Power Conservation Approaches to the Border Coverage Problem in Wireless Sensor Networks

Mohamed K. Watfa** and Sesh Commuri

Abstract- Recent advances in wireless communications and electronics have enabled the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances. As sensor nodes are typically battery operated, it is important to efficiently use the limited energy of the nodes to extend the lifetime of the sensor network. In this paper, the border coverage problem in sensor networks is rigorously analyzed. Unlike previous works in this area, we provide distributed algorithms that allow the selection and activation of an optimal border cover for both 2D and 3D sensor networks. We also provide self healing algorithms as an optimization to our border coverage algorithms which allow the border cover to adaptively reconfigure and repair itself in order to improve its own performance. Border coverage is crucial for optimizing sensor placement for intrusion detection and a number of other useful applications. Mathematical as well as experimental proofs are provided to validate the correctness and efficiency of our algorithms.

Keywords: *Ad-Hoc and Sensor Networks, Surveillance, Intrusion Detection, Border Coverage, Energy Savings.*

I. Introduction

Wireless sensor networks (WSNs) have been under development for many years and are about to gain widespread use as technology improves, prices drop, and new applications are developed [1]. Algorithms for wireless sensor networks must have low communication overhead, rely as much as possible on local information, adapt to failures and changes in network conditions, and produce results in a timely fashion. Given the requirements to minimize the power, it is desirable to select the bare essential number of sensor nodes dedicated for the task while all other sensor nodes should preferably be in the hibernation or off state. Even though target tracking has been widely studied for sensor networks with large nodes and distributed tracking algorithms are available [2], intrusion detection in ad hoc networks with micro sensor nodes poses different challenges due to communication, processing and energy constraints.

*Submitted to ICWN 06

**Corresponding author and Presenter

This work has been supported by U.S. Department of Defense, Army Research Office through grant # DAAD 19-03-1-0142.

Mohamed K. Watfa is with the School of Electrical and Computer Engineering at the University of Oklahoma in Norman, OK 73019. He can be reached at (405) 325-4302 (tel), (405) 325-3442 (fax); or at mwatfa@ou.edu (by email).
Sesh Commuri is an Associate Professor in the School of Electrical and Computer Engineering at the University of Oklahoma in Norman, OK 73019. His email id is scommuri@ou.edu

Border surveillance is one of the major applications of sensor networks. The border represents the physical extent of the region to be monitored and depending on the application, it is required to sense the intrusion into the monitored region or exit from the monitored region of the object being monitored. In a typical deployment of sensor nodes, sensor nodes are distributed across the entire region of interest and it is necessary to determine a minimal set of sensor nodes that can adequately monitor the border. Thus, it is necessary to find a scalable and energy efficient solution to the border coverage problem. Such a solution would extend the scalability of wireless sensor networks and enable the monitoring of one of the largest international borders [3].

The full coverage problem, which verifies if every point in the region of interest is covered by at least one active sensor, has been studied in a variety of contexts. Our previous work [4, 5] focused on the full coverage problem in 2D and 3D regions and provided algorithms to locate redundant sensor nodes in the region and deactivate them using simple geometric techniques. In [6], Gupta and Das design and analyze algorithms for self organization of a sensor network to reduce energy consumption. In particular, they develop the notion of a connected sensor cover and design a centralized approximation algorithm that constructs a topology involving a near optimal connected sensor cover. The works in [7] and [8] consider a large population of sensor nodes, deployed randomly for area monitoring. The goal is to achieve an energy-efficient design that maintains area coverage. As the number of sensor nodes deployed is greater than the optimum required to perform the monitoring task, the solution proposed is to divide the sensor nodes into disjoint sets, such that every set can individually perform the area monitoring tasks. Shakkottai et al. in [9] consider an unreliable sensor grid-network and derive necessary and sufficient conditions for the coverage of the region and connectivity of the network in terms of the transmission radius, sensing radius, and failure rate of the sensor nodes. In [10], Lieska et al. formulate coverage problems to address the quality of service (surveillance) provided by a sensor network. In particular, they address the problem of finding maximal paths of lowest and highest observabilities in a sensor network. The coverage concept with regard to the robot systems was introduced by Gage [11]. He defined three types of coverage: blanket coverage, barrier coverage, and sweep coverage.

In this paper, the problem of determining the minimum number of sensor nodes for covering the boundaries of a target region is addressed. Unlike the full coverage problem, here the primary interest is in the detection of the unauthorized movement of an object across the boundary. A recent work [12] considers the determination of holes in the coverage area of a sensor. In this work, the authors introduced a new technique for detecting holes in coverage by means of homology, an algebraic topological invariant. In

[13] Carbunar, *et. al.* study the problem of detecting and eliminating redundancy in a sensor network with a view to improving energy efficiency, while preserving the network's coverage. They also examine the coverage boundary detection by reducing it to the computation of Voronoi diagrams.

Unlike any other related works in this field, we first provide optimal two dimensional and three dimensional techniques for the deployment of a sensor network for border coverage of a given region. The auxiliary problem of selecting the minimal subset of previously deployed active sensor nodes for border coverage is then addressed. The energy efficiency of a WSN is studied in the context of the deployment of sensor nodes and the border coverage obtained. The minimum number of sensor nodes required for border coverage is used to specify a "measure of optimality" that can serve as a metric for the energy efficiency of a WSN. The border coverage algorithm developed is used to determine the savings that can be obtained by deactivating a subset of nodes while still maintaining the coverage of the region. Fault-tolerance is crucial for many systems and is becoming vitally important for computing- and communication- based systems as they become intimately connected to the world around them, using sensor nodes and actuators to monitor and shape their physical surroundings. We also extend our algorithms to allow the border cover to self heal itself in order to cover some border holes resulting from node failures or deaths and therefore, increase its own performance. The proposed techniques are analyzed mathematically and the algorithms are demonstrated through numerical examples.

The rest of the paper is organized as follows. The border coverage problem is formulated in section 2. A deterministic sensor deployment to guarantee border coverage of a region is provided in section 3. In section 4, distributed algorithms for selecting a optimal subset of nodes that lie on the border of the regions of interest are studied. Numerical simulation results that validate the proposed algorithms are presented in section 5. Possible extensions to our developed algorithms and conclusions are summarized in section 6.

II. PROBLEM FORMULATION

In this section, the notion of sensing region and border coverage are first defined. The border coverage and optimization problems are then formulated.

Let O_i be the output of a sensor S_i that is capable of sensing a phenomenon P. Let the sensing radius of the sensor node S_i be R_i. It is assumed that each sensor node is aware of its own location, the location of the boundaries of the region to be monitored and the location of its neighbors. This assumption is not too stringent and it can be satisfied by communications between adjacent nodes in the network on startup.

Definition 2.1: The phenomenon P located at $y \in R^3$ is said to be **detected** by sensor S_i located at $X_i \in R^3$ if and only if there exists a constant threshold δ such that

$$O_i(y) \begin{cases} \geq \delta & \text{if the phenomenon } P \text{ is present,} \\ = \varepsilon & \text{otherwise.} \end{cases}$$ □

The quantity 'δ' in the definition above is the signal threshold and is specific to the type of sensor used.

The sensing region of sensor S_i located at $X_i(x_i, y_i, z_i)$ is the collection of all points where the phenomenon P can be detected by the sensor S_i, i.e. $A_i = \{ y \in R^3 \mid P \text{ is detectable by } S_i \}$. In this paper, without loss of generality, we will restrict the sensing region of S_i to be a closed ball centered at $X_i \in R^3$. In the case of 2D, the sensing region is assumed to be a disk of radius R_s. The sensing boundary (circle) of sensor S_i, in this case, is denoted by Cir_i.

Most of the research works thus far assume simplified boolean sensing model (Circular disc) for coverage for protocol design and evaluation. In this model all events within the circular disc are assumed to be detected with probability 1. This simplified model is clearly not applicable to all types of sensing measurements however since the sensor nodes will be deployed in large numbers and there is a need for simulation and theoretical analysis, lead researchers in the areas of wireless sensor networks use this model in their research [1-9]. We also assume that any two nodes S_i and S_j can directly communicate with each other if their Euclidean distance is less that the communication range R_c. Although a network can be rendered useless if it looses its connectivity, we characterize the system lifetime by just observing the resulting border coverage. Zhang and Hou [14] showed that if the communication range is at least twice the sensing range, then complete coverage of a convex area implies connectivity among the nodes. Assuming the communication range is at least twice the sensing range ($R_c \geq 2R_s$), the theorems in [19] could be easily extended to handle border coverage of the region as well.

Definition 2.2: Let **R** be a subset of the (2D or 3D) space. The point 'p' is said to be **near R** if every neighborhood of 'p' contains a point from **R** i.e.

$\forall \varepsilon > 0, \exists x \in Ball(p, \varepsilon) \text{ and } x \in R.$ □

In the definition above, $Ball(p, \varepsilon)$ means the set of all points whose Euclidian distance from p is less than ε.

Definition 2.3: The set of all points in **R** and near **R** is called the **closure** of **R** and is denoted by $cl(R)$ i.e. $cl(R) = (R) \cup \{All\ points\ near\ R\}$. □

Definition 2.4: The **border** of a region **R** denoted by $B(\mathbf{R})$ is defined as the set of all points that are common to **R** and its complement i.e. $B(\mathbf{R}) = cl(\mathbf{R}) \cap cl(\overline{\mathbf{R}})$ where $\overline{\mathbf{R}}$ is the complement of the region **R** i.e. all the points that don't belong to **R**. □

An intruder is any object that is subject to detection by the sensor network as it crosses the border. A reasonable assumption is made that no intruder is aware of the location of the deployed sensor nodes. A region is said to be border covered if and only if an intruder is always detected as it crosses the border of the region.

Definition 2.5a: A set of sensor nodes C_{Border} is said to be a **border cover** of a region **R** if every point on the border of **R** belongs to the sensing region of at least one sensor in C_{Border}; i.e. $\forall$ p $\in$ $B(\mathbf{R})$, p $\in$ S_i for some S_i $\in$ C_{Border}. □

According to definition 2.5a, a region is border covered by a set of sensor nodes but that doesn't necessarily mean that the number of sensor nodes border covering **R** is minimum since there might exist some redundant sensor nodes.

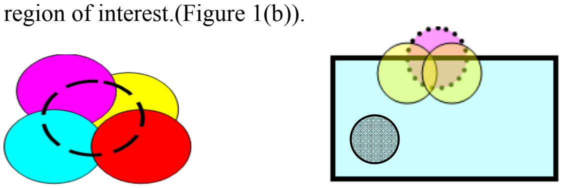
<u>*Definition 2.5b*</u>*:* A set of sensor nodes $C_{Border,Reduced}$ is said to be a **reduced border cover** of a region **R** if $\forall$ p $\in$ $B(\mathbf{R})$, p $\in$ S_i for some S_i $\in$ $C_{Border,Reduced}$ and no proper subset of $C_{Border,Reduced}$ is a border cover of **R**. i.e. $C_{Border,Reduced} - S_l$, *for any* $S_l \in C_{Border,Reduced}$, is not a border cover of **R**. □

<u>*Definition 2.6*</u>*:* A sensor node is called a **redundant sensor** node if its sensing region is completely covered by its neighboring sensor nodes. Deactivating a redundant sensor won't affect the full coverage of the region of interest. □

<u>*Definition 2.7a*</u>*:* A sensor node is called a **border redundant sensor** node if its intersecting sensing region with the border is completely covered by its neighboring sensor nodes. □

<u>*Definition 2.7b*</u>*:* A sensor node is called a **non-border sensor** node if its sensing region does not intersect the boundary of the region of interest. □

From definitions 2.7a and 2.7b, it can be easily seen that the deactivation of a border redundant sensor or a non-border sensor will not affect the overall border coverage of the region of interest.(Figure 1(b)).



(a) (b)

Figure 1: (a) Example of a redundant sensor (dashed circle). (b)Example of a border redundant sensor (dashed circle) and a non-border sensor (black shaded circle).

Using definitions 2.1 – 2.7, the border coverage problem is analyzed in this paper by dividing it into the following two sub problems:

I. **<u>Optimal Deployment for Border Coverage</u>**: Find the minimum number of sensor nodes and their placements for border coverage of a given region **R**.
II. **<u>Optimal Selection for Border Coverage</u>**: Given a dense deployment of sensor nodes in a region **R**, find a minimum subset of nodes that guarantee border coverage of **R**.

III. Optimal Deployment

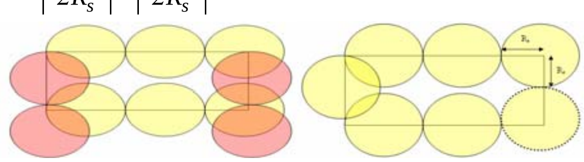
A critical issue in WSNs is the deployment and organization of the sensor network. Although many scenarios assume random deployment, such a deployment is not optimum and therefore a lot of energy is wasted due to multiple active nodes in a given region. When flexibility in deployment exists, it is advantageous to find an optimum border deployment of the sensor nodes so that border coverage can be achieved using a minimum number of nodes. In this section, theorems for optimal deployment of the sensor nodes are developed. These theorems provide lower bounds on the number of nodes needed to border cover both 2-dimesnional and 3-dimensional regions of interest.

3.1 Optimal 2D Deployment for Border Coverage

In the 2D deployment problem, the minimum number of sensor nodes modeled as disks and their locations for border coverage of a given rectangular region **R** are to be determined. While the region to be border covered is assumed to be a rectangular region, the algorithms could be easily extended to border cover any arbitrary shape of a region with minor modifications.

<u>Lemma 3.1.1:</u> Consider a rectangular region **R** of length '*L*' and width '*W*'. The lower bound on the number of sensor nodes needed to achieve border coverage of **R** is $2(\left\lceil \frac{L}{2R_s} \right\rceil + \left\lceil \frac{W}{2R_s} \right\rceil) - 3$ where $\lceil \ \rceil$ represents the operation of finding the least upper bound integer. □

<u>Proof:</u> The optimal way to deploy the sensor nodes to achieve border coverage of the region is to deploy the sensor nodes across the perimeter of the entire region such that any 2 adjacent sensor nodes that are on the same row or column are tangent to each other. $\left\lceil \frac{L}{2R_s} \right\rceil$ is the least number of sensor nodes to cover a line of length '*L*'. For a rectangular region of length '*L*' and width '*W*', the perimeter can be optimally covered by $2(\left\lceil \frac{L}{2R_s} \right\rceil + \left\lceil \frac{W}{2R_s} \right\rceil)$ sensor nodes. However, such a cover will have overlapping sensing coverage at the vertices of the rectangle as shown in Figure 2. A better way would be to select the next position of the center such that its circle intersects the last circle in its boundary intersection. Since the sensor nodes have equal sensing radii and their centers lie on the boundary of the rectangle, then in the best case scenario, $\frac{R_s}{2}$ of the boundary line will be covered. The best enhancement on the first placement of sensor nodes would be minimizing the number of sensor by one on three boundary edges (Figure 2). So, the lower bound on the number of sensor nodes to cover a 2D rectangular region each with sensing radius R_s is $2(\left\lceil \frac{L}{2R_s} \right\rceil + \left\lceil \frac{W}{2R_s} \right\rceil) - 3$. □



(a) (b)

Figure 2: (a) The optimal deployment of 10 sensor nodes modeled as circles in 2D to border cover a rectangular region. (b) Illustration of the best possible way to minimize the number of sensor nodes covering the border.

If we assumed that the region to be monitored is large in comparison to the sensing region of an individual sensor node, then the necessary and sufficient number of nodes to cover a rectangular region would simply be $2(\left\lceil \frac{L}{2R_s} \right\rceil + \left\lceil \frac{W}{2R_s} \right\rceil)$.

3.2 Optimal 3D Deployment for Border Coverage

The three dimensional optimal sensor deployment for border coverage is far more complex than that of the two dimensional case. It is addressed by determining the minimum number of sensor nodes required to cover the boundary of a cubical region of interest. Since the coverage region of a sensor node is modeled as a closed ball, the border coverage problem requires the determination of all the points on the surface of the cube that are covered by the sensor nodes. To address this issue, the intersection of the sensing regions and a boundary plane is first defined. This definition will then be used to determine the least number of sensor nodes required for border coverage.

Definition 3.2.1: A **great circle** on a sphere is the intersection of that sphere with a plane passing through the center of the sphere. □

Lemma 3.2.1: The centers of all the optimal deployed sensor nodes must lie on one of the faces of the cube. □

Proof: It is clear that each sensor covers a maximum area when the coverage region lies in the plane passing through the center of the sphere representing the sensing region. Thus, minimizing the number of sensor required to cover the surfaces of the region to be monitored is equivalent to maximizing the coverage area of each sensor. This is possible only when the centers of all the sensor nodes lie on the surface of the region to be monitored. □

Theorem 3.2.2: The optimal deployment locations of sensor nodes modeled as spheres to border cover a 3D cubical region is the locations of the spheres whose centers form a lattice of spacing $\Lambda = 1.7322R_s$ on each face of the cube. □

Proof: In a two-dimensional space, the optimal covering of a region with circles is obtained when the centers of the circles lie at the vertices of a hexagonal lattice. If the distance between adjacent vertices in this case is one unit, then the entire region can be covered by copies of a disc whose covering radius is $R_{cover} = \frac{1}{\sqrt{3}} = 0.5773$ (minimum sensing radius that will cover the region **R**). Such a lattice is also periodic and completely reduced. Moreover, the thickness of the cover is $\theta = \frac{2\pi}{3\sqrt{3}} = 1.2092$ (average number of sensor nodes that cover a point in the space). Thus, the deployment will be optimal if the spacing between the centers of adjacent discs equal $\Lambda = \frac{R_s}{0.5773}$. So, the optimal deployment to cover the border of a three dimensional cubical region is by placing the sphere at centers of the lattice on each face of the cube with spacing $\Lambda = 1.7322R_s$. □

Theorem 3.2.3: Consider a cubical region **R** of side 'a' ('a' is sufficiently large in comparison to R_s). An approximation on the lower bound on the number of sensor nodes of sensing radius R_s to achieve border coverage of **R** is

$$N_{min} = \left\lceil \frac{2.309a^2 - 3.845a}{R_s} \right\rceil$$ □

Proof: Our approach is based on the problem of covering a square by circles which has been studied by Kershner [15] and Verblunsky [16] where N_C, the least number of circles of unit radius which can cover a square was determined. They proved that there is a constant $c \geq \frac{1}{2}$ such that for all sufficiently large 'a', $a^2 + ca < \frac{3\sqrt{3}}{2} N_C < a^2 + 8a + 16$. Since there are 6 faces of a cube to be covered and we are interested in finding the lower bound of the number of sensor nodes needed, then $c = \frac{1}{2}$. Let R_s be the sensing radius of each sensor therefore a basic lower bound is $N_{min} = 2(\frac{2a^2 + a}{R_s\sqrt{3}})$. But since spheres can cover 2 or even 3 faces of the cube at the same time, the optimal way would be if the 2 faces intersect the sphere in semi great circles resulting in minimizing number of nodes by approximately $\frac{5a}{R_s}$. Since we are concerned with a lower bound $N_{min} = \left\lceil \frac{2.309a^2 - 3.845a}{R_s} \right\rceil$ is a valid lower bound on the three dimensional border cover. □

IV. OPTIMAL SELECTION

In practice, given an existing distribution of sensor nodes, it is often necessary to minimize the number of nodes that remain active while still achieving border coverage of the entire region. In this section, an algorithