Connectivity-based Distributed Coverage Hole Detection in Wireless Sensor Networks

Feng Yan, Philippe Martins, Laurent Decreusefond

▶ To cite this version:

Connectivity-based Distributed Coverage Hole Detection in Wireless Sensor Networks

Feng Yan, Philippe Martins, Laurent Decreusefond
Network and Computer Science Department
TELECOM ParisTech
Paris, France
Email: {fyan, martins, decreuse@telecom-paristech.fr}

Abstract—**Coverage is considered as an important measure of quality of service provided by a wireless sensor network (WSN). Yet, coverage holes may appear in the target field due to random deployment, depletion of sensor power or sensor destruction. Discovering the boundaries of coverage holes is important for patching the sensor network. In this paper, we adopt two types of simplicial complexes called Čech complex and Rips complex to capture coverage holes and classify coverage holes to be triangular and non-triangular. A distributed algorithm with only connectivity information is proposed for non-triangular holes detection. Some hole boundary nodes are found first and some of them initiate the process to detect coverage holes. Simulation results show that the area percentage of triangular holes is always below 0.1% when the ratio between communication radius and sensing radius of a sensor is two. It is also shown that our algorithm can discover most non-triangular coverage holes.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted significant research attention due to their wide potential applications such as battlefield surveillance, environmental monitoring, intrusion detection etc. One of the fundamental problems that arises in WSNs is coverage. In general, coverage can be considered as the measure of quality of service provided by a WSN [1]. In most applications, we are interested in reliable monitoring of the environment in such a way that there are no holes in the coverage. However, it is difficult to ensure a WSN without coverage holes for several reasons. On the one hand, in order to ensure complete coverage, one possible approach is to deploy sensors precisely according to a regular pattern (hexagon, square grid, rhombus, or equilateral triangle) [2]. However, when the target filed is hostile or unapproachable for human beings, it is impractical to make precise deployment and random deployment can be an alternative. With random deployment, sensors may cluster in some place and leave coverage holes in other places. On the other hand, even if sensors can be precisely deployed at the beginning, a coverage hole can still appear as time goes on. Some sensors can deplete energy faster or can be destroyed. Consequently, it is essential to discover coverage holes in order to ensure complete coverage. Once coverage holes are localized, some mobile sensors can be utilized to patch the holes by making use of boundary information of these holes.

There is already extensive research on the coverage problem in WSNs. These coverage methods can be classified into three categories: location-based, range-based and connectivity-based. In the first category, some methods utilize computational geometry approaches (such as Voronoi diagram and Delaunay triangulations) to discover coverage holes [3] [4] [5]. In some other studies, the authors proved that a sensor node does not border a coverage hole if its sensing border is entirely covered by the sensing ranges of its neighbors [6][7][8]. However, all of these methods require precise information of sensor locations. This requirement substantially limits their applicability since acquiring accurate location information is either expensive or impractical in many settings. Range-based methods attempt to discover coverage holes based on relative distances between neighboring sensors [9][10][11]. Similarly, obtaining precise range between neighbor nodes is also costly. Therefore, connectivity-based schemes are of great interest for us. The pioneer work of Ghrist et al [12] is the first purely connectivity-based coverage hole detection method. They construct the Rips complex corresponding to the connectivity graph of the network and determine the coverage by verifying whether the first homology group of the simplicial complex is trivial. However, their method is centralized, which makes it impractical in large-scale sensor networks. Some distributed algorithms are proposed to implement the above mentioned ideas in [13][14]. But these homology based algorithms are all dependent on the assumption that the communication range of a sensor is smaller than $\sqrt{3}$ times the sensing range of the sensor. And it is possible that a complete coverage network can be mistakenly estimated to have coverage holes. In [15], the authors propose a novel graph theoretical criterion for coverage determination. But they only consider the cases that communication range of a sensor is smaller than two times the sensing range of the sensor.

