Detecting Coverage Boundary Nodes in Wireless Sensor Networks

Chi Zhang, Yanchao Zhang, *Student Member, IEEE*, and Yuguang Fang, *Senior Member, IEEE*

Abstract— **The objective of this paper is to provide a distributed protocol that allows individual sensor nodes to identify themselves as being located on the coverage boundary, which is required in a number of functionalities at both the network and application levels. We develop a deterministic method for boundary node detection based on localized Voronoi polygons, the technique originated from the computational geometry. The advantages of our method are: it is a deterministic one that can be applied to any arbitrarily deployed sensor network, it is truly localized, only need one-hop neighbors' information, which guarantees the scalability and energy efficiency of the detection algorithms and it requires only a limited number of simple local computations. We also provide mathematical as well as experimental evidence for the correctness and efficiency of this method.**

I. Introduction

A wireless sensor network (WSN) is a collection of a large number of small devices each with sensing, computation and wireless communication capability. These tiny sensor nodes are deployed in the target field and collaborate to form an ad-hoc network capable of reporting the phenomenon to a data collection point called sink or base station. A node in a WSN is thus performing two demanding tasks simultaneously: (1) sensing of the phenomenon and (2) communicating with neighbors for relaying the sensed data to the base station. Therefore, coverage and connectivity are two most important aspects of WSNs [8].

If any of initial deployment errors, sensor failures, or change in sensor positions cause the coverage hole of the target area or disconnection of network, detection the boundary of coverage or equivalently, identifying the boundary nodes, is required for answering the fundamental question such as how well does WSN observe the physical space or to support a number of functionalities at both the network and application levels, such as routing [1], topology control [3], self-monitoring and network repairing [7]. However, in the context of WSNs, to solve this problem becomes quite challenging for the absence of a central control unit, and the limited capacities and available information of sensor nodes.

In this paper, we study the fundamental problem of WSNs: how can we use a limited amount of strictly local information to gain distributed knowledge of global network properties such as coverage or connectivity? We develop a completely local rule, based on localized Voronoi polygons, the technique originated from the computational geometry, for each sensor node in a WSN to test if it is on the coverage boundary. Both our analysis and simulations show that this method can correctly identify coverage boundary by only using one-hop neighbors' information which can achieve energy efficiency, without introducing more communication burden and has a quick response to the change of the coverage boundary.

The remainder of the paper is organized as follows. In Section II, network model, problem definition and related work are discussed. In Section III, we develop a localized-Voronoi-polygon based approach for coverage boundary detection and prove the correctness of the algorithm in Section IV. Simulation results and performance analysis are given in Section V, and this paper is finally concluded in Section VI.

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and the US National Science Foundation under grant ANI-0093241 (CAREER Award).

Chi Zhang, Yanchao Zhang and Yuguang Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (email: zhangchi@, yczhang@, fang@ece.ufl.edu).

II. Network model and Problem Statement

A. Assumption and Network Model

Throughout this paper, we assume that any two sensor nodes can directly exchange messages if their Euclidean distance is not greater than r_c, the *communication range*; and a position in the plane can be perfectly monitored (or covered) by a sensor node if their Euclidean distance is not greater than r_s, the *sensing range*. We also assume that sensor nodes are homogeneous, i.e., r_c and r_s are the same for all nodes, and keep constant during sensor's lifetime, since tuning r_c and r_s individually is almost impossible for large-scale WSNs.

For $l > 0$, define A_l as the square region to be monitored (called the *region of interest* or ROI) of side l centered at the origin, i.e., $A_l = [-l/2, l/2]^2$, and ∂A_l the *border* of ROI. We denote sensor nodes which are deployed in the plane and contribute to the monitoring of ROI to be $V = \{s_1, s_2, \cdots, s_n\}$ $(s_i, s_j \in A_l \oplus Disk_0, s_i \neq s_j$, for $1 \leq i \neq j \leq n, i, j \in N)$, where s_i represents the position of node i. Note that for Minkowski-addition, we have $A \oplus B = \{u + v : u \in A, v \in B\}$ for $A, B \subset \mathbb{R}^2$.

Conventionally, the WSN described above can be modelled as a simple loopless undirected graph $G = G(V, E)$, where V is a set of vertices and E is a set of edges, a specified collection of unordered pairs from V, i.e., $E \subset V \times V$.

