Energy-Efficient Coverage Measurement for Wireless Sensor Networks

Chi Zhang, Yanchao Zhang, Yuguang Fang
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611
Tel: (352)392-2746; Fax: (352)392-0044; Email: {zhangchi@, yczhang@, fang@ece.}ufl.edu

Abstract— **In this paper, we propose a coverage measurement protocol (CMP) which can provide the base station an accurate and in-time measurement of the current connected coverage in an energy-efficient way, and our CMP is location-error tolerant. Moreover, the major component of our CMP, i.e., *BOundary Node Detection* (BOND) scheme, can be reused to provide many other functionalities for WSNs.**

I. Introduction

Wireless Sensor Networks (WSNs) have attracted a lot of attention recently due to their broad applications in both military and civilian operations. Usually, a WSN consists of hundreds or even thousands cheap sensor nodes, which deployed over a geographic area (called the *region of interest* or ROI), collaborate with each other and communicate with external world through one or several powerful nodes, i.e., base stations (BSs) [1], [12].

As a consequence of this special network architecture, from the BS's point of view, a position in the ROI is really under the surveillance of the WSN if and only if this position is within the sensing range of at least one sensor node connected to the BS. We define the collection of all these positions in the ROI as the *connected coverage* (or *coverage* in short). Obviously, it is one of the most important performance metric measuring the quality of surveillance a WSN can provide, and the BS also should have the ability to monitor the coverage status in real time.

Although much research [2], [4], [5], [7], [10], [13], [14] has been conducted to ensure high network coverage and connectivity for the WSN, none of them addresses how to help the BS measure the connected coverage by identifying the coverage boundary when coverage holes emerge. Possible causes leading to coverage holes include energy depletion of sensor nodes, intended attacks on sensor nodes, and so on. In many WSN applications, especially security-sensitive applications, it is a must to accurately detect the coverage boundary. The protocol developed in this paper can affirmatively answer this open challenging issue.

On the other hand, problems related to the self-monitoring of a WSN have been studied in the literature for various applications and purposes. For example, Chessa and Santi [18] propose a single time-out scheme to monitor the system-level fault diagnosis. In [25], a residual energy scan is designed to approximately depict the remaining energy distribution within a WSN. In addition, a self-monitoring mechanism for detecting node failures is proposed in [9]. However, All these schemes cannot be directly used for the coverage inference, as they are either centralized schemes or assume that each individual sensor in the WSN needs to be monitored. This is not true for our case because the BS only needs to ensure that a ceratin percentage of the sensors are functioning, especially when the WSN is densely deployed.

This paper makes the following contributions. First, we present a *Coverage Measurement Protocol* (CMP) which can provide the BS an accurate and in-time measurement of the current connected coverage in an energy-efficient way. Second, we show that the major component of our CMP - *BOundary Node Detection* (BOND) scheme - can be reused to provide many other functionalities for WSNs, such as topology control, efficient routing, sleeping scheduling, and spatial aggregation. Therefore, our schemes can be exploited to seamlessly integrate multiple functionalities with low overhead. Moreover, we devise extensions to CMP which can tolerate location errors.

II. Preliminaries

A. Network Model

Throughout this paper, we assume that any two sensor nodes can directly communicate via bi-directional wireless links 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*. Similar to [2], [14], [22], we also assume that sensor nodes are homogeneous in the sense that r_c and r_s are the same for all nodes, and keep constant during each node's lifetime.

Instead of considering all the possible combinations of r_c and r_s, we focus on the case of $r_c = 2r_s$ in this paper. There are two reasons for doing so. First, as pointed out in [24], the specification of $r_c \geq 2r_s$ holds for most commercially available sensors such as Berkeley Motes and Pyroelectric infrared sensors. Second, for arbitrary spatial distributions of sensor nodes, $r_c \geq 2r_s$ is the sufficient and necessary condition for the existence of local boundary node detection algorithms. Therefore, we set $r_c = 2r_s$ to reduce communication energy consumption and interference. However, it should be noted that our algorithms are still applicable to the scenarios of $r_c > 2r_s$ without any changes.