In this paper, we study 1-coverage problem based only on connectivity information for static WSNs. We adopt Čech complex and Rips complex [12] to capture coverage holes. The relationship between Čech complex and Rips complex in terms of coverage hole is analyzed under different ratios between communication radius and sensing radius of a sensor. We find that when the communication radius is no less than two times sensing radius, if there is a hole in Rips complex, there must be a hole in Čech complex. Based on that, we classify coverage holes to be triangular and non-triangular. A connectivity-based distributed algorithm is proposed to discover non-triangular

holes. And the area percentage of triangular holes is computed by simulations.

The remainder of the paper is organized as follows. In section II, we present our network models and give the relationship between Čech complex and Rips complex. Section III proposes a connectivity-based distributed algorithm to detect non-triangular holes in Rips complex. Performance evaluation of our algorithm is included in section IV. Finally, Section V concludes the paper.

II. MODELS AND ASSUMPTIONS

Considering a collection of stationary sensors (also called nodes) deployed randomly in a planar target field. Each sensor has the capabilities of local sensing and communication. However sensors have only access to connectivity information. We assume that each sensor has a maximum sensing range of R_s and a maximum communication range of R_c. Each sensor can detect any event occurred in its sensing range and communicate with any other sensor within its communication range. We use $G(V, E)$ to denote the connectivity graph of the network, where V is the set of sensors, E is the set of edges connecting sensors. According to former assumptions, there is an edge between any two sensors within Euclidean distance R_c.

In addition, some other assumptions are as follows.

1) There are sensors located on the external boundary of the target field. They are known as fence sensors and other sensors are referred to as internal sensors. Each fence sensor has two fence neighbors.
2) Although sensors are not aware of their locations, every sensor can know whether it is a fence or an internal node by using the mechanisms presented in [10] or other methods as in [15]. In fact, it is a conventional assumption adopted by many existing range-based methods [10][11] or connectivity methods [14][15].
3) The network is connected.

As in [16], we adopt Čech complex and Rips complex to characterize coverage. Here we will give their definitions. For further readings, see [16] and references therein.

Definition 1 (Čech complex). *Given a collection of sets* $\mathcal{U} = \{U_\alpha\}$, *Čech complex of* $\mathcal{U}$, *$\check{C}(\mathcal{U})$, is the abstract simplicial complex whose k-simplices correspond to nonempty intersections of k + 1 distinct elements of* $\mathcal{U}$.

Definition 2 (Rips complex). *Given a set of points* $\mathcal{X} = \{x_\alpha\} \subset \mathbb{R}^n$ *in Euclidean n-space and a fixed radius ϵ, the Rips complex of $\mathcal{X}$, $\mathcal{R}(\mathcal{X})$, is the abstract simplicial complex whose k-simplices correspond to unordered (k + 1)-tuples of points in $\mathcal{X}$ which are pairwise within Euclidean distance ϵ of each other.*

Assume $\mathcal{V} = \{v_i\}$ denotes the set of sensor locations in a WSN and $\mathcal{S} = \{s_i\}$ is the collection of sensing ranges of these sensors, where v_i is the location of the i-th sensor and $s_i = \{x \in \mathbb{R}^2 : \|x - v_i\| \leq R_s\}$ is the sensing range of the i-th sensor. Then according to the definition, the Čech

complex and Rips complex of the WSN can be constructed, which are denoted by $\check{C}_{R_s}(\mathcal{V})$ and $\mathcal{R}_{R_c}(\mathcal{V})$ respectively. As the target field is planar, we do not need to consider simplices with dimension higher than 2 when constructing complex. In this case, it can be seen that the Rips complex of a WSN can be obtained by filling all triangles in the corresponding connectivity graph. An explanation of coverage, Čech complex, Rips complex and connectivity graph of a WSN is given in Figure 1.
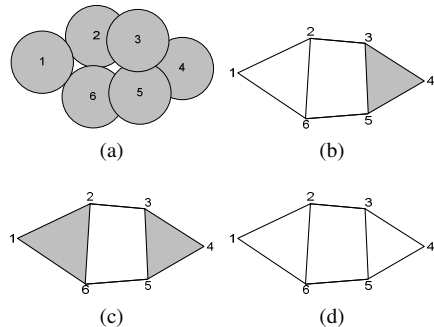

Fig. 1: Example of (a) coverage; (b) Čech complex; (c) Rips complex and (d) connectivity graph.

In Figure 1, there are two coverage holes in the WSN, surrounded by nodes 1, 2, 6 and 2, 3, 5, 6 respectively. And both of the two holes can be found in Čech complex. But only the hole bounded by nodes 2, 3, 5, 6 is found in Rips complex. In fact, as proved in [16], any coverage hole can be found in Čech complex. Furthermore, there are following relations between Čech complex and Rips complex:

$$\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V}) \subset \mathcal{R}_{2R_s}(\mathcal{V}), \text{ if } R_c \leq \sqrt{3}R_s. \quad (1)$$

According to (1), some relationships between Čech complex and Rips complex in terms of coverage hole can be derived as illustrated in the following corollaries. For convenience, define $\gamma = R_c/R_s$.

Corollary 1. *When $\gamma \leq \sqrt{3}$, if there is no hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, there must be no hole in Čech complex $\check{C}_{R_s}(\mathcal{V})$.*

Proof: If there is no hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, it means that $\mathcal{R}_{R_c}(\mathcal{V})$ can be triangulated. Since $\gamma \leq \sqrt{3}$ means $R_c \leq \sqrt{3}R_s$, according to the first inclusion in (1), we can see that $\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V})$. Consequently, Čech complex $\check{C}_{R_s}(\mathcal{V})$ can also be triangulated. And when $R_c \leq \sqrt{3}R_s$, each triangle must be covered by the sensing range of its vertex nodes [16]. So there is no hole in $\check{C}_{R_s}(\mathcal{V})$. ∎

Corollary 2. *When $\gamma \geq 2$, if there is a hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, there must be a hole in Čech complex $\check{C}_{R_s}(\mathcal{V})$.*

Proof: If there is a hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, there must be a cycle with more than three edges in $\mathcal{R}_{R_c}(\mathcal{V})$ that can not be triangulated, as the cycle $\{2, 3, 5, 6, 2\}$ in Figure 1(c). Since $\gamma \geq 2$ means $R_c \geq 2R_s$, according to the second inclusion in (1), we can see that $\check{C}_{R_s}(\mathcal{V}) \subset \mathcal{R}_{2R_s}(\mathcal{V}) \subset \mathcal{R}_{R_c}(\mathcal{V})$.

Consequently, there must also be a non-triangulable cycle in $\check{C}_{R_s}(\mathcal{V})$. And there is a coverage hole in the cycle. ∎

Corollary 3. *When $\sqrt{3} < \gamma < 2$, there is no guarantee relation between Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$ and Čech complex $\check{C}_{R_s}(\mathcal{V})$ in terms of holes.*

Proof: This corollary can be directly derived from Corollary 1 and 2. ∎

From Corollary 1, a sufficient condition for coverage verification can be derived, which has been figured out in [14]. But it is not a necessary condition. It is possible that there is no hole in $\check{C}_{R_s}(\mathcal{V})$, while there is a hole in $\mathcal{R}_{R_c}(\mathcal{V})$. From Corollary 2, we can find a necessary condition for the existence of a hole in $\check{C}_{R_s}(\mathcal{V})$. Corollary 3 indicates that when there is no hole in $\mathcal{R}_{R_c}(\mathcal{V})$, it is possible that there is a hole in $\check{C}_{R_s}(\mathcal{V})$. When there is a hole in $\mathcal{R}_{R_c}(\mathcal{V})$, it is also possible that $\check{C}_{R_s}(\mathcal{V})$ contains no hole.

From the discussion above, we find that there are two types of coverage hole: triangular hole and non-triangular hole. The formal definition is given as follows.

Definition 3 (Triangular and non-triangular hole). *For any triangle in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$ of a WSN, if the triangle is not covered by sensors in the WSN, the hole inside the triangle is considered as a triangular hole, any other holes are non-triangular.*

In this work, we consider the case $\gamma \geq 2$ and aim at detecting and localizing coverage holes based on connectivity information. Since it is impossible to construct Čech complex for a WSN without location information, we use Rips complex to capture coverage holes. We construct the Rips complex of a WSN based on connectivity information and propose a distributed algorithm to discover the minimum cycle that bounds a hole in Rips complex. As discussed above, only non-triangular holes can be found using this algorithm. For triangular hole, it is impossible to find them based only on connectivity information. We compute the average number and area percentage of triangular holes by simulations. If the area percentage is within an acceptable level, we can neglect the triangular holes. For convenience, we choose $\gamma = 2$. But our algorithm is also applicable for the case $\gamma > 2$.

III. DISTRIBUTED COVERAGE HOLE DETECTION ALGORITHM

In this section, we will propose a connectivity-based distributed coverage hole detection algorithm to find boundaries of holes in Rips complex. The general steps of the algorithm are given first and the details of each step are presented in the following parts.

The entire process of our algorithm can be summarized as follows.

1) Each sensor gets its 1- and 2-hop neighbors information and constructs a neighbor graph.
2) Based on the neighbor graph, each node can determine whether it can be a hole boundary node or not by checking whether there exists a Hamilton cycle in its

neighbor graph. If there exists, the node is assumed to be a non-hole boundary node. Otherwise, it is assumed to be hole boundary.
3) When each node has determined whether it is a hole boundary node or not, it can broadcast this information to its neighbors. Then each node knows the status of its neighbors. After that, some nodes can initiate hole discovery process to find cycles bounding holes.
4) It is possible that some cycles found in step 3 are not minimum or some cycles bound the same hole. So it is necessary to minimize and make choices among all cycles found.

Each step is described in more detail in the following parts.

A. Neighbor discovery

In this step, each node needs to obtain all its 1-hop and 2-hop neighbors information. This can be easily achieved by two broadcasts of hello message. In the first one, each node broadcasts its id. When it gets all the ids of its 1-hop neighbors, each node continues to broadcast a hello message containing the ids of its 1-hop neighbors. After receiving the neighbor list of its neighbors, each node can construct its neighbor graph. Assume ID_i denotes the id of i-th node, $G_i(V_i, E_i)$ is the neighbor graph of node ID_i, where V_i denotes the set containing ids of its 1-hop neighbors and E_i is the set containing all the edges between its 1-hop neighbors. Notice that ID_i is not included in V_i.

B. Hole boundary nodes discovery

After each node knows its one and two hop neighbors, it needs to determine whether it is a hole boundary node or not. we propose a method based on Hamilton cycle for the verification. For each internal node ID_i, if it has less than three neighbors or there exists not a Hamilton cycle in its neighbor graph $G_i(V_i, E_i)$, this node is considered to be hole boundary. Otherwise, it is considered as a non-hole boundary node. For fence nodes, the node itself should be added to its neighbor graph and verify whether there exists Hamilton cycle in the new graph. So the problem here is to find a method to search Hamilton cycle in a graph. In general, it is a NP-complete problem. However, considering the graph in our problem is the neighbor graph of a node, it usually contains small number of nodes and so it is possible to use exhaustive search method. As an alternative, a more efficient algorithm is proposed in [17] for Hamiltonicity detection with complexity $O(1.657^n)$. Figure 2(a) is an example showing the detection results. Nodes denoted by squares are hole boundary nodes recognized by the efficient method. We can see from the figure that some node (such as 8) bordering a hole is not recognized as hole boundary.

C. Hole discovery

After determining the status (hole boundary or not), each hole boundary node broadcasts a hello message indicating its status. When each node has received all the broadcasting message from its neighbors, it knows the status of its 1-hop
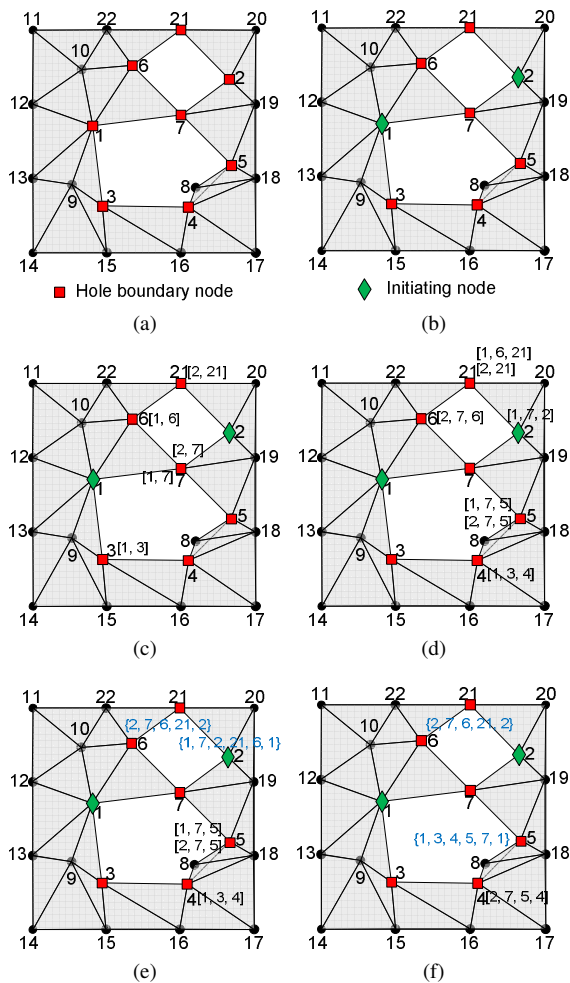
Fig. 2: The process of algorithm. (a) hole boundary node discovery; (b) initiating node selection; (c) hole discovery: step 1; (d) hole discovery: middle of step 2; (e) hole discovery: cycles found; (f) cycles selection.

TABLE I: The Structure of Message

node_seq	type
ids	0, 1 or 2

when they receive a broadcasting message, it is possible that some hole can not be discovered. But if all nodes, no matter hole boundary or not, broadcast the message, the message complexity will be high. As a trade off, we set a probability p for non-hole boundary nodes. When a non-hole boundary node receives a broadcast message with type 0, it will broadcast the message with probability $(1 - p)$ and set the message type as 1. When a non-hole boundary node receives a type 1 message, it will broadcast with probability $(1 - 2p)$ and set the message type as 2. If a non-hole boundary node receives a type 2 message, it will neglect it. Any hole boundary node will set the message type as 0 when it broadcasts a message. In addition, for any fence node, if it is not hole boundary, it will not forward any message. When broadcasting a message, each node attaches its id to the node_seq of the message sequentially. In this way, the message contains all the ids of nodes from an initiating node to the current one.

We assume in the process, if some neighbors of one node broadcast messages at the same time, the node can correctly receive these messages simultaneously. For each message, the node needs to verify whether it is a message containing new information. If so, the node keeps the message; otherwise, it deletes the message. After that, among all kept messages, the node needs to verify whether a cycle is found. For any two messages, if the first ids in the node_seqs are the same and the last ids are different and are not neighbors, it is assumed a cycle is found. Then the node constructs the cycle and keeps it. It is possible that the cycle found is not minimum, which can be processed by the next step illustrated in Part D. Meanwhile, the node deletes the two messages. After that, the node continues to check whether the new found cycle is different from cycles found before. If two cycles are considered to border the same hole, the shorter cycle will be kept. After the process, if there is still some message, the node will broadcast the message as illustrated in the last paragraph. In addition, in order to reduce the message complexity, when any initiating node receives a message initiated by another one, it will delete the message if the other initiator has higher id than itself.

We consider the example in Figure 2 to explain the process. Nodes 1 and 2 are selected as initiators as their ids are smaller than ids of all their hole boundary neighbors. So they broadcast a message containing their ids respectively and set the message type as 0 because they are both hole boundary nodes. After their broadcastings, their neighbors can receive the message. For example, neighbors of node 1 (nodes 3, 6, 7, 9, 10, 12, 13) receive the message. As nodes 12 and 13 are non-hole boundary fence nodes, they will not forward message. Nodes 9 and 10 are non-hole boundary internal nodes, they can choose to broadcast with probability $(1 - p)$. If they decide not to

neighbors. The next step is to find the cycles bounding a hole. Since the algorithm is distributed, there is no central unit to select some nodes to initiate the process. In our algorithm, we assume that if one hole boundary node has no hole boundary neighbor or has the minimum id among all its hole boundary neighbors, it is selected by itself to initiate the process. Figure 2(b) is the result of selection. Nodes denoted by diamond (nodes 1 and 2) are initiators.

Each selected node initiates the process by broadcasting a message. The structure of the message is shown in Table I. The element node_seq contains the ids of nodes that have received and forwarded the message. The type indicates the type of message. We define three types of messages, denoted by 0, 1 and 2. 0 indicates that the last node in node_seq is hole boundary. 1 indicates that the last node is non-hole boundary and 2 indicates that the last two nodes are non-hole boundary.

As seen from Figure 2(a), some nodes which are on the boundary of a cycle are not recognized as hole boundary. Given that only hole boundary nodes continue broadcasting

broadcast in this round, they can neglect the message. Nodes 3, 6 and 7 are hole boundary nodes, and the message is new for them. So each of them keeps the message and add its id to the node_seq of the message and set message type as 0 (the message type is not shown in the figure). Similarly, neighbors of node 2 do the same process. The result of this round is shown in Figure 2(c). It can be seen that node 7 keeps the message from nodes 1 and 2. In the second round, nodes 3, 6, 7 and 21 will broadcast. We assume that they utilize proper MAC protocol to avoid collision. Nodes 6, 7 and 21 broadcast sequentially. After nodes 6 and 7 broadcasting, the result is shown in Figure 2(d). When node 6 broadcasts, the message is not new for node 1, so node 1 neglects it. When node 7 broadcasts, the message with node_seq [2, 7] is new for node 1. But as node 1 is an initiator and the message is initiated by node 2 whose id is larger than 1, node 1 also neglects the message from node 7. After node 21 broadcasts, node 6 finds a cycle 2, 7, 6, 21, 2 and node 2 discovers another one $\{1, 7, 2, 21, 6, 1\}$, as shown in Figure 2(e). We can see that the cycle found by node 2 is not minimum and it bounds the same hole as cycle $\{2, 7, 6, 21, 2\}$. So we need another step to choose cycles which is presented in next part. The process continues and finally another cycle $\{1, 3, 4, 5, 7, 1\}$ is also found, as presented in Figure 2(f). We can see from Figure 2(f) that the cycle $\{1, 3, 4, 5, 7, 1\}$ is minimum but it is not the closest to hole as node 8 is inside the cycle. With only connectivity information, we can not determine weather a cycle is the closest to a hole or not.

D. Cycles selection

When any cycle is found, it is possible that the cycle is not minimum or borders the same hole with other cycles or even is not really bordering a hole. So it is necessary to analyze these cycles and make choices. This can be realized distributively by each node in the cycle. When some node discovers a cycle, it first checks whether there exists shorter path between itself and other nodes in the cycle. For each node in the cycle, let S_1 denote the set of nodes with more than one hops distance from the node and S_2 denote the set of nodes with more than two hops away. In addition, we use N_i^1 and N_i^2 to denote the set of node i's one hop and two hop neighbors. If S_1 has intersection with N_i^1, it means some one hop neighbor of the node is in the cycle with more than one hops away from the node. So the node can shorter the cycle by connecting itself with that neighbor and continues to check whether S_2 has intersection with N_i^2 in the new cycle. After the verification, if the node finds the cycle borders the same hole as another cycle found before, it can keep the cycle with shorter length and delete another one. Otherwise, it sends the cycle to another neighbor. For example, in Figure 2(e), node 7 can find the cycle is not minimum. Since node 6 is its one hop neighbor but is two hops away in the cycle. So node 7 can shorter the cycle to be $\{7, 6, 21, 2, 7\}$. Finally, nodes in the cycle will discover the cycle $\{7, 6, 21, 2, 7\}$ is the same as the one found by node 6. So only one is kept as shown in Figure 2(f). After cycles are found, some patching method can be adopted to fill the

coverage holes, which is not the focus of this paper.

IV. SIMULATIONS AND PERFORMANCE EVALUATION

In this section, we will present the average area percentage of triangular holes in the target field. Furthermore, the probability that our algorithm fails to discover non-triangular holes is given.

First, we give parameter settings in our simulations. The target field is assume to be a 100×100 m^2 square area. The sensing radius R_s of each node is 10 meters. The communication radius R_c is set to be 20 meters and so $\gamma = 2$. There are fence sensors locating along the edges of the square with 20 meters distance between neighbors. Other internal sensors are randomly distributed in the area according to Poisson point process with intensity λ.

A. Performance related to triangular holes

The area percentage of triangular holes under different λ are shown in Figure 3. For each λ, 500 simulations are implemented. It can be seen that the area percentage of these holes increases first and decreases later with the increase of λ. And in all cases, the area percentage is always below 0.03%. It means that in this case, if all non-triangular holes can be found and patched, then more than 99.97% of the target field can be covered, which is acceptable in most scenarios.
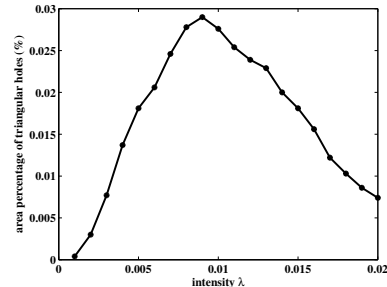

Fig. 3: area percentage of triangular holes.

B. Performance of the algorithm

Simulations of the algorithm are performed with MATLAB in Windows 7. In order to evaluate the performance of our algorithm, we use a publicly available package for computing homology of simplicial complex known as Jplex [18]. Jplex can be installed in MATLAB. For each simulation, we use Jplex to compute the number of holes in Rips complex and compare it with the result obtained by our algorithm to check whether there is any missing hole.

We set λ to be 0.006, 0.008 and 0.010 to represent sparse, moderate and dense WSNs respectively. In addition, we set the parameter p to be 0.2 and 0.4. For each case, 500 simulations are performed and the average probability that the algorithm fails to detect a hole is presented in Figure 4. It can be seen from Figure 4(a) that when p is certain, the failure probability increases with the increase of λ from 0.006 to 0.010. It is because when λ is small, the network is relatively sparse and

nearly all the nodes bordering a hole can be recognized. These nodes broadcast messages with probability 1 in the process. But when λ is high, the network is relatively dense and in this case, some nodes bordering a hole are not recognized. And these nodes broadcast messages with probability $(1 - p)$ or $(1 - 2p)$ as illustrated in part III.C. This may result in failure of detecting some holes. We can also find that the failure probabilities are both below 5% when p is 0.2 and λ is 0.006 or 0.008. In addition, from Figure 4(b) we can see that when λ is higher, the parameter p has larger impact on the failure probability. When λ is 0.8, failure probability increases by 0.015 when p changes from 0.2 to 0.4. And in the case λ is 0.010, failure probability increases by 0.02. Simulation results show that the algorithm is more applicable for non-dense WSN.
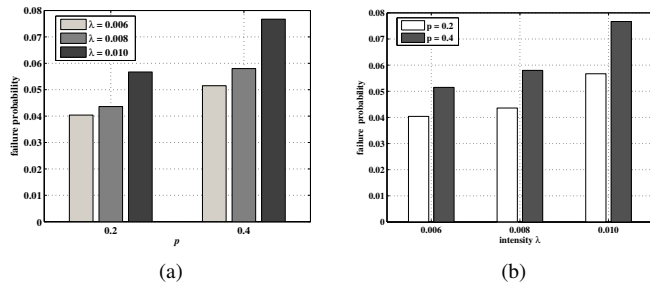

Fig. 4: (a) error probability vs parameter p; (b) error probability vs intensity λ.

V. Conclusion and future work

In this paper, we adopt two types of simplicial complex called Čech complex and Rips complex to capture coverage holes of a WSN. The relationship between Čech complex and Rips complex in terms of coverage holes is first analyzed under different ratios between communication radius and sensing radius of a sensor. Based on that, we define two types of coverage holes: triangular and non-triangular hole. When the ratio between communication range and sensing range of a sensor is no less than 2, it is impossible to discover triangular holes with only connectivity information. But it is possible to find non-triangular holes in this case. So a connectivity-based distributed algorithm has been proposed to detect non-triangular holes. Simulation results show that the area percentage of triangular holes is very small and our algorithms can

discover most non-triangular holes. As for future work, we will derive an analytical expression for the area percentage of triangular holes under different ratios between communication radius and sensing radius.

References

[1] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava., "Coverage problems in wireless ad-hoc sensor networks," in *Infocom'01*, 2001, pp. 1380 – 1387.

[2] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *In ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2006, pp. 131–142.

[3] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Infocom'04*, 2004, pp. 2458–2468.

[4] G. Wang, G. Cao, and T. F. L. Porta, "Movement-assisted sensor deployment," in *Infocom'04*, 2004, pp. 2469–2479.

[5] C. Zhang, Y. Zhang, and Y. Fang, "Localized algorithms for coverage boundary detection in wireless sensor networks," *Wireless networks*, vol. 15, no. 1, pp. 3–20, 2009.

[6] C. Huang and Y. Tseng, "The coverage problem in a wireless sensor network," in *WSNA'03 Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, San Diego, California, USA, sep 2003, pp. 115–121.

[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *SenSys'03: Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003, pp. 28–39.

[8] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89–124, 2005.

[9] C. Zhang, Y. Zhang, and Y. Fang, "Detecting coverage boundary nodes in wireless sensor networks," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control, ICNSC '06*, 2006.

[10] Y. Bejerano, "Simple and efficient k-coverage verification without location information," in *Infocom'08*, 2008, pp. 897–905.

[11] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," in *Proceedings of ACM Mobicom'09*, 2009, pp. 169–180.

[12] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Fourth International Conference on Information Processing in Sensor Networks (IPSN05)*, 2005, pp. 254–260.

[13] A. Muhammad and M. Egerstedt, "Control using higher order laplacians in network topologies," in *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

[14] A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification in sensor networks without location information," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1837 – 1849, 2010.

[15] D. Dong, Y. Liu, K. Liu, and X. Liao, "Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches," in *ICDCS'10*, 2010, pp. 106–115.

[16] V. de Silva, R.Ghrist, and A. Muhammad, "Blind swarms for coverage in 2-d," in *Proceedings of Robotics: Science and Systems*, June 2005.

[17] A. Bjorklund, "Determinant sums for undirected hamiltonicity," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010.

[18] Jplex home page. [Online]. Available: http://comptop.stanford.edu/u/programs/jplex/index.html