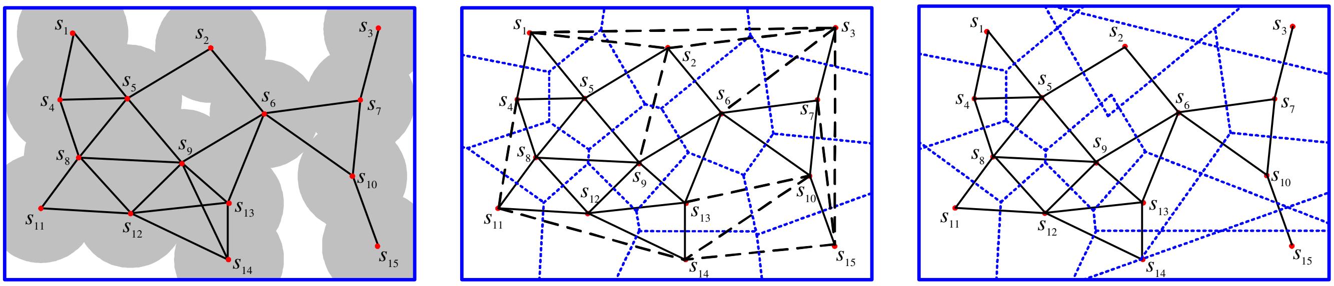
Definition 1: **(neighbor)** We say that nodes s_i and s_j ($i \neq j$ and $s_i, s_j \in V$) are *neighbors* or there exists a *direct wireless links* between them if and only if $\|s_i - s_j\| \leq r_c$.

Thus the edge set E of G, which includes all possible communication links, can be defined as

$$E = \{\overline{s_i s_j} : s_i, s_j \in V \wedge s_j \in Neig\,(s_i)\}\,. \qquad (1)$$

where $Neig\,(s_i)$ the neighbors of node s_i (not including s_i).

Using the graph model, it is convenient to describe the connectivity properties of WSNs:



(a) An example of wireless sensor networks. Solid lines represent all the possible communication links and shaded area represent sensing coverage.

(b) Voronoi diagram of (a). Dotted lines represent Voronoi polygons, solid lines link Voronoi neighbors with distance $\leq r_c$ and dashed lines link Voronoi neighbors with distance $> r_c$.

(c) Localized Voronoi diagram of (a). Dotted lines represent localized Voronoi polygons, and solid lines link local Voronoi neighbors.

Fig. 1. Illustration of Voronoi diagram and localized Voronoi diagram.

Definition 2: **(connectedness)** We say two nodes s_i and s_j are *connected* if there exists at least one sequence $s_1, s_2, \cdots, s_m$ such that $\overline{s_i s_{i+1}} \in E$ for $i = 1, 2, \cdots, m-1$.

Definition 3: **(cluster)** A connected set of nodes $Clust\,(s_i) = s_1, s_2, \cdots s_m$ $(i = 1, 2, \cdots m)$ is said to be a *maximally connected set*, or a *cluster* if the addition of any other node s_{m+1} in the WSN to $Clust\,(s_i)$ breaks the connectedness property. We write $Clust\,(s_i)$ for the cluster containing the node s_i, and call it as the *cluster with* s_i.

In order to characterize coverage properties of WSN, we still need to introduce Boolean model from coverage processes and stochastic geometry [6]. Based on the sensing model, the sensing disk of sensor node s_i can be given by

$$Disk_i = Disk(s_i, r_s) = \{u : |u - s_i| \leq r_s \wedge u \in \mathbb{R}^2\}. \quad (2)$$

Specifically, let $\mathbf{0}$ indicate the origin and we have $Disk_0 = Disk(\mathbf{0}, \mathbf{r_s})$.

In most cases, we are only interested in the *connected coverage* or *cluster coverage*, which is defined as follows:

Definition 4: **(coverage of a cluster)** Given $Clust\,(s_i)$, we refer to the set of all points in the monitored field that are within radius r_s from any node of $Clust\,(s_i)$ as the set *covered* by cluster $Clust\,(s_i)$. Denoting this set by $Cover\,(s_i)$, we have

$$Cover\,(s_i) = \left(\bigcup_{u \in Clust(s_i)} (u + Disk_0) \right) \bigcap A_l. \quad (3)$$

We claim A_l as being *completely covered* if there is at least one cluster $Clust(s_i)$ whose nodes can cover every point in A_l, namely, $Cover(s_i) = A_l$.

For convenience only, we set $r_c = 2r_s$ throughout the rest of the paper. There are two reasons for doing so. First, it is proved in [9] that, for arbitrary spatial distributions of sensor nodes, $r_c \geq 2r_s$ is enough to ensure the network connectivity as long as region A^l is completely covered. Therefore, we set $r_c = 2r_s$ to reduce communication energy consumption and interference. Second, as pointed out in [10], the specification of $r_c = 2r_s$ holds for most commercially available sensors such as Berkeley Motes and Pyroelectric infrared sensors. However, it should be noted that our algorithms are still applicable to the scenarios of $r_c > 2r_s$.

B. Problem Definition

For a cluster $Clust(s_i)$, there are two types of boundaries: the *outside boundary* that is the unique boundary curve separating $Cover(s_i)$ from the outside, and inside boundaries that separate $Cover(s_i)$ from coverage holes.

