverage holes around the
node.
Case 3: When the obtuse triangle is formed by the node with its one-hop neighbor and two-hop neighbor, proof of case 3 is similar to proof of case 2.

4.2. Algorithm Description

Based on theorems mentioned above, a computational geometry based coverage holes discovery approach is proposed. This approach is distributed, and each sensor node can run the algorithm concurrently. The detail process of our algorithm is described as follows.

Step 1 Select a node randomly in the network, its coordinates are \( (a, b) \);

Step 2 Construct set \( N \) by one-hop and two-hop neighbor nodes of node \( X \);

Step 3 Choose nodes from set \( N \) whose \( y \) coordinate is greater than \( b \), construct set \( N_t \) by these nodes, and sort nodes from set \( N_t \) according to their \( x \) coordinate in ascending order;

Step 4 Choose nodes \( A_i \) and \( A_j \) from set \( N_t \), and \( x \) coordinate of \( A_i \) is less than \( A_j \);

Step 4.1 Compute circumradius \( R \) and circumcenter \( Z \) of the triangle \( XA_iA_j \);

Step 4.2 Judge triangle \( XA_iA_j \) is acute triangle, right triangle or obtuse triangle;

Step 4.3 If (triangle \( XA_iA_j \) is acute triangle or right triangle)

if \( (R < R_s) \) there are not coverage holes around node \( X \);

else there are coverage holes around node \( X \);

Step 4.4 If (triangle \( XA_iA_j \) is obtuse triangle)

if \( (R < R_s) \) there are not coverage holes around node \( X \);

else {if (angle subtended at node \( X \) is acute)

there are not holes around node \( X \);

else {if (circumcenter \( Z \) is covered by any of neighbors of node \( X \))

there are not coverage holes around node \( X \);

else there are coverage holes around node \( X \);}

}

Step 4.5 Delete \( A_i \) from set \( N_t \), loop to step 4 until set \( N_t \) is empty;

Step 5 Choose nodes from set \( N \) whose \( y \) coordinate is less than \( b \), construct set \( N_b \) by these nodes, and sort nodes from set \( N_b \) according to their \( x \) coordinate in ascending order; Set \( N_b \) is performed operations similar to step 4;

Step 6 Choose nodes from set \( N \) whose \( x \) coordinate is greater than \( a \), construct set \( N_r \) by these nodes, and sort nodes from set \( N_r \) according to their \( y \) coordinate in ascending order; Set \( N_r \) is performed operations similar to step 4;

Step 7 Choose nodes from set \( N \) whose \( x \) coordinate is less than \( a \), construct set \( N_l \) by these nodes, and sort nodes from set \( N_l \) according to their \( y \) coordinate in ascending order; Set \( N_l \) is performed operations similar to step 4.

5. SIMULATION RESULTS

In order to verify the effectiveness and performance of our algorithm presented in this work, in this section, our algorithm and another algorithm that is proposed in [12] are performed.

5.1. Algorithm Effectiveness Verification

The experiment on Matlab 7.0 aims at evaluating the efficiency of our method proposed in this paper. Table 1 shows related simulation parameters in detail. All the experiments were implemented on Matlab 7.0, performed on a system with 2.8GHz Pentium D processor with 2GB of RAM, and ran on Windows XP system.

<table>
<thead>
<tr>
<th>Simulation parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network size (m²)</td>
<td>1000*1000</td>
</tr>
<tr>
<td>Sensing radius (m)</td>
<td>100</td>
</tr>
<tr>
<td>Communication radius (m)</td>
<td>200</td>
</tr>
<tr>
<td>the number of deployed nodes</td>
<td>100</td>
</tr>
<tr>
<td>Initial reserved energy of nodes (J)</td>
<td>30</td>
</tr>
<tr>
<td>Energy consumption by sending each packet (J)</td>
<td>1</td>
</tr>
<tr>
<td>Energy consumption by receiving each packet (J)</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Figure 1 shows that the experiment is performed by randomly locating 100 sensor nodes inside a 1000*1000 square. Each sensor node runs the algorithm concurrently, and it can discover coverage holes and boundary nodes of the whole network. There are 4 coverage holes and 17 boundary nodes in the square. The sensing ranges of 17 boundary nodes are red bold as shown in Figure 2.
5.2. Algorithm Performance Evaluation

To evaluate the performance of our approach proposed in this work, it is compared with another algorithm proposed in [12]. The performance of the two algorithms is compared and analyzed in terms of detection accuracy (the number of boundary nodes detected by our algorithm / the number of boundary nodes actually). All experiments are performed by randomly locating {200, 400, 600, 800, 1000} sensor nodes in a square of size of 1000*1000 respectively. The sensing radius of sensor nodes is 40m; communication radius of sensor nodes is 80m.

Figure 3 compares our algorithm and algorithm in [12] about discovery accuracy. As seen from Fig. 2, discovery accuracy of our algorithm is better than those of algorithm in [12] in all experiments. It means that our algorithm has better performance and higher discovery accuracy.

6. CONCLUSIONS AND FUTURE WORK

Aiming at the problems of coverage holes in wireless sensor networks, we propose a computational geometry based on distributed coverage holes discovery algorithm in this work, and it is proved theoretically. This algorithm intends not only to detect the presence of coverage holes and boundary nodes in the network, but also to improve discovery accuracy, which can make algorithm have a better performance. Moreover, after we detecting coverage holes effectively, the next focus of the research is how to patch coverage holes and further improve the network quality of service.

REFERENCES:


