Fractured Voronoi Segments: Topology Discovery for Wireless Sensor Networks

Jiliang Wang
CSE Department
Hong Kong University of Science and Technology
aliang@cse.ust.hk

Mo Li
Fok Ying Tung Graduate School
Hong Kong University of Science and Technology
limo@cse.ust.hk

Yunhao Liu
CSE Department
Hong Kong University of Science and Technology
liu@cse.ust.hk

Abstract*—Wireless sensor networks are deployed in various territories executing different tasks. In many applications, it is very useful to understand their topological characteristics. This paper studies the problem of discovering the topological properties of a sensor network such as boundaries and holes. Previous works have revealed that, such a problem could be addressed with knowledge of node locations, measures of inter-distances, or ideal assumptions of particular communication models, e.g., unit disk graph model. In this work, however, we explore the possibility of discovering sensor network topology merely with connectivity information. We propose a virtual voronoi diagram approach to detect both the inner and outer boundaries of a sensor network. We do not rely on any communication models, yet any geometric knowledge of the network. Compared with previous connectivity based approaches, we further release the assumption of regular wireless signals. Our approach works even for anisotropic network with irregular wireless links. We design our approach to be light-weight, preventing frequent global operations that have been intensively used in previous designs. We conduct intensive simulations in networks of different topologies with different node degrees and densities, and containing various signal irregularities. The results validate the effectiveness and efficiency of our approach.*

1. Introduction

Wireless Sensor Networks (WSNs) have rapidly developed during the past years. Current electronic and communicational techniques have made it possible to deploy sensor networks in a very large scale with a large number of individual sensor nodes cooperatively working in various applications. To efficiently manipulate such huge and complex networks, it becomes extremely important to macroscopically abstract the geometric and topological features, such as network localization, topology discovery, sensor clustering and etc [4, 6, 8, 11]. This work studies the problem of topology discovery, e.g., boundary and hole detection, of the underlying sensor network.

Topology discovery has been considered essential in large scale WSNs for many reasons. (1) The topological information such as boundaries acts as basis for implementing many upper layer networking operations. By solely knowing the network boundaries, researchers are able to step further in developing simpler, more efficient and scalable mechanisms like range-free localization [13] and topological routing [3], which are designed adaptive to the intrinsic topology properties of sensor networks. (2) The topological information is also of great importance while designing operations in WSNs and refining the structure of the network. For instance, in the deployment of WSNs, network boundaries indicate those void areas where sensors can not cover and are not connected [1]. Such information can be used to guide the redeployment of sensors to satisfy application specific needs, e.g., extending the coverage area or adjusting the network density. Mobile sensor networks can also use the topological information to reconstruct the network to achieve a better coverage [18]. (3) A WSN is inherently bound with the underlying physical environment it observes. Thus the boundary information of a sensor network indeed reflects important characteristics of its tightly related physical environment. For example, the boundaries of the inner void areas in the sensor network usually indicate obstacles that are physically inaccessible or isolated in communications. Such information is especially important for applications such as environment monitoring, event detection and navigation, etc.

Existing works. As we can see, topological features play an important role in WSNs, many efforts have been made to address the issue of topology discovery, especially for boundary recognition and hole detection, driven by such a wide range of applications.

Early works rely on sensor locations to detect the network boundaries. Having the locations of sensor nodes indeed provides us rich geometric information that largely facilitates the boundary recognition. Fang et al. observe that when a packet is geographically forwarded, it will get stuck at the nodes on network boundaries. Thus they are able to identify the boundaries by detecting the stuck nodes [5]. They assume the availability of locations and the unit disk graph (UDG) communication model. Unfortunately, subject to the WSN deployment, location information is not always available [14] and precise [20]. Yet, the requirement of location information limits the application of boundary recognition, e.g., the range-free localization approach [13] takes the network boundaries as inputs before the sensor locations can be obtained.

To accommodate the unavailability of location information, more works are proposed in the location free context. Fekete et al. [7] observe the fact that in a dense network, nodes near holes in the network have a smaller degree compared with those inside the network. The proposed algorithm, however, works only for those networks where nodes are uniform distributed, and of very high density (degree > 100). They further report an algorithm [12] that

978-1-4244-7489-9/10/$26.00 ©2010 IEEE

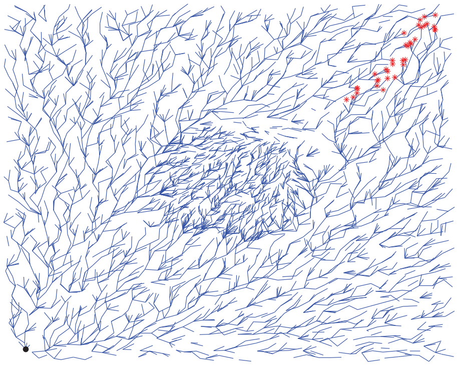


Figure 1. The boundary recognition algorithm of Wang et al. [19] misreports cut pairs under link irregularities.

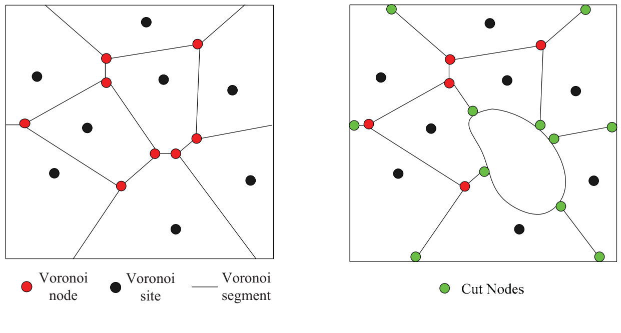


Figure 2. (a) The voronoi diagram divides the plane into subregions, and (b) The voronoi segments are blocked by network boundaries (at green points).

detects the network boundaries by searching the "flower" structures. By iteratively augmenting the interior node set, their algorithm is able to reach the network boundaries. Such an algorithm does not need the location information but assumes the d-quasi UDG model [2] to be valid. A more recent work by Saukh et al. designs several node connectivity patterns and detects the boundaries by finding the weak and strong patterns in the sensor network [15]. Similarly, the UDG or d-quasi UDG model is presumed as a *de facto* wireless communication model in their work. Funke develops a method for boundary recognition that utilizes only connectivity information [9, 10, 17]. The main idea is to propagate from a root node and build iso-contours based on the hop count from the root. A heuristic is that the contours are broken when they meet the network inner boundaries. The proposed approach identifies the nodes close to the boundaries without connecting them in a meaningful way. Its correctness is proven under the assumption of UDG model. Nevertheless, as widely proven by practical experiment results [16-18], the UDG or quasi UDG model is far from accurately approximating the physical behaviors in node communication. Thus it is still questionable how realistic to apply the above theoretical results practically.

Towards a more practical solution with less critical assumptions, Wang et al. present a topological method for boundary recognition solely with connectivity information, which does not assume any specific node communication model [19]. Their approach explores the fact that the shortest path tree splits when meeting the network boundaries. By detecting the cut pairs and compositing all inner holes into one, their approach first finds the coarse boundary. Finally, the fine grained boundaries can be obtained by refining the coarse boundary. Without relying on any geometric dependence, their approach steps further towards real applicability in practical sensor networks that are usually deployed in constraint environment where the geometric information is difficult to access and node communication style is difficult to model. Such an approach, on the other hand, has its own drawbacks. Many rounds of global operations are conducted in their approach, in building the shortest path tree, finding the coarse boundary, refining the boundaries, and the like. Those global operations inevitably introduce huge traffic burden to the resource constraint sensor networks. Their approach is invalidated when the radio signals of sensors become irregular, i.e., the radio signal ranges are different for different sensors or on different directions of a sensor. As figure 1 depicts, the shortest path tree built by Wang's algorithm misreports many cut pairs as shown by red nodes in a network of sensors with irregular signal ranges. Those sensors with smaller radio signal ranges in the central area are taken as a large void by the misreported cut pairs.

The goal and contributions. This paper studies the network topological features, more specifically, boundaries and holes in the network which separate the connected sensor network component from the void field. Indeed, these topological features of a network are not related to any specific geometric embeddings. This enables us to design topological approaches for recognizing the network boundaries and holes inside, like Wang's work, valid for various situations where the geometric information is difficult to access. In this work, we propose a Virtual Voronoi Diagram (VVD) based approach, for topological boundary recognition in wireless sensor networks.

This approach is designed to be more *effective*, detecting both the inner and outer boundaries of the targeted network. It tolerates network irregularities. According to our simulation results, even when the network links are not uniform, our approach achieves accurate boundary recognition. None previous studies consider this problem, and as a matter of fact, most existing approaches fail under such network irregularities.

This approach is designed to be more *efficient*, with much reduced network traffic. It limits most of its operations within a local area such that the traffic will not be propagated to the entire network, reducing the overhead.

This approach is designed to be more *applicable*, lowering the application barriers with minimum requirements. Our approach prevents to use the geometric information like distance measures or node positions. We do not assume any simplified node communication models like UDG or quasi UDG models. The network links are not necessarily uniform. Different sensor nodes may have different communication scopes.

Paper organization. The rest of this paper is organized as follows. In Section 2, we present an overview of our design. In Section 3, we elaborate our protocol in details. We prove the correctness of the principle of our protocol in Section 4. We discuss related issues of our protocol in Section 5, and in Section 6 we conduct intensive simulations to validate our design. Finally, we conclude this work in Section 7.

2. Overview

We design a Virtual Voronoi Diagram (VVD) based approach to detect the inner and outer boundaries of a wireless sensor network. Like existing works did, we assume symmetric links between any pair of sensors in the network. Indeed, for an asymmetric network graph, we can easily reduce it into a symmetric graph by eliminating the unidirectional links. We also assume reliable message delivery in the network. This can be easily achieved by reliable communication techniques on those links, e.g., acknowledgements and retransmissions. Unlike most existing works, we do not assume any prior knowledge of sensor locations; neither the communication model to be UDG or quasi UDG model. The radio signals of sensors are not required to be regular.

Our approach is inspired by the observation that if we build a voronoi diagram on the network connectivity graph, the network boundaries (the outer boundary of the network or inner boundaries around voids inside the network) will block the segments in such a graph. As figure 2 (a) depicts, a voronoi diagram is built according to a specified set of site points. It is a subdivision of the plane determined by the distances between each pair of the site points. Each site point is denoted by a black dot in figure 2 (a). The points that have equal distance to two or more site points compose the voronoi segments. Those points equidistant to 3 or more site points are defined as voronoi nodes, denoted as red dots in the figure. The voronoi cell of a site point p consists of points that are closer to p. More formally, denote the set of voronoi sites by $S = \{s_i \mid i=1, 2, \ldots, k\}$ and the distance between two point s and t by $d(s, t)$. The voronoi cell of each site s_i is then denoted as $Cell_i = \{p \mid d(s_i, p) \le d(s_j, p)$ for any $j \neq i\}$. The voronoi segments are the set of points $\{p \mid\ p \in\ Cell_i \cap\ Cell_j,\ where\ i \neq j\}$ and the voronoi nodes are the set of points $\{p \mid \exists\ i \neq j \neq k,\ d(s_i, p) = d(s_j, p) = d(s_k, p) \le d(s_l, p)$ for any $l \neq i \neq\ j \neq k\}$.

The boundaries of the network will block the voronoi segments and make them disconnected. As figure 2 (b) depicts, the green nodes, which are denoted by cut nodes, reside at the place where the voronoi segments are blocked. Hereby, by identifying the cut nodes, i.e., where the voronoi segments are blocked, we are aware of the network boundaries. By connecting the adjacent cut nodes, we can first obtain coarse boundaries. Utilizing the voronoi cells, our VVD approach automatically generates separated coarse boundaries around different voids as well as the outer boundary of the network. After that, by refining the coarse boundaries, VVD can reach the real boundaries of the network. Most of the operations in VVD are local, strictly limited by voronoi cells, preventing unnecessary traffic generation over the entire network.

3. Protocol Description

The key idea of VVD is that first we find some featuring points on the boundaries, and then by connecting the featuring points with voronoi sites we are able to obtain the coarse skeleton of the boundaries. Finally by refining the coarse skeleton, we recognize the real boundaries.

We elaborate VVD in steps and show the details of each step in the following subsections. In figure 3, we illustrate the execution of the protocol on an example network. As depicted in figure 3 (a), we execute our VVD on a 2500 node network containing a concave void area inside. It is a sparse network with an average degree of around 8.

3.1 Building Voronoi Diagram

The basis of VVD is building a voronoi diagram. In order to build the voronoi diagram, several sensor nodes in the network are selected as voronoi sites. A randomized approach can be used to select sensor nodes as voronoi sites. As the number of sites can be determined before the selection, a constant number p standing for the selection probability between 0 and 1 can be calculated. The number p is then flooded into the network from the sink to inform all nodes in the network. Every node generates whether it is indentified as a voronoi site according to the probability p. Such an approach ensures that the expected number of selected voronoi sites is equal to the specified number and the selected nodes are uniformly distributed in network.

Base on the selected voronoi sites, VVD builds the voronoi diagram. As there are no location or distance measures input to our algorithm, VVD utilizes the hop count information to approximate the distance when building the voronoi diagram.

Each voronoi site floods a *Build Voronoi Diagram Message* (*BVDMsg*) within the network that records the ID of the voronoi site and the hop count to it. Other sensors on receiving the *BVDMsg* maintain a distance table that records the ID of each received voronoi site and updates its shortest distance to the site. The *BVDMsg* is only relayed by a sensor when it contains a shorter distance from the same voronoi site. Each sensor checks the distance table to determine the number of the closest voronoi sites. Sensors with two or more equidistant closest voronoi sites claim themselves to be on the voronoi segments. Sensors with three or more equidistant closest voronoi sites claim to be the voronoi nodes. Figure 3 (a) depicts the voronoi diagram built at this step. The nodes on voronoi segments are green dotted.

3.2 Detecting Cut Nodes

According to our definition, the cut nodes are those nodes where the network boundaries block the voronoi segments. As depicted in figure 3 (c), it is easy to find that cut nodes are resulted from the discontinuity of the voronoi segments on the network boundaries. A specific feature of a cut node is that it has the local maximal distance to some node on the same voronoi segment. By exploring such a specific feature, we develop a method for VVD to detect the cut nodes on the voronoi segments.

After the voronoi diagram is built, each voronoi node floods a *Cruise Message* (*CRUISEMsg*) within the voronoi segments conjunct at itself. The *CRUISEMsg* records the ID of the voronoi node and counts the hops to it. Those nodes on each voronoi seg ment update their distances to the two voronoi nodes on the two ends of the segment according to the *CRUISEMsg*. If the voronoi