For simplicity, we assume that the ROI is a 2-D square planar field hereafter. Our results, however, can be easily

This work was supported in part by the U.S. National Science Foundation under grant ANI-0093241 (CAREER Award) and grant DBI-0529012.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

extended to 2-D or 3-D ROIs of arbitrary shapes. For $l > 0$, let A_l denote the square ROI of side length l centered at the origin, i.e., $A_l = [-l/2, l/2]^2$, and ∂A_l be the *border* of A_l. We examine a large-scale WSN consisting of hundreds or even thousands of stationary sensor nodes, and denote the sensor nodes deployed in the ROI to be $V = \{s_1, \cdots, s_i, \cdots, s_n\}$ ($s_i \in A_l$, for $1 \leq i \leq n, i \in \mathbb{N}$), where s_i represents the position of node i and n is the total number of sensor nodes (or *network size*).

B. Design Goals

In this paper, we intend to design a coverage measurement protocol which can provide the BS an *accurate* and *timely* measurement of the current connected coverage in an *energy-efficient* way. Specifically, our design goals include:

Effectiveness: The BS should have an accurate measurement of the connected coverage for an arbitrary node distribution even in the presence of location errors and a high packet loss rate due to wireless communication errors.

Low overhead: Minimizing energy consumption to prolong the system lifetime is always a major design objective for all WSN protocols.

Seamless Integration: For resource-limited embedded devices, it is undesirable to design an isolated protocol for a single functionality. The reason is that with the increasing number of functionalities, the protocol stacks of sensor nodes will become too complicated, and the network efficiency will be dramatically decreased [12]. Therefore, when designing a protocol for the WSN, we should opt for the schemes in the design space whose basic operations can be highly reused in providing other functionalities.

III. BOND: Boundary Node Detection Scheme

A connected set of nodes is said to be a **cluster** if the addition of any other node to it will break the connectedness property. Obviously, the problem of finding the boundary of connected coverage, is equivalent to detecting the boundary nodes of clusters with connections to the BS. Based on this observation, it is possible to design a distributed CMP if we can first find a localized way to detect boundary nodes.

This section presents BOND which enables each node to locally self-detect whether it is a boundary node. We begin with several important definitions, followed by the illustration of BOND.

A. Localized Voronoi Polygons

Our BOND scheme is based on two novel geometric concepts called *Localized Voronoi Polygon* (LVP) and *Tentative LVP* (TLVP) which are nontrivial adaptations of *Voronoi Polygons* (VPs) [17] from computational geometry[1].

We first define VPs, LVPs and TLVPs 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 set of points which are at least as close to s_i as to s_j, i.e.,

$$Dom(s_i, s_j) = \left\{ v \in \mathbb{R}^2 : \|v - s_i\| \leq \|v - s_j\| \right\}. \quad (1)$$

Obviously, $Dom(s_i, s_j)$ is a half place bounded by the perpendicular bisector of s_i and s_j, which separates all points in the plane closer to s_i than those closer to s_j.

Definition 1: **[VP, LVP and TLVP]** *The* ***VP*** *associated with s_i denoted by $Vor\,(s_i)$, is the subset of the place that lies in all the dominance regions of s_i over other points in V, namely,*

$$Vor\,(s_i) = \bigcap\nolimits_{s_j \in V - \{s_i\}} Dom\,(s_i, s_j). \quad (2)$$

In the same way, the ***LVP*** *denoted by $LVor(s_i)$, and the* ***TLVP*** *denoted by $TLVor(s_i)$, associated with s_i are defined as:*

$$LVor\,(s_i) = \bigcap\nolimits_{s_j \in Neig(s_i)} Dom\,(s_i, s_j); \quad (3)$$

$$TLVor\,(s_i) = \bigcap\nolimits_{s_j \in SubNeig(s_i)} Dom\,(s_i, s_j), \quad (4)$$

where $SubNeig\,(s_i)$ is a proper subset of $Neig\,(s_i)$, i.e., $SubNeig\,(s_i) \subset Neig\,(s_i)$.

The collection of LVPs given by

$$\mathfrak{LVor}\,(V) = \{LVor\,(s_1), \cdots, LVor\,(s_n)\} \quad (5)$$

is called the *localized Voronoi diagram* (LVD) generated by the node set V. 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*.

B. LVP-Based Boundary Node Detection

In this subsection, we present our BOND scheme for each node to detect whether it is a boundary node based on its own LVP or TLVP by taking node s_i as an example.

1) Input: Our BOND is a distributed scheme in that we only need positions of node s_i's neighbors as the input of our algorithm. We temporally assume that there is no location error and will relax this assumption in Section IV-B. We need to consider two cases based on whether the information about the border of A_l, i.e., ∂A_l, is available. In the first case when ∂A_l is unavailable at node s_i, our detection scheme is based on the construction of $LVor\,(s_i)$ (or $TLVor\,(s_i)$); in the second case when ∂A_l is available, we need to exploit this information by calculating $LVor\,(s_i) \cap A_l$ (or $TLVor\,(s_i) \cap A_l$). It can be shown that $LVor\,(s_i) \cap A_l$ must be a finite convex polygon. Thus, the second case can be transformed into the first case by introducing dummy nodes into $Neig(s_i)$. When ROI is a square field, four dummy nodes, d_1 through d_4, are introduced such that perpendicular bisectors between s_i and the dummy nodes generate the four border edges of ROI. Then we can calculate $LVor\,(s_i) \cap A_l$ by following the same procedure for calculating $LVor\,(s_i)$. Therefore, we will only discuss the first case in what follows.

[1] A similar concept called *Localized Voronoi Diagrams* (LVDs) is also introduced as the dual of *Localized Delaunay Triangulations* (LDTs) in the literature [11], [16]. The edge complexity of LDT is analyzed in [11] and its applications in topology control and routing for wireless networks are discussed in [16]. However, there is no indication on how to relate this concept with the coverage problems in WSNs, and unlike our work no description on how to efficiently construct LVDs is given in [11], [16]. The idea of using TLVP to reduce the overhead of the detection algorithm in this paper is also new.

2) Algorithm: Our goal is to construct the $LVor(s_i)$ (or $TLVor(s_i)$) which is sufficient for the boundary node detection with the minimal requirement on the information about s_i's neighbors. We first divide $Disk\,(s_i, r_c)$ into four[2] quadrants. Then we construct the TLVP of s_i by using the nearest neighbors in each of the four quadrants. Without lose of generality, we denote these four nearest neighbors as s_1, s_2, s_3, and s_4. The first TLVP is calculated by

$$TLVor\,(s_i) \leftarrow \bigcap\nolimits_{j=1}^{4} Dom(s_i, s_j).$$

If all vertices of the TLVP is covered by $Disk\,(s_i, r_s)$, the procedure stops and this TLVP is saved. Otherwise, we need to find new neighbors which are the nearest to the uncovered vertices of the TLVP, add those neighbors to $SubNeig(s_i)$, and calculate the TLVP again:

$$TLVor\,(s_i) \leftarrow TLVor\,(s_i) \bigcap \Big(\bigcap\nolimits_{s_j \in SubNeig(s_i), j \neq 1,2,3,4} Dom\,(s_i, s_j)\Big).$$

The new vertices of the new TLVP will be checked to see whether they are covered by $Disk\,(s_i, r_s)$. This procedure continues until all the vertices of the TLVP are covered by $Disk(s_i, r_s)$ or the LVP of s_i is calculated and saved.

Note that when ∂A_l is unavailable, $LVor(s_i)$ may be infinite, which means that it is possible that we cannot find any nodes in one or more quadrants in the first step. If a quadrant contains no neighbors, we define two sectors of angle 45° which are directly adjacent to the quadrant as the assistant area, and add the nodes in this area to $SubNeig(s_i)$ first. If all the nodes in the assistant area cannot make TLVP finite, we can conclude that LVP must be infinite without need to do further calculation.