In order to treat it in a distributed way, the coverage boundary problem should be transferred into the identification of all boundary nodes. Given s_i, the cluster with s_i, i.e., $Clust\,(s_i)$, is unique. Therefore, the *boundary nodes* and *interior nodes* of $Clust\,(s_i)$ can be briefly denoted as $BN\,(s_i)$ and $IN\,(s_i)$ respectively:

$$BN\,(s_i) = \{u \in Clust\,(s_i) : \min \|u - v\| = r_s \text{ for } \; v \in \partial Cover\,(s_i)\}\,;. \quad (4)$$

$$IN\,(s_i) = \{u \in Clust\,(s_i) : u \notin BN\,(s_i)\}\,. \quad (5)$$

the problem we address in this paper can be formally described as the following:

1) *Objective*: for a node $s_i \in V$, find a algorithm for s_i to automatically decide whether $s_i \in IN\,(s_i)$.

This task will be much simpler if we do it in a centralized way and the exact location of each node is available. For example, a single node has access to locations of all sensors (an "image" of the sensor distribution). In this scenario, traditional ways of edge detection in image processing are applicable. However, due to the energy constraints, this scenario is impractical for most WSNs. In this paper, we design our algorithm with the following requirement in mind:

2) *Algorithm Requirement*: the algorithm must be localized and when detecting whether it itself is on the boundary, every node can only use one-hop information (the information of their immediate neighbors), which could be the orientation and the distance to each neighbor.

Such a localized algorithm is highly desirable for a few reasons. First, it helps minimize related communications to achieve energy conservation. Second, it can bound the time for coverage boundary detection and quickly respond to changes of network conditions. Last, requiring only one-hop information can greatly reduce the vulnerability of the algorithm to security attacks because compromised nodes may supply false information.

C. Related Work

The work presented in this paper most closely resembles the work done in [1], [2], [7], where the authors use Voronoi and Delaunay diagrams to determine the coverage boundary nodes. Briefly speaking, the *Voronoi diagram* of a set of nodes V in the space, $\mathfrak{Vor}\,(V)$, is the partition of the space into polygons, i.e., *Voronoi polygons* (VP) $Vor\,(s_i)$, $s_i \in V$ such that all the points inside $Vor\,(s_i)$ are closer to s_i than to any other node in V. The *Delaunay triangulation*, $\mathfrak{Del}\,(V)$, is the dual graph of $\mathfrak{Vor}\,(V)$: an edge $\overline{s_i s_j}$ is in $\mathfrak{Del}\,(V)$ if and only if $Vor\,(s_i)$ and $Vor\,(s_j)$ share a common boundary, and s_j is called the Voronoi neighbor of s_i(see Fig. 1(b) for an example). According to the closeness property of Voronoi polygons, if some portion of the polygon is not covered by the sensor lying inside the polygon, it will not be covered by any other sensor, thus contributing to coverage holes. Therefore the authors in [1], [2], [7] claim that given the Voronoi polygon, each node can locally check whether itself is on the coverage boundary. However, the problem is that Voronoi polygon cannot be derived locally. Fig. 2 gives an example. In order to calculate $Vor\,(s_i)$ of node s_i, we at least need to know all the Voronoi neighbors of s_i. In Fig. 2, node s_l is a Voronoi neighbor of s_i, however, it is 10 hops away from s_i. From the computational geometry, we know that, in general, the Voronoi polygons of boundary nodes cannot be locally computed [4]. One main contribution of this paper is that we develop a truly localized Voronoi polygon method, and provide a new boundary detection algorithm which only need one-hop neighbors' information.

III. Localized Voronoi Polygons and the Boundary Node Detecting Algorithm

In this section, we first introduce the concept of *localized Voronoi polygon* (LVP). Then, we describe the algorithm for identifying boundary nodes based on LVP.

A. Definition and Properties of LVP

***Definition** 5:* **(Localized Voronoi Polygon)** For $s_i \in V$, we call the region given by

$$LVor\,(s_i) = \{v : \|v - s_i\| \leq \|v - s_j\| \text{ for } s_j \in Neig\,(s_i)\,, \text{ and } v \in A_l\} \quad (6)$$

the LVP associated with node s_i, and the set given by

$$\mathfrak{LVor}\,(V) = \{LVor\,(s_1)\,, \cdots, LVor\,(s_n)\} \quad (7)$$

the *localized Voronoi diagram* (LVD) generated by the node set V. An example of LVD is shown in Fig. 1(c).

In order to facilitate the discussion of the properties of localized Voronoi polygons we should alternatively define Voronoi polygon and its localized counterpoint in terms of half planes. For two distinct points $s_i, s_j \in V$, the *dominance region* of s_i over s_j is defined as the subset of the plane which is at least as close to s_i as to s_j and is denoted by

$$Dom\,(s_i, s_j) = \left\{v \in \mathbb{R}^2 : \|v - s_i\| \leq \|v - s_j\|\right\}. \quad (8)$$

Obviously the set $Dom\,(s_i, s_j)$ is a half plane bounded by the perpendicular bisector of s_i and s_j, which separates all

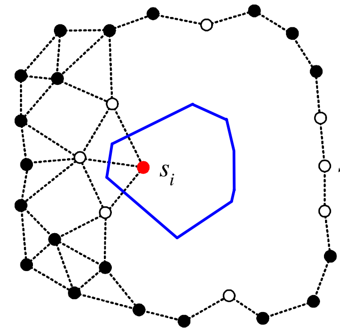


Fig. 2. Example for the scenario that Voronoi polygon cannot be computed locally. Dotted lines describe all the possible communication links. Solid lines represent the Voronoi polygon and open nodes are Voronoi neighbors of node s_i.

points in the plane closer to s_i than those closer to s_j. Thus the Voronoi polygon associated with s_i is the subset of the plane that lies in all the dominances of s_i over the remaining points in V, namely,

$$Vor\,(s_i) = \bigcap\nolimits_{s_j \in V - \{s_i\}} Dom\,(s_i, s_j). \quad (9)$$

Accordingly, the LVP associated with s_i is given by:

$$LVor\,(s_i) = \bigcap\nolimits_{s_j \in Neig\{s_i\}} Dom\,(s_i, s_j). \quad (10)$$

The boundary of $LVor(s_i)$, i.e., $\partial LVor(s_i)$, may consist of line segments, half lines, or infinite lines, which are all called *local Voronoi edges*. Mathematically, the local Voronoi edge generated by s_i and s_j is defined as

$$e(s_i, s_j) = LVor(s_i) \cap Dom(s_j, s_i); \text{if } e(s_i, s_j) \neq \emptyset.$$

Note that $e(s_i, s_j)$ may degenerate to a point. If not, we call s_j a *local Voronoi neighbor* of node s_i, or equivalently, $\overline{s_i s_j} \in \mathfrak{LDel}(V)$. An endpoint of a local Voronoi edge is called a *local Voronoi vertex* which must be an intersection point of two local Voronoi edges.

In practice, we often deal with a bounded region, e.g., A_l, and consider the set given by

$$\mathfrak{LVor}(V) \cap A_l = \{LVor(s_1) \cap A_l, \cdots, LVor(s_n) \cap A_l\}$$

and call this set the *localized Voronoi diagram bounded by* A_l. If $\partial\,(LVor\,(s_i) \cap A_l)$ shares the border of A_l, we call $LVor\,(s_i) \cap A_l$ an *LVP on border*.

***Lemma** 1:* **(Properties of Localized Voronoi Polygons)**

(i) $Vor(s_i) \subseteq LVor(s_i)$ and $Vor(s_i) = LVor(s_i)$ if and only if $\|s_j - s_i\| \leq r_c$ for all $\overline{s_i s_j} \in \mathfrak{Del}(V)$;

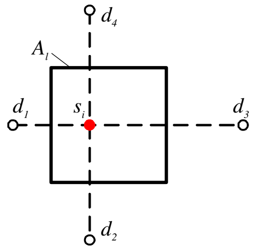
(ii) $\mathfrak{LDel}\,(V) \subseteq \mathfrak{Del}\,(V)$ and $\mathfrak{LDel}\,(V) = \mathfrak{LDel}\,(V)$ if and only if $\|s_j - s_i\| \leq r_c$ for all $\overline{s_i s_j} \in \mathfrak{Del}\,(V)$;

(iii) $LVor\,(s_i)$ is a convex set.

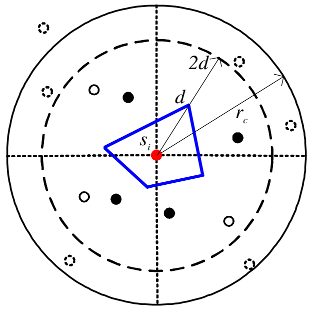
Proof: From (9) and (10), we have

$$Vor\,(s_i) = LVor\,(s_i) \bigcap \left(\bigcap\nolimits_{s_j \in V, s_j \notin Neig\{s_j\}} Dom\,(s_i, s_j) \right),$$

which directly leads to Lemma 1 (i) and (ii). For Lemma 1 (iii), since a half plane is a convex set and the intersection



(a) Inserting dummy nodes (open node) to the neighbor list of node s_i.



(b) Construction of $LVor\,(s_i)$ or $LVor\,(s_i) \cap A_l$.

Fig. 3. Calculating $LVor\,(s_i)$ or $LVor\,(s_i) \cap A_l$.

of convex sets is a convex set[1], an LVP as well as a VP is a convex set. ■

Lemma 2: **(Properties of localized Voronoi diagram)** Localized Voronoi diagram, e.g., $\mathfrak{LVor}\,(V)$, always provides a complete coverage of the whole plane $\mathbb{R}^2$:

$$\bigcup_{s_i \in V} LVor\,(s_i) = \mathbb{R}^2. \quad (11)$$

Proof: It is well known in computational geometry that 2-dimensional Voronoi diagram is a tessellation for the plane $\mathbb{R}^2$, so we have

$$\bigcup_{s_i \in V} Vor\,(s_i) = \mathbb{R}^2. \quad (12)$$

(see [4, Property V1, pp. 77] for a reference). Combined (12) with Lemma 1 (i) that $Vor\,(s_i) \subseteq LVor\,(s_i)$, we can directly obtain (11). ■

Therefore, the set $\mathfrak{LVor}\,(V) \cap A$ can fully cover arbitrary set A when $A \subseteq \mathbb{R}^2$. Note that this result can be easily extended to any cluster in V, e.g., for $Clust\,(s_i)$ we have

$$\bigcup_{s_j \in Clust(s_i)} LVor\,(s_j) = \mathbb{R}^2. \quad (13)$$

B. LVP-Based Boundary Node Detecting Algorithm

In this subsection, we present an algorithm for each node to detect whether it is on the coverage boundary based on its own LVP, which is illustrated with node s_i as an example.

Input: relative positions of node s_i's neighbors and/or ∂A_l (the border of A_l).

Note that we need both directional and distance information of all the neighbors of s_i. Without loss of generality, we assume that s_i has k_i neighbors, denoted by $s_j \in Neig(s_i)$ for $j = 1, \cdots, k_i$.

Step 1: calculating the LVP for node s_i.

Case (i): when only relative positions of neighbors are available. We need to calculate $LVor\,(s_i)$. In this case, some $LVor\,(s_i)$ will be infinite.

Case (ii): when relative positions of neighbors and border of ROI (A_l) are available. We need to calculate $LVor\,(s_i) \cap A_l$. It can be proved that $LVor\,(s_i) \cap A_l$ must be a finite convex polygon. Case (ii) can be transformed into case (i) by introducing dummy nodes in node s_i's neighbor list. See Fig. 3(a) for an example: four dummy nodes d_1 through d_4 are introduced such that perpendicular bisectors between s_i and the dummy nodes generate the four edges of the border of ROI. Therefore, we can calculate $LVor\,(s_i) \cap A_l$ by following the same procedure for case (i).

Equation (10) give a naive method to construct $LVor\,(s_i)$. When the WSN is densely deployed (k_i is large), we can use the property of LVP to optimize naive method:

Algorithm LVOR: **constructing** $LVor\,(s_i)$

input: s_i and its neighbors s_j, $j = 1, \cdots, k_i$
divide $Disk\,(s_i, r_c)$ into four quadrants
search for nearest neighbors in each of the four quadrants: s_1, s_2, s_3, s_4; or give up when no nodes in this quadrant
$LVol\,(s_i) \leftarrow \bigcap_{j=1}^{4} Dom(s_i, s_j)$
$d \leftarrow \max \|s_i - v\|$, v is the vertex of $LVor\,(s_i)$
for $s_j \in Disk\,(s_i, 2d) \cap Disk\,(s_i, r_c) \wedge j \neq 1, 2, 3, 4$ **do**
    $LVol\,(s_i) \leftarrow Dom\,(s_i, s_j) \cap LVol\,(s_i)$
end for
return $LVor\,(s_i)$

As shown in Fig. 3(b) we first construct the tentative localized Voronoi polygon of s_i by the nearest neighbors in each of the four (note that other values will also work) quadrants. In Fig. 3(b), the nearest neighbor are solid nodes. Let d be the distance from s_i to the farthest point of its tentative localized Voronoi polygon, then no node farther than $2d$ from s_i can have any effect on the actual localized Voronoi polygon of s_i, which means that we only need to consider the small (expected constant) number of nodes (open nodes) that are in the dashed circle of Fig. 3(b).

Step 2: checking vertices of $LVor(s_i)$.

For the every vertex v of $LVor\,(s_i)$, check $\|s_i - v\|$, if it is larger than r_s, then v is not covered by s_i. Note that when $LVor\,(s_i)$ is infinite, we do not need this step.

Output: $s_i \in IN\,(s_i)$ or $s_i \in BN\,(s_i)$.

If $LVor\,(s_i)$ is infinite, s_i must be on the boundary.

If $LVor\,(s_i)$ is finite and all the vertices of $LVor\,(s_i)$ are covered by s_i, then $s_i \in IN\,(s_i)$; if at least one vertex of $LVor\,(s_i)$, e.g., v, is uncovered by s_i, then $s_i \in BN\,(s_i)$, and uncovered area is in the direction of $\overline{s_i v}$.

IV. Validation of LVP-Based Boundary Node Detecting Algorithm

First, we show that $LVor\,(s_i)$, as well as $Vor\,(s_i)$, can be used to characterize the coverage properties of a WSN.

Theorem 1: *If there is a point $v \in LVor\,(s_i)$ which is not covered by s_i, i.e., $v \notin Disk\,(s_i, r_s)$, there must exist a point $h \in LVor\,(s_i)$ that is not covered by any node, and s_i must be a boundary node.*

Proof: Without loss of generality, we assume that the node nearest to s_i and outside $Disk\,(s_i, r_s)$ is s_m, and $\|s_i - s_m\| = r_c + \delta$ for $\delta > 0$. Let s'_m be the point on $\overline{s_i v}$ satisfying $\|s_i s'_m\| = \|s_i s_m\|$, and h be another point

[1]This lemma can be briefly proved as follows: Let B_i, $i \in \mathbb{I}$, be a convex set and $B = \bigcap_{i \in \mathbb{I}} B_i$. If u and v are two points in B, then they are in each B_i, so the line joining u to v lies in each B_i and therefore in B.

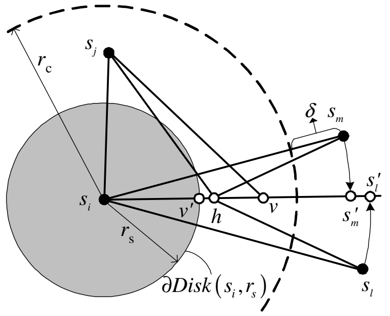


Fig. 4. Illustration of the proof of theorem 1.

on $\overline{s_i v}$ such that $\|s_i h\| = r_s + \delta/2$ (see Fig. 4). By the triangular inequality, we have $\|s_m h\| + \|s_i h\| \geq \|s_i s_m\| = \|s_i s'_m\| = \|s_i h\| + \|h s'_m\|$. Therefore, $\|s_m h\| \geq \|h s'_m\| = \|s_i s'_m\| - \|s_i h\| = r_c + \delta/2$, which means that s_m cannot cover h and neither does any other node in $\overline{Disk\,(s_i, r_s)}$. The reason is that, since $\|s_i s_l\| > \|s_i s_m\|$ holds for any node $s_l \in \overline{Disk\,(s_i, r_s)}$ and $s_l \neq s_m$, we have $\|s'_l h\| > \|s'_m h\|$ where point s'_l is on the line $\overline{s_i v}$ and $\|s_i s'_l\| = \|s_i s_l\|$. Therefore, $\|s_l h\| \geq \|s'_l h\| > \|s'_m h\| = r_s + \delta/2$.

Since $v \in LVor\,(s_i)$, for any node $s_j \in Disk\,(s_i, r_c)$ and $s_i \neq s_j$, we have $\|s_j v\| \geq \|s_i v\|$ which implies that $\angle s_j s_i v \geq \angle s_i s_j v > \angle s_i s_j h$. Therefore, $\|s_j h\| \geq \|s_i h\|$ and no nodes in $Disk\,(s_i, r_c)$ can cover h. Consequently, we can conclude that no node in the plane can cover h because $Disk\,(s_i, r_c) \cup \overline{Disk\,(s_i, r_c)} = \mathbb{R}^2$. Note that from the above proof process, we can see that h can be arbitrary close to v', the intersection of circle $\partial Disk\,(s_i, r_s)$ and $\overline{s_i v}$. Therefore, $v' \in \partial Cover\,(s_i)$, and s_i is a boundary node. ■

***Theorem** 2: If there is a point $v \in A_l$ not covered by any sensor node, for every cluster $Clust(s_i)$ there must exist at least one sensor $s_j \in V$ whose $LVor\,(s_j)$ is not completely covered.*

Proof: According to Lemma 2 or (13), we have

$$\bigcup_{s_j \in Clust(s_i)} (LVor\,(s_j) \cap A_l) = A_l \tag{14}$$

Therefore, for any $v \in A_l$, it must lie in at least one $LVor\,(s_j) \cap A_l$ for $s_j \in Clust(s_i)$. ■

Theorems 1 and 2 prove that the condition that $LVor\,(s_i) \cap A^l$ is completely covered by s_i for all $s_i \in Clust\,(s_j)$ is the sufficient and necessary condition for $Clust\,(s_j)$ to completely cover A_l. The following theorem shows that when $LVor\,(s_i)$ or $LVor\,(s_i) \cap A_l$ is finite, the coverage of vertices of $LVor\,(s_i)$ by node s_i is equivalent to the coverage of the whole $LVor\,(s_i)$ by node s_i, which guarantees the correction of our LVP-based algorithm.

***Theorem** 3: $LVor\,(s_i)$ is completely covered by s_i if and only if $LVor\,(s_i)$ is finite and all the vertices of $LVor\,(s_i)$ are covered by s_i.*

Proof: Let $Ve\,(s_i)$ be the set of vertices of $LVor\,(s_i)$. Obviously, when $LVor\,(s_i)$ is completely covered by s_i, i.e., $LVor\,(s_i) \subset Disk\,(s_i, r_s)$, we have $v \in Disk\,(s_i, r_s)$ for all $v \in Ve\,(s_i)$. Since the coverage area of s_i, i.e., $Disk\,(s_i, r_s)$) is finite, $LVor\,(s_i)$ of course is also finite

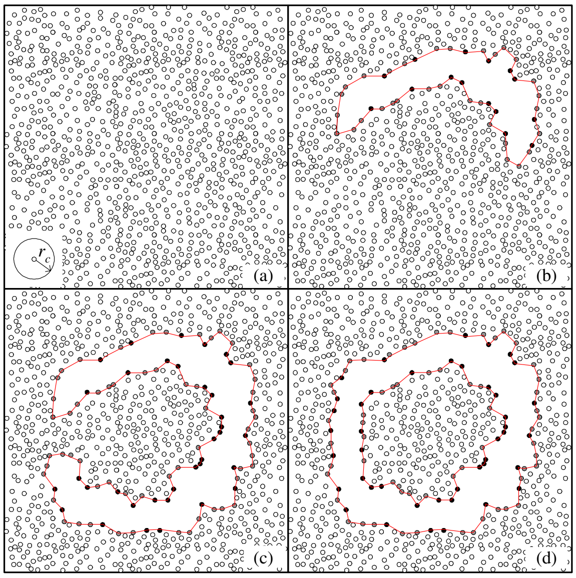


Fig. 5. Snap-shots in the simulation run.

when it is completely covered by s_i. Since $LVor\,(s_i)$ and $Disk\,(s_i, r_s)$ are both convex sets, we obtain

$$\max_{u \in LVor(s_i)} \{\|s_i - u\|\} \leq \max_{v \in Ve(s_i)} \{\|s_i - v\|\}.$$

Therefore, when $v \in Disk\,(s_i, r_s)$ for all $v \in Ve\,(s_i)$, we have $u \in Disk\,(s_i, r_s)$ for all $u \in LVor\,(s_i)$. ■

V. Experimental Evaluation and Performance Analysis

In this section, we will first validate the accuracy of our algorithm by the simulation. Then we will show by the theoretical analysis and experimental results that our algorithm outperforms the VP-based algorithm in energy consumption.

A. Experimental Result

Fig. 5 gives the detection example of a large scale WSN with an intended attacks (physically destruction such as the planned bombing of the WSN). The intention of adversary is very clear, by destroying parts of sensor nodes, disconnecting the WSN, and making a large part of the WSN lose its function. Fig. 5 gives some snap-shots of this process, and the detection results of each state. The ROI is a square with the size of $14r_c \times 14r_c$ and original deployed sensors completely covered the ROI (see Fig. 5(a)). The boundary nodes detected by the NEP-based algorithm and the coverage boundary linked by the theoretical boundary nodes, are shown as shaded dots and solid lines. All the boundary nodes are correctly detected.

B. Cost Analysis

It has been shown that the LVP and the NEP based algorithms can correctly identify the boundary nodes, just like VP-based one. It is the restriction of using only one-hop information that distinguishes our algorithm from VP-based one. Intuitively, this restriction can help to reduce the

TABLE I

PARAMETER SETTINGS FOR SIMULATION

Symbol	Quantity	Values in Simulation
k	number of neighbors	$4.5 < k < 45$
l	side of ROI	$200m$
r_c	communication range	$20m$
S_{pos}	size of the data message for node position information	64 byte
E_T	amount of energy needed to transmit one bit of information	$0.8\mu J/bit$
E_R	amount of energy needed to receive one bit of information	$0.6\mu J/bit$

cost (communication overhead and energy consumption). We make this intuition rigorous as follows:

***Theorem** 4: If there exist boundary nodes, the costs of the NEP-based and the LVP-based algorithms are always smaller than that of the VP-based one.*

The proof of this theorem depends on the following lemma:

***Lemma** 3:* **(locality of computating VPs)** For any $s_i \in V$, Voronoi polygon $Vor\,(s_i)$ can be locally computed (only use one-hop information) if and only if $Clust\,(s_i)$ can completely cover the plane $\mathbb{R}^2$ (or ROI A_l, when the information of the border of A_l is available), i.e., $Cover\,(s_i) = \mathbb{R}^2$ (or $Cover\,(s_i) \cap A_l = A_l$).

Proof: From Theorems 1 and 2, a node set V can completely cover $\mathbb{R}^2$ if and only if $LVor(s_i)$ is fully covered by $Disk(s_i, r_s)$ for any $s_i \in V$. From Lemma 1, we know that this means that $Vor(s_i) = LVor(s_i)$ for any $s_i \in V$. Therefore, $Vor(s_i)$ can be locally computed by s_i just as $LVor(s_i)$. Let $d = \max \|v - s_i\|$ for any $v \in Vor(s_i)$. Since $Vor(s_i)$ is a convex set, then $d = \infty$ if $Vor(s_i)$ is infinite, otherwise d is the distance from a vertex of $Vor(s_i)$ to s_i. Note that $Vor(s_i)$ can also be computed via Algorithm LVOR with set V as input. We can determine that the construction of $Vor(s_i)$ is completed when all the nodes in $Disk(s_i, 2d)$ have been counted (see Fig. 3(b)). Therefore, $Vor(s_i)$ can be locally computed, which implies that $2d \le r_c$ or $d \le r_s$ and thus guarantees the complete coverage of $Vor(s_i)$. Since this holds for all $s_i \in V$, we can ensure the complete coverage of the plane. ■

Therefore, when there are boundary nodes, it is impossible to compute all $Vor(s_i)$'s locally based on only one-hop information. Since multi-hop communications are unavoidable, the cost of the VP-based approach will be higher than LVP-based one. Only when the node density is so high that the ROI is completely covered (not considering the ROI border), is the cost of the VP-based approach equal to that of ours. However, in this case, there is no need for coverage boundary detection at all. So Theorem 4 guarantees that when boundary detection algorithms are helpful, the cost of our algorithms is definitely smaller than the VP-based one. The following question is how significant this improvement?

We assume the sensor nodes are distributed in A_l as a homogeneous Poisson point process with node density λ. Therefor, the expected number of neighbors of each node is $k = \pi {r_c}^2 \lambda$ when $l \to \infty$. Following [5], we take the parameter settings in Table I for the simulation (note that when $k \ge 45$, A_l is completely covered with probability 1). Fig. 6 gives the average node energy consumption for the VP-based (EC_{VP}) and the LVP-based (EC_{LVP}) algorithms as the function of k, which shows that EC_{VP} is almost 3-times of EC_{LVP} for all the values of k (in fact, we can theoretically prove this result, and the detailed analysis is omitted here due to the lack of space). Obviously, the energy saving is significant.

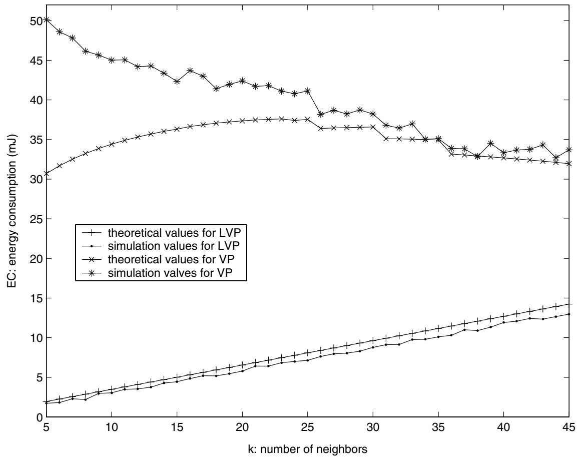


Fig. 6. Energy consumption for the LVP-based (NEP-based) and the VP-based algorithms.

VI. CONCLUSION

In this paper, we develop a deterministic, distributed, localized algorithms for detecting coverage boundary nodes in WSNs. In contrast to previous proposals, our algorithm do not need knowledge about the distribution of sensor nodes, and only depend on one-hop information and a few simple local computations.

REFERENCES

[1] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *INFOCOM'04*, Hong Kong, China, March 2004.

[2] A. Ghosh. Estimating coverage holes and enhancing coverage in mixed sensor networks. In *LCN'04*, Washington, DC, Nov. 2004.

[3] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom'04*, Philadelphia, PA, Oct. 2004.

[4] A. Okabe. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, New York, 2000.

[5] H. Sabbineni and K. Chakrabarty. Location-aided flooding: an energy-efficient data dissemination protocol for wireless sensor networks. *IEEE Transactions on Computers*, 54(1):36–46, January 2005.

[6] D. Stoyan, W. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. Wiley, New York, 2nd edition, 1995.

[7] G. Wang, G. Cao, and T. L. Porta. Movement-assisted sensor deployment. In *INFOCOM'04*, Hong Kong, China, March 2004.

[8] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM SenSys'03*, Los Angeles, CA, Nov. 2003.

[9] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in sensor networks. In *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Fort Lauderdale, FL, Feb. 2004.

[10] H. Zhang and J. Hou. On the critical total power for asymptotic k-connectivity in wireless networks. In *INFOCOM'05*, Miami, FL, March 2005.