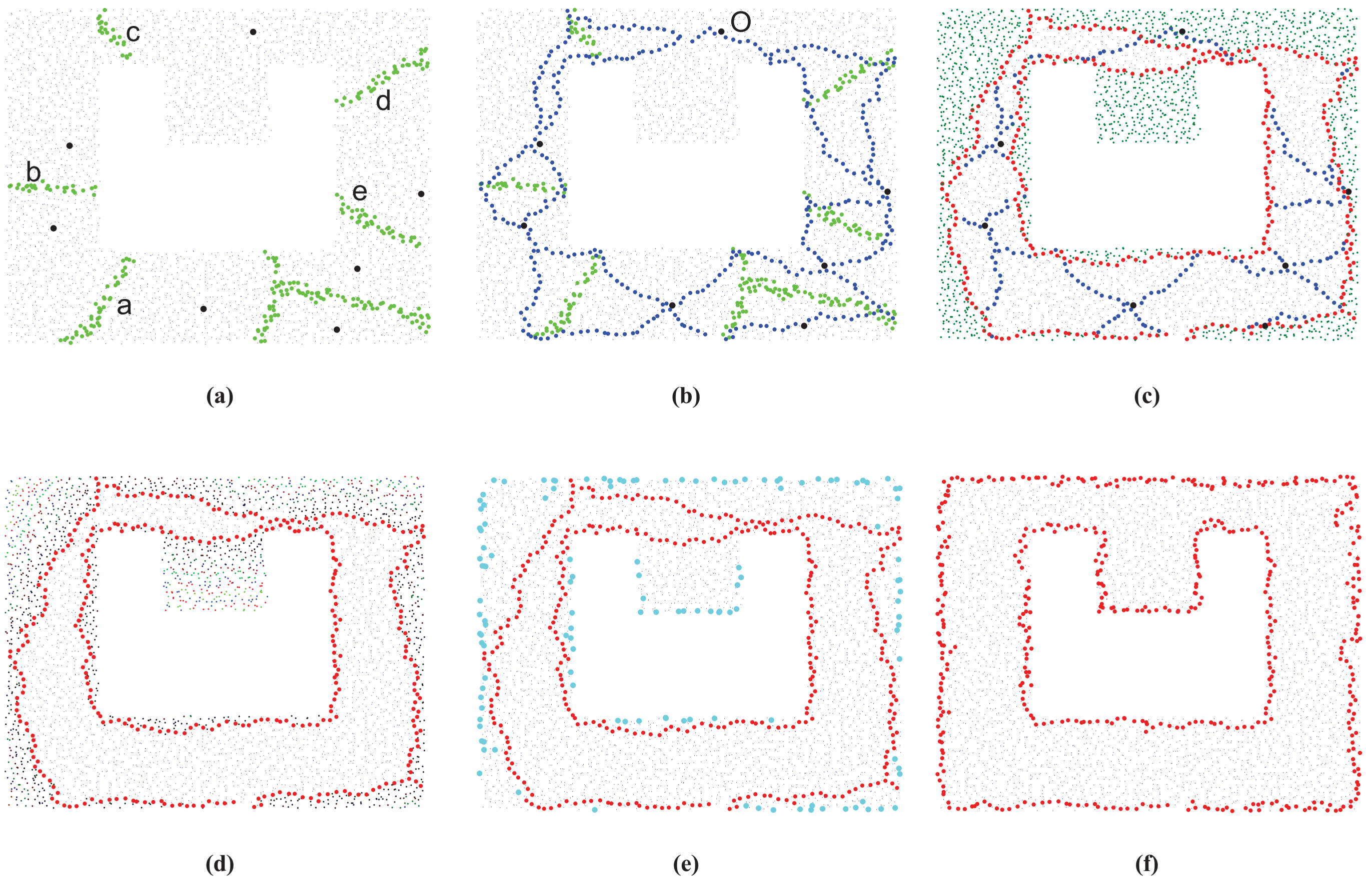


Figure 3: An example of the execution of VVD in a sparse network of a void with an average node degree of around 8. (a) Building the voronoi diagram; (b) Connecting adjacent cut nodes with the voronoi site in each voronoi cell; (c) Building the restricted area for limited flooding operations; (c) Concatenating adjacent cut nodes to form the coarse boundaries; (d) Flooding the restricted areas to detect extremal nodes on network boundaries; (e) Finding the extremal nodes; (f) Refining coarse boundaries to be tight.

segment is not blocked by a network boundary, the *CRUISEMsg* from one voronoi node at one end of the voronoi segment can reach the other voronoi node. Otherwise, the message will stop at some node on the voronoi segment with the maximal distance on the voronoi segment. Those nodes where the *CRUISEMsgs* stop are detected as cut nodes. At this step, each voronoi node only needs to flood one *CRUISEMsg* to voronoi segments conjunct at itself in order to detect all cut nodes.

One special case at this step is that there might be the voronoi segment neither of whose two endpoints is a voronoi node. Such a case indeed implies that both two endpoints of the voronoi segment are cut nodes. Examples can be found in figure 3 (a) where the voronoi segments denoted *a*, *b*, *c*, *d* and *e* are all of this case. Therefore, in such a case the *CRUISEMsgs* will not be flooded on the voronoi segment. Instead, we let the intermediate nodes on the voronoi segment flood a *Find Voronoi Node Message* (*FVNMsg*) to the two ends of the segment. It is obvious that the two endpoints of the voronoi segment both have local maximal hop distance to the sender and they will claim to be cut nodes.

3.3 Connecting Cut Nodes with Sites

In the process of building the voronoi diagram, each sensor node receives the *BVDMsg* from the closest voronoi sites. Thus multiple shortest path trees rooted at the voronoi sites have been built in different voronoi cells. Each node knows its hop distance to its closest voronoi site and maintains a pointer to the parent node on the path from itself to the closest voronoi site. Nodes on voronoi segments maintain multiple pointers since they have two or more closest voronoi sites.

As a result, in each voronoi cell, starting from the cut nodes and following the path along the shortest path tree, we can easily connect cut nodes with the voronoi site. We denote the edge connecting the cut node and the voronoi site as *VC edge*. As depicted by the blue edges in figure 3 (b), the VC edges indeed connect adjacent cut nodes through voronoi sites. They form connected-circles containing the network boundaries, which we call VC circles. In the example of figure 3 (b), there are two such circles. One contains the inner boundary around the void area inside, and the other contains the outer boundary of the network.

3.4 Finding Coarse Boundaries

Adjacent cut nodes are two cut nodes which are next to each other on the same boundary. Normally, each voronoi site connects two adjacent cut nodes so that adjacent cut nodes on the network boundaries will also be adjacent on the circles formed by VC edges. Therefore, by flooding along the VC circles each cut node can find its neighbor cut nodes. Nevertheless, it is more complicated when a voronoi site connects cut nodes on different boundaries as depicted as site O in figure 3 (b). In this case, VC circles of different boundaries intersect at one voronoi site so that searching on one VC circle may reach cut nodes on another VC circle. To address this problem in VVD we introduce a tricky method. We first color a restricted area in each voronoi cell corresponding to one VC circle, and then search neighboring cut nodes on the same boundary within the restricted area.

First, the node on the voronoi segment which is closest to the cut node but not on the VC edges locally floods a *Virtual Hole Message* (*VHMsg*) within several hops, usually a value less than 10 is sufficient, to generate a virtual hole. Nodes on VC edges discard this message such that the virtual hole does not affect the nodes within S. The virtual hole prevents messages sent by cut nodes from going out of S. The virtual hole and the VC circles compose a virtual boundary of the restricted area S. The objective of constructing the restricted area is to distinguish VC circles of different network boundaries and restrict the further operations of our algorithm in a local area.

Within the restricted area S, each cut node floods an *Adjacent Cut Node Notification Message* (*ACNMsg*) with its id. The restricted area S forces the *ACNMsg* flooding to the correct adjacent cut node no matter how many cut nodes from what different boundaries connect to a single voronoi site. After flooding the *ACNMsg* in S, every cut node eventually knows its adjacent cut nodes.

Till now, each cut node knows its adjacent cut nodes. We then connect adjacent cut nodes with the shortest path between them. Indeed it will be very inefficient to rebuild the shortest path tree to find the shortest path. Hence, we aggregate such a process by inserting a hop counter to the *ACNMsg*. With the hop counter, every cut node can build a shortest path tree rooted at itself while flooding the *ACNMsgs* within S. Thereby, shortest paths between two adjacent cut nodes can be built without any extra traffic overhead.

The shortest paths between adjacent cut nodes concatenate themselves into a circle as the coarse boundary. Two coarse boundaries are formed, denoted in red. They are corresponding to the inner and outer boundaries of the network.

3.5 Refining the Boundaries

At currently stage, the coarse boundaries divide the network and indicate the topology shape of the network. Such boundaries, however, may not be tight enough, especially when the boundary is concave.

To refine the detected boundaries, we map the coarse boundaries to those extremal nodes on the real boundaries, similar with what Wang et al. did [19]. Extremal nodes are those nodes whose hop counts to the coarse boundaries are local maximum. To explore the extremal nodes, we tour along each coarse boundary R, and label the nodes on R in order. As depicted in figure 3 (d), after a local flooding from the nodes on R, each node will have a minimal hop count to the nodes on R and a parent pointer pointing the shortest path to R. The label of node at hop count k can be computed from the average of labels of its neighbor nodes at hop k-1. The extremal nodes are then labeled and pointed along the shortest paths to R. By checking neighbor extremal nodes with adjacent labels, we can connect those extremal nodes to components. Figure 3 (e) depicts the detected extremal nodes and their connected components (colored in sky blue).

We then force the boundary R to go through the connected components of extremal nodes. We denote the nodes on R that have branches to the extremal nodes as *branch nodes*. They can be detected through tracing the shortest paths from the extremal nodes to R. During the tracing process, reverse pointers are recorded to form a path from branch nodes to extremal node. After the branch nodes are detected, we tour R in a decreasing order according to the node labels, and force the boundary to go through the extremal nodes when encountering the branch nodes. Once branching to extremal nodes, we tour along the connected components of extremal nodes as long as possible and then go back to R. Thereby, we refine the coarse boundary R to a tight boundary. The final result is shown in figure 3 (f), where both the inner and outer boundaries of the network are successfully recognized.

4. Proof of Correctness

In this section, we prove the correctness of VVD protocol in continuous case. Though practical sensor networks are in discrete case, the proof gives intuition of the correctness for discrete case. In practical, this approximates a scenario that sensor nodes are deployed where void areas are much larger compared to communication range. The communicational path between two nodes is thus a curve on the solid field connecting two points.

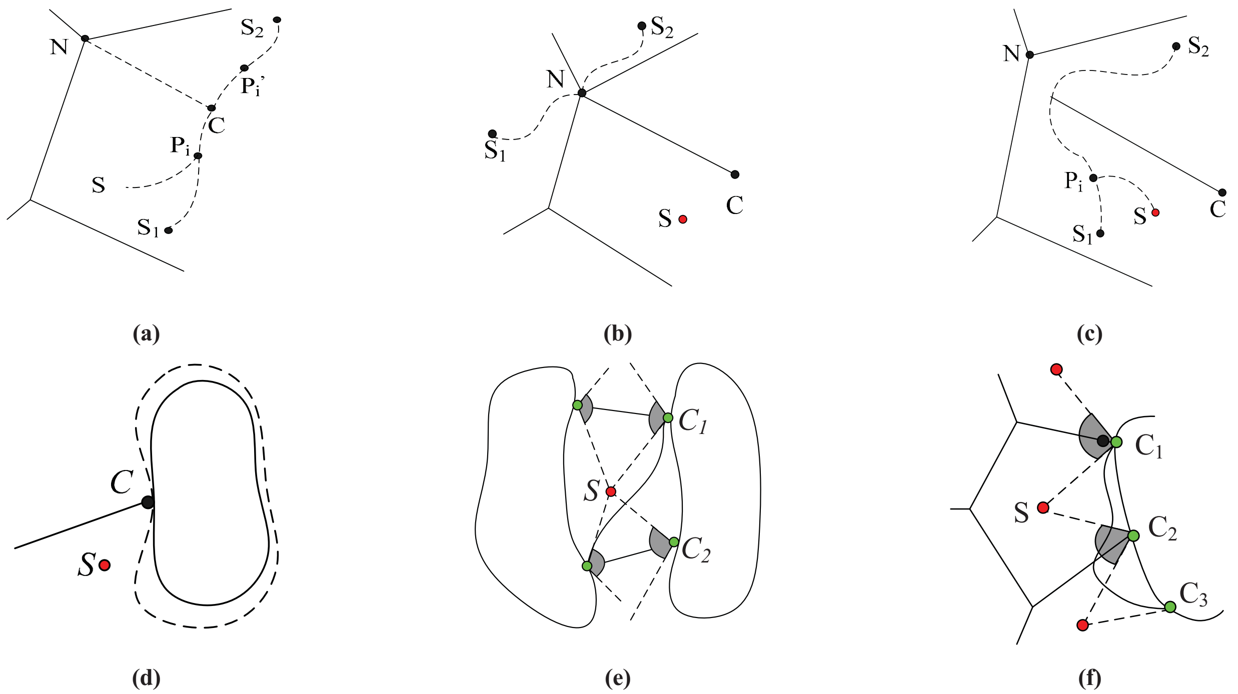
Lemma 1. Let $N_\varepsilon(C)$ be the ε-neighborhood of a cut node C, $N_\varepsilon(C)$ cannot fall into the same voronoi cell.

Proof. We prove it by contradiction. As figure 4 (a) illustrates, assume the two closest voronoi sites to the cut node C are S_1 and S_2. Denote the shortest paths from S_1 and S_2 to C as S_1C and S_2C.

If the ε-neighborhood of C falls into the same voronoi cell, the entire voronoi segment containing C has to be within some voronoi cell VC_i of some voronoi site S_i. We show contradictions in the following 3 cases:

Case 1: Both the two shortest paths S_1C and S_2C pass through some points in VC_i, say P_i and P_i'. Then there exists one path $S_iP_i+P_iC$ from S_i to C which is less than $S_1P_i+P_iC$. This is because P_i resides in cell VC_i, such that $S_iP_i< S_1P_i$. Similarly, the shortest path S_iC is less than S_2C. This leads to the conclusion that S_i is a closer voronoi site to C than both S_1 and S_2 which is contradicted to the fact that there are two closest voronoi sites S_1 and S_2.

Case 2: At least one of the shortest path passes through the voronoi segment. Assume S_1C passes through the voronoi segment NC where N is a Voronoi node as depicted in figure 4 (b). The fact that NC is a part of the shortest path S_1C indicates that S_1C passes the voronoi node N. According to the definition of the vo-



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Proof of correctness.

ronoi node, N has at least 3 equidistant voronoi sites. If one point on the shortest path S_1C has 3 equidistant voronoi sites, then the remaining part of the path also has 3 equidistant sites. Therefore, the points on NC which belongs to the shortest path has at least 3 equidistant sites. This leads to a contradiction since cut node C is not a voronoi node and has only 2 equidistant voronoi sites.

Case 3: The last case is that the voronoi segment NC does not connect to a voronoi node. As depicted in figure 4 (c), both the shortest paths S_1C and S_2C in such a case certainly pass some points in some voronoi cell VC_i in order to connect to point C. Similar to case 1, in this case, the voronoi site S_i will be the only closest sites to C. This contradicts the fact that there are two closest sites S_1 and S_2. □

Lemma 2. All cut nodes reside on the network boundaries.

Proof. If a cut node is not on the network boundaries, it is easy to see that the ε-neighborhood of this node will be in the same voronoi cell. This contradicts with Lemma 1. □

We denote the voronoi segments enclosing the voronoi cell of voronoi site S as the *corresponding voronoi segments* of S and the cut nodes on the corresponding voronoi segments of S as the *corresponding cut nodes* of S. We have the following lemma.

Lemma 3. A voronoi site cannot have exactly one corresponding cut node on one boundary.

Proof. Assume a voronoi site S has exactly one cut node C on the boundary as depicted in figure 4 (d). The ε-neighborhood of cutnode C should be in the same voronoi cell since any two points in the ε-neighborhood of C can be connected without crossing any other voronoi segments. This contradicts with Lemma 1. □

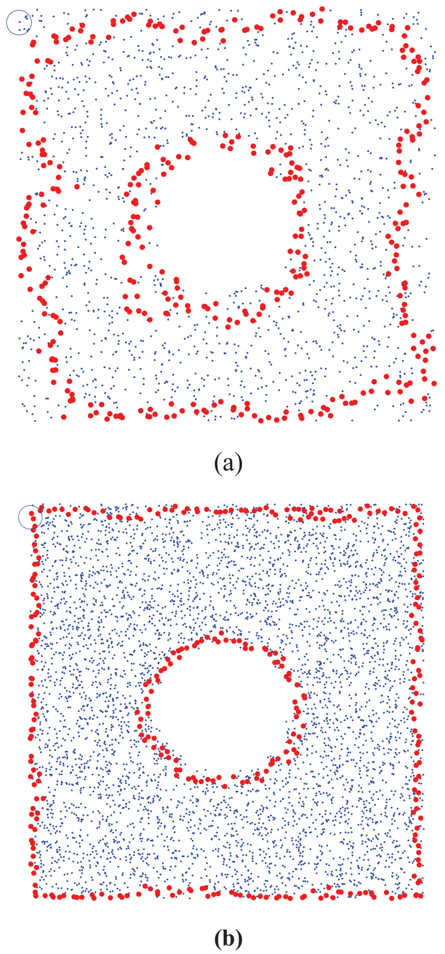
Recall that the protocol executes in the following 4 steps.

1. VVD selects points as the voronoi sites and accordingly builds the voronoi diagram on the plain.
2. VVD detects all cut nodes and builds VC edges that connect cut nodes with the voronoi sites of the voronoi cells they reside in.
3. By concatenating the adjacent cut nodes in order, VVD finds the coarse boundaries in the network. They are connected by the shortest path betweens them through flooding in the restricted areas.
4. VVD refines the coarse boundaries with the help of the extremal nodes.

According to Lemma 2, all the cut nodes are on the boundaries. As two cut nodes are adjacent if and only if they are adjacent on the same boundary, we have the following lemma.

Lemma 4. Only adjacent cut nodes on the same boundary will be connected with the shortest path between them at step 3.

Proof. We show this lemma by contradiction. According to Lemma 2, all cut nodes reside on the network boundaries. The virtual hole and VC edge form a circle containing the entire hole inside. Therefore, as depicted in figure 4 (e), if two cut nodes C_1 and C_2 on different boundaries can be connected by such a flooding, then the path connecting the two cut nodes must cross the VC edge or virtual hole. This contradicts with our requirement that the flooding is constrained inside the restricted area. Therefore, only cut nodes on the same boundary are connected through such a restricted flooding. Similarly, as shown in figure 4 (f), connecting two cut nodes on the same boundary which are not adjacent either crosses the VC edges or



(a)

(b)

Figure 5: Results of VVD for networks of different node densities. (a) Node degree = 7 (b) Node degree = 20.

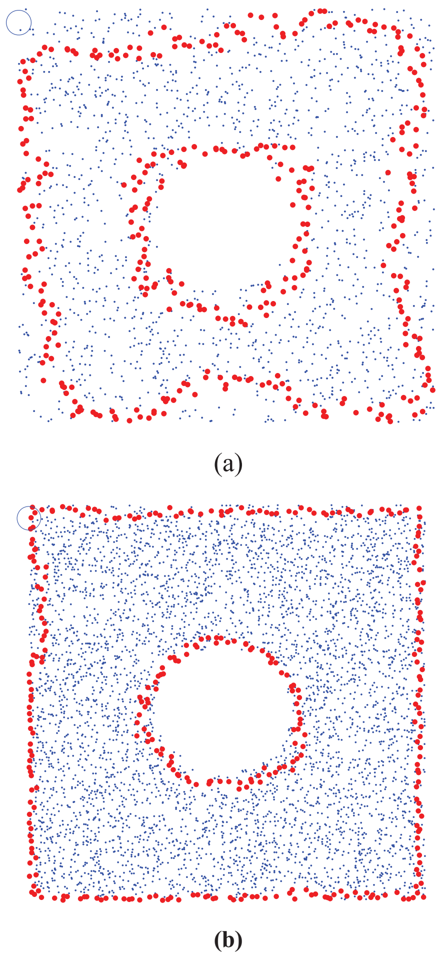
the virtual hole. Thus flooding in the restricted area only connects adjacent cut nodes. □

Lemma 5. Tight boundaries for voids of convex shapes can be obtained after step 3.