3) Output: If $LVor\,(s_i)$ is infinite, s_i must be a boundary node. If $LVor\,(s_i)$ (or the final $TLVor\,(s_i)$) is finite with all the vertices are covered by s_i, then $s_i \in IN\,(s_i)$. Otherwise, $s_i \in BN\,(s_i)$.

C. Discussion on BOUND

Correctness. The correctness of BOUND is proved in our previous work [23] for arbitrary node distributions.

Low Overhead. It has been shown in [23] that in general VPs cannot be computed locally. Therefore, the traditional VP-based schemes [4], [6], [21] are not distributed and are very expensive in terms of communication overhead. Our BOND scheme is a truly localized polygon-based solution because computing $LVor\,(s_i)$ (or $TLVor\,(s_i)$) only needs one-hop information (this can be directly obtained from the Eqs. 3 and 4). Assuming that the number of trusted neighbors is k, each node can compute its own $LVor(s_i)$ with complexity smaller than $O(k)$. In addition, the computation of the $LVor(s_i)$ only involves some simple operations on polygons which can be efficiently implemented (e.g., PolyBoolean library [15]). We further simplify the detection process by constructing TLVPs first. For a densely deployed WSN, we have $LVor(s_i)$ or $TLVor(s_i) \to Vor(s_i)$, and it is well known in computational geometry that under the homogeneous spatial Poisson point process, the average number of vertices of $Vor(s_i)$ is 6 [17]. Therefore, when the node density is high, BOND on average only needs 4 to 6 nearest neighbors' information to successfully detect the boundary nodes. Moreover, when a neighbor node dies, BOND needs do nothing unless the dead node is used to construct the final $TLVor(s_i)$ or $LVor(s_i)$ in the last turn of LVP or TLVP construction. This unique property will greatly simplify the update of detection results and save precious energy of each sensor node. All these advantages cannot be achieved by other localized boundary node detection schemes in the literature, such as the perimeter-coverage checking approach [10] and the crossing-coverage checking approach [8], [24]. For lack of space, we refer to [23] for a detailed comparison.

Other Applications. We are aware of the following applications of BOND or its basic operations, which are by no means a complete list.

- *Topology control and routing*. It has been shown in [11], [16] that the dual of LVP, called LDT, can be used to design distributed topology control and routing protocols for wireless ad hoc networks with energy efficiency and the guarantee of the delivery. From the property of duality, we can directly obtain LVPs if LDTs are determined, or vise versa.
- *Spatial aggregation*. In distributed data processing for WSNs, to reduce the sampling errors in the aggregated spatial data, it is proposed in [19] to first calculate the VP of each sensor. As mentioned before, since the VP cannot be computed locally, the LVP can be used as a good approximation of the VP in spatial aggregation.
- *Coverage-preserving node sleeping scheduling*. Since sensor nodes are usually deployed with redundancy, it is possible to prolong the network lifetime while preserving the connected coverage by scheduling some nodes into the sleeping state [20]. Each node can locally decide whether its own LVP is covered by its neighbors. If this is the case, a node declares itself eligible for sleeping, announces this fact to its neighbors, and then goes to sleep. To avoid the forming of coverage hole caused by eligible nodes switching into sleep simultaneously, a randomly delayed announcement scheme using common random backoff approaches is proposed in [20].

IV. CMP: Coverage Measurement Protocol

A. Basic Protocol

Our design philosophy is that, since the minimum information required to describe the coverage is the positions of boundary nodes, we just need to detect boundary nodes. In other words, our scheme can ensure that, for the BS to reconstruct the "coverage image" without any distortion, the information transmitted from sensors to the BS is minimized. Also note that our BOND only involves local message exchanges. In a large-scale WSN, the overhead from local broadcasts is very small as compared to that from the end-to-end communications from sensor nodes to the BS. Therefore, our approach can save the precious energy of sensor nodes. Fig. 1 illustrates the basic operations of our BOND-based CIP. which consist of the following three steps:

[2] Other values will also work well.

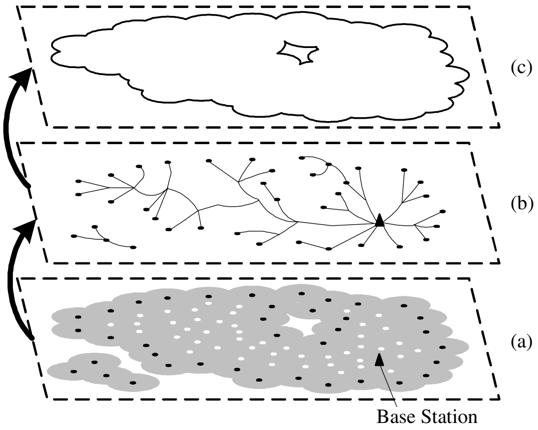

Fig. 1. **Basic operations of the CMP.** *(a) Each sensors executes BOND individually. (b) Boundary nodes (black dots in (a)) report themselves to the base station. (c) The BS reconstructs the coverage boundary. Note that the shaded area in (a) represents the coverage of sensors, and that the shaded area at the left bottom corner in (a) is lost in (c) because it is not the connected coverage.*

1) Neighborhood monitoring and self-detection: After the deployment of the WSN, each node first collects the position information of its neighbors and then executes BOND to detect whether it is a boundary node. If so, it will report its position to the BS. We refer to the neighbors used to construct the LVP or TLVP in the last run of BOND as its *consulting neighbors*.

In our protocol, both interior and boundary nodes are required to broadcast an *Existence Updating Packet* (EUP) to their neighbors for a random period of time exponentially distributed with rate T_{EUP}. In addition, each interior node, say s_i, maintains a timer $C_0(j)$ of expiry value much larger than T_{EUP} for each of its non-consulting neighbors, say s_j. If s_i does not overhear any packet (either an EUP or data packet) from s_j before $C_0(j)$ expires, it will treat s_j as a dead neighbor, which can become alive if s_i overhears any packet from it later. s_i also maintains two timers for each of its consulting neighbors, say s_k: the neighbor-monitoring timer $C_1(k)$ and the neighbor query timer $C_2(k)$. If s_i does not overhear any packet from s_k before $C_1(k)$ expires, it unicasts a *Neighbor Query Packet* (NQP) to s_k and starts $C_2(k)$. If still alive, s_k is required to send back an EUP immediately and wait for an ACK from s_i. If node s_i still does not overhear any packet from s_k before $C_2(k)$ expires, s_i will treat s_k as a dead neighbor and re-execute BOND with alive neighbors as input. In general, the expiry values of $C_1(k)$ and $C_2(k)$ should be in the same order of and much smaller than T_{EUP}, respectively.

Unlike previous neighbor-monitoring schemes employing a single timer [18] or treating neighbors as the same [9], our scheme set different timers for non-consulting and consulting neighbors. The major reason for doing so is that data packets and EUP-like broadcast packets are subject to loss due to wireless transmission errors or collisions. As a result, a node may falsely identify an alive neighbor as a dead one. Obviously, for non-consulting neighbors, we can decrease the false positive rate by setting a larger timer value. However, using a larger timer value for consulting neighbors will increase the response delay, i.e., the delay from when coverage holes emerge to when they are detected by boundary nodes. Therefore, we use two timers for consulting neighbors to ensure both a shorter response delay and a lower false positive rate: although the expiry value of $C_1(k)$ is small, we actively query the questionable neighbor before we treat it as the dead neighbor, which may significantly increase the accuracy of our scheme. As mentioned before, if the node distribution follows Spatial Poisson point process, each node only has 4 to 6 consulting neighbors on average, which means the high feasibility of our two-timer scheme.

Also note that the communication overhead of our neighbor-monitoring scheme is very small because the EUP packets can piggyback onto regular local broadcast packets used to learn link conditions, maintain the routing information, and facilitate other network operations. In addition, in the presence of data traffic, any packet overheard by a node should be regarded as an EUP packet, and any data packet sent from a node can cancel the next EUP packet it should broadcast.

2) Self-reporting of boundary nodes: Whenever identifying itself as a boundary node, a sensor node should send its position information to the BS which can reconstruct the "image of the coverage" based on all the received position information of boundary nodes.

3) Explicit ACKs from the BS: Since the packet loss ratio due to collisions or noise is pretty high in the WSN [12], boundary nodes need some mechanisms to ensure that their reports have been received by the BS. Otherwise, they have to repeatedly resend their reports which causes energy waste. A native solution is to require the BS to send individual ACK to each node. Here, we use Bloom filters [3] to design an energy-efficient approach for the BS to broadcast only one ACK to acknowledge multiple boundary nodes.

Let $s_1, s_2, \cdots, s_a$ be the a boundary nodes which the BS wants to acknowledge explicitly. Let $h_1, h_2, \cdots, h_b$ be the b hash functions of the Bloom filter, each with range $\{1, 2, \cdots, l\}$. Let $ack(t) = (b_1, b_2, \cdots, b_l)$ be a bit vector of length l. $ack(t)$ is the t-th Bloom filter used to indicate boundary nodes the BS wants to acknowledge, and it is initially set to $(0, 0, \cdots, 0)$. For $\forall i$, $1 \leq i \leq b$ and $\forall j$, $1 \leq j \leq a$, the BS computes $h_{ij} = h_i(ID_{s_j})$ where ID_{s_j} indicates the unique ID of node s_j, and sets $b_{h_{ij}} = 1$. The BS can than use the Bloom filter $ack(t)$ as ACK and send it back using efficient broadcast or geographic multicast protocols for WSNs. When a boundary node s_j receives the $ack(t)$, it performs a membership test: it computes $h_i(ID_{s_j})$ for $\forall i$, $1 \leq i \leq b$; if all of these positions are 1 in the $ack(t)$, then boundary node s_j knows its report has been received and acknowledged by the BS.

Note that the Bloom filter may induce a small number of false positives, i.e., a few un-acknowledged boundary nodes may pass the membership test and therefore believe that their reports have been received by the BS. On the other hand, Bloom filters ensure that there are no false negatives, i.e., all acknowledged nodes are guaranteed to pass the membership test. In practice we can tune our b and l parameters to enable tradeoff between communication and computational overhead and false positive rate.

B. Location-Error-Tolerant CMP

So far we have assumed that each node knows its accurate location. Our CMP can also be extended to tolerate bounded location errors. In this subsection, we assume that the location error (defined as the distance between the actual location of a node and its estimated location) is upper-bounded by δ. We then have the following theorem.

***Theorem** 1: If the location error is upper-bounded by δ, and a given node, e.g., s_i, is an interior node when all nodes are at their estimated locations and each node uses a sensing range $r_s - \delta$, node s_i must be an interior node when all nodes are at their actual locations and each node uses the actual sensing range r_s .*

Proof: We prove by contradiction. Let a and a' represent the actual and estimated locations of point a, respectively. Suppose that node s_i is a boundary node with the actual sensing range r_s when there is no location error. There must exist a point p in $LVor(s_i)$ that is not covered by $Disk(s_i, r_s)$. On the other hand, p must be covered by $Disk(s'_i, r_s - \delta)$ when p is still in $LVor(s'_i)$ or p is in $LVor(s'_j)$ for $s'_j \in Neig(s'_i)$. Note that here $LVor(s'_i)$ and $LVor(s'_j)$ are calculated with estimated locations and the sensing range $r_s - \delta$. We therefore have $\|s'_i - p\| \leq r_s - \delta$ or $\|s'_j - p\| < \|s'_i - p\|$ for $s'_j \in Neig(s'_i)$. Since $\|s_i - s'_i\| \leq \delta$ and $\|s_j - s'_j\| \leq \delta$, from the triangle inequality we have: $\|s_i - p\| \leq \|s'_i - p\| + \|s_i - s'_i\| \leq r_s$ or $\|s_j - p\| \leq \|s'_j - p\| + \|s_j - s'_j\| < \|s_i - p\|$. Hence, p is covered by $Disk(s_i, r_s)$ or p is not in $LVor(s_i)$, which contradicts our assumption that p is in $LVor(s_i)$ and not covered by $Disk(s_i, r_s)$. ■