Proof. According to Lemma 2, all cut nodes are on the network boundaries Lemma 4 shows that only adjacent cut nodes will be connected by the shortest paths we explore. Obviously, the shortest path between any pair of two points on the boundary of a convex void area goes along the boundary of the void area. Thus the boundary connected by the shortest paths between adjacent cut nodes is tight for voids of convex shape. □

Theorem 1. After refining the coarse boundaries by forcing the boundaries go through the extremal nodes, we obtain tight boundaries for concave void areas.

Proof. According to Lemma 3, we obtain the coarse boundary by concatenating the adjacent cut nodes. According to the proof in [19], the extremal nodes of the coarse boundary are on the real boundary, and tight boundaries can be obtained by forcing the coarse boundary to go through the extremal nodes. □



(a)

(b)

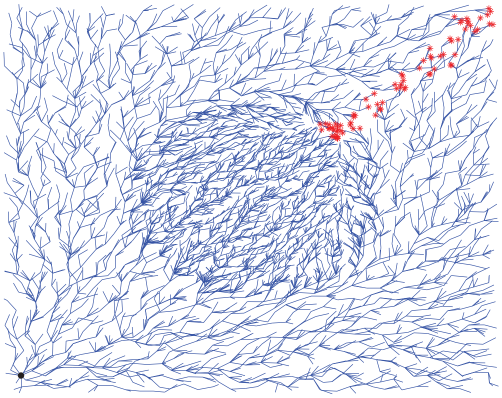
Figure 6: Results of the approach of Wang et al. for networks of different node densities. (a) Node degree = 7; (b) Node degree = 20.

Theorem 1 shows that on the continuous plain our VVD approach successfully recognizes the boundaries for the solid area of both convex and concave shaped boundaries.

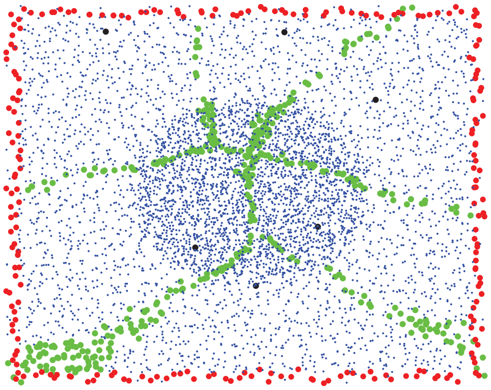
5. Performance Evaluation

We conduct intensive simulations to validate the design of VVD and compare it with the most recent topological approaches. In our simulations, we assume sensor nodes are uniformly randomly deployed in the field. Each node is firstly modeled with a basic communicational range. We then add random perturbations on the communications of each node to make the communication more realistic. When testing the network irregularities, we further divide sensor nodes into different groups, each of which has a different basic communicational range.

The simulation results show the effectiveness and efficiency of our approach. Our result shows that VVD has outstanding performance even with sparse and irregular networks.



(a)



(b)

Figure 7: a sensor network where nodes in the center of the field have smaller communication ranges.

5.1 Detection Accuracy with Node Density

In this simulation, sensor nodes are uniformly randomly deployed in a field of fixed size with a circular void at the center.

We vary the number of sensors deployed inside the field and test the performance of VVD under different node densities. Figure 5 depicts the detected network boundaries by VVD when the average node degree is varied from 7 to 20. On the upper-left corner, we depict the basic node communicational range by a circle as reference. As figure 5 depicts, the detection accuracy of VVD is increased with the node density. We can observe from figure 5 (a) that even the average node degree is about 7 when there are less than 2000 nodes, VVD can achieve a good result.

Certainly, node density is a critical factor that affects the performance of most boundary detection approaches. As indicated by Wang et al. [19], other approaches such as [12] and [9] have extreme difficulties in dealing with the sparse networks. They generate many misreported boundary nodes when the node density is below 16. The work of Wang et al. [19] has been shown to be the most accurate topological method when dealing with sparse networks. In comparison, we simulate their approach and output the results in figure 6. From the results, we can find that both their approach and our VVD achieve similar accuracy with different node densities. Nevertheless, our further results show that VVD outperforms their approach when the network communications become irregular.

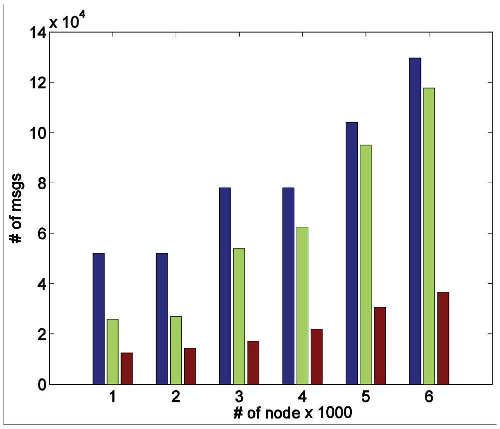


Figure 8: The traffic overhead comparison of VVD and other works.

5.2 Network Irregularities

In this simulation, we test the performance of VVD under network irregularities. Network irregularities can be the result of many reasons, e.g., heterogeneous communication ranges, signal irregularities, environment interferences, and etc. We model the network irregularities by varying the node communication ranges. We divide the sensor nodes into different groups and assign them different basic communicational ranges. We compare VVD with the work of Wang et al. [19] under this scenario.

We examine a sensor network where sensor nodes in the center of the field have smaller communication ranges. As depicted in figure 7 (a), the shortest path tree created in the work of Wang et al. bypasses the central area, meeting at the other end. The misreported cut pairs announce a non-existing void area. This is because the shortest path tree created in their approach is prone to explore the long links around the central area that lead to smaller hop distances. Oppositely, as depicted in figure 7 (b), with only 5 voronoi sites, VVD successfully detects the outer boundary of the network without any misreports.

Generally, the network irregularities introduce inevitable errors in approximating the real distances with hop count distance. The work of Wang et al. aggressively utilizes the distance measures on network topology, and thus leads to failures when large errors are introduced due to network irregularities. The VVD approach, on the other hand, prevents to directly utilize the concrete value of network distance, and is thus immune to such irregularities.

5.3 Traffic Overhead

Finally, we conduct simulations to examine the traffic overhead in recognizing the boundaries. Sensor network is widely known as resource constraint and the communication module plays a major role in energy consumption. A reasonable way to evaluate the network efficiency is to compute the number of messages transmitted.

In this simulation, we compare the number of messages transmitted in VVD with those in Wang et al. and Funke's approaches [9, 19]. We vary the number of sensors deployed in a 60 × 60 area. Results in figure 8 show that VVD is much more efficient with less than one third of their total message amount. Such effect is much more apparent when the network size is large.

The reason is that most of the operations in VVD are bounded within local areas. Due to the multiple floodings, most of the time, the cost of Funke's approach has the highest cost. The number of messages in Wang's approach increases rapidly when the network becomes large, because their approach needs to check more leaf nodes on the shortest path tree to verify whether they are nodes on cut.

6 Conclusion

Topological boundary recognition enables us to obtain meaningful boundaries of sensor networks with minimum information under practical network settings. In this paper, we proposed VVD approach to detect network boundaries in WSNs solely with node connectivity information. We do not assume any distance or position measures, yet we do not as sume UDG or quasi UDG models for node communication styles, making our method more applicable in a broad scope of WSN applications. Compared with existing topological approaches, VVD is more effective in dealing with network irregularities and more efficient on reducing traffic overhead. Simulation results show that, compared with existing approaches, VVD achieves outstanding performance with sparse networks, irregular links and those networks of multiple void areas.

Acknowledgements

This work is supported in part by NSFC/RGC Joint Research Scheme N_HKUST602/08, National Basic Research Program of China (973 Program) under Grant No. 2011CB302705, China NSFC Grants 60933011, the National Science and Technology Major Project of China under Grant No. 2009ZX03006-001-01, and the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2009A080207002.

References

[1] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in Proceedings of ACM MobiHoc, 2006.

[2] L. Barrire, P. Fraigniaud, and L. Narayanan, "Robust Position-based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges," in Proceedings of ACM DIAL M, 2001.

[3] J. Bruck, J. Gao, and A. A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Network," in Proceedings of ACM MobiCom, 2005.

[4] S. Dulman, A. Baggio, P. Havinga, and K. Langendoen, "A Geometric Perspective on Localization," in Proceedings of ACM MELT, 2008.

[5] Q. Fang, J. Gao, and L. J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," in Proceedings of IEEE INFOCOM, 2004.

[6] Q. Fang, J. Liu, L. Guibas, and F. Zhao, "RoamHBA: Maintaining Group Connectivity in Sensor Networks," in Proceedings of IEEE/ACM IPSN, 2004.

[7] S. P. Fekete, A. Kroller, D. Pfister, S. Fischer, and C. Buschmann, "Neighbor-based Topology Recognition in Sensor Networks," in Proceedings of ALGOSENSORS, 2004.

[8] C. Frank and K. Romer, "Algorithms for Generic Role Assignment in Wireless Sensor Networks," in Proceedings of ACM SenSys, 2005.

[9] S. Funke, "Topological Hole Detection in Wireless Sensor Networks and its Applications," in Proceedings of Joint Workshop on Foundations of Mobile Computing, 2005.

[10] S. Funke and C. Klein, "Hole detection or: "how much geometry hides in connectivity?"" in Proceedings of ACM SCG, 2006.

[11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks," in Proceedings of ACM MobiCom, 2003.

[12] A. Kroller, S. P. Fekete, D. Pfisterer, and S. Fischer, "Deterministic Boundary Recognition and Topology Extraction for Large Sensor Networks," in Proceedings of ACM-SIAM SODA, 2006.

[13] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," IEEE/ACM Transactions on Networking, vol. 18, pp. 320-332, 2010.

[14] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, Localization, and Localizability," Journal of Computer Science and Technology, pp. 274-297, 2010.

[15] O. Saukh, R. Sauter, M. Gauger, P. J. Marron, and K. Rothermel, "On Boundary Recognition without Location Information in Wireless Sensor Networks," in Proceedings of IEEE/ACM IPSN, 2008.

[16] S. Schmid and R. Wattenhofer, "Algorithmic Models for Sensor Networks," in Proceedings of IEEE IPDPS, 2006.

[17] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental Analysis of Concurrent Packet Transmission in Low-Power Wireless Networks," in Proceedings of ACM SenSys, 2006.

[18] I. Stojmenovic, A. Nayak, and J. Kuruvila, "Design Guidelines for Routing Protocols in Ad Hoc and Sensor Networks with a Realistic Physical Layer," IEEE Communications Magazine, vol. 43, pp. 101 - 106, 2005.

[19] Y. Wang, J. Gao, and J. S. B. Mitchell, "Boundary Recognition in Sensor Networks by Topological Methods," in Proceedings of ACM MobiCom, 2006.

[20] Z. Yang and Y. Liu, "Quality of Trilateration: Confidence based Iterative Localization," IEEE Transactions on Parallel and Distributed Systems, vol. 21, pp. 631-640, 2010.