From Theorem 1, we can design our location-error-tolerant CMP as follows. When the location error is upper-bounded by δ, our CMP can assume the sensing range of $r'_s = r_s - \delta$. Since $\delta > 0$, the condition that $r_c/r'_s \geq 2$ still holds and our BOND can correctly detect boundary nodes with r'_s and estimated locations. Based on Theorem 1, all the boundary nodes with the actual sensing range r_s and locations will be detected by BOND. It is worth noting that it is possible that some interior nodes in terms of their actual locations will be mislabelled as boundary nodes. Therefore, our location-error-tolerant CMP gives a conservative inference on the connected coverage, which is desirable for many WSN applications such as security-critical ones.

V. Conclusion

In this paper, we propose a coverage measurement protocol which can provide the BS an accurate and in-time measurement of the current connected coverage based on a novel computational geometric technique called LVPs.

As the future research, we plan to evaluate the performance of the proposed schemes in real sensor platforms. We also intend to further investigate other potential applications of our BOND scheme in WSNs, such as load balancing, topology control, distributed storage and network health monitoring.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Elsevier) Journal*, 38(4):169–181, 393-422 2002.

[2] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *MobiHoc'06*, Florence, Italy, May 2006.

[3] B. Bloom. Spaxe/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422C426, June 1970.

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

[5] M. Franceschetti, L. Booth, M. Cook, R. Meester, and J. Bruck. Percolation of multi-hop wireless networks. Technical Report UCB/ERL M03/18, EECS Department, University of California, Berkeley, 2003.

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

[7] P. Gupta and P. R. Kumar. *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, chapter Critical Power for Asymptotic Connectivity in Wireless Networks. Birkhauser, Boston, 1998.

[8] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley Sons Inc., New York, 1988.

[9] C. Hsin and M. Liu. Self-monitoring of wireless sensor networks. *Journal of Computer Communications special issue on Sensor Networks*, 29(4):462–476, February 2006.

[10] C. F. Huang and Y. C. Tseng. The coverage problem in a wireless sensor network. In *Proc. of the 2nd ACM international Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, San Diego, CA, September 2003.

[11] S. Kapoor and X. Li. Proximity structures for geometric graphs. In *WADS 2003*, Ottawa, Canada, July 2003.

[12] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley Sons Inc., New York, 2005.

[13] S. Kumar, T. H. Lai, and A. Arora. Barrier coverage with wireless sensors. In *MobiCom'05*, Cologne, Germany, August 2005.

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

[15] M. Leonov. Polyboolean library (2004). http://www.complex-a5.ru/polyboolean/.

[16] X. Li, G. Calinescu, P. Wan, and Y. Wang. Localized delaunay triangulation with applications in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, Oct. 2003.

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

[18] S.Chessa and P.Santi. Comparison based system-level fault diagnosis in ad-hoc networks. In *IEEE 20th Symp. on Reliable Distributed Systems (SRDS)*, New Orleans, LA, October 2001.

[19] M. Sharifzadeh and C. Shahabi. Utilizing voronoi cells of location data streams for accurate computation of aggregate functions in sensor networks. *GeoInformatica*, 10(1):9–36, March 2006.

[20] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. of the First ACM Inyernational Workshop on Wireless Sensor Networks and Applications (WSNA 02)*, Atlanta, GA, September 2002.

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

[22] 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.

[23] C. Zhang, Y. Zhang, , and Y. Fang. Localized coverage boundary detection for wireless sensor networks. In *Proc. of the Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, Waterloo, Canada, August 2006.

[24] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Network*, 1(1-2):89–123, January 2005.

[25] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, Anchorage, AK, May 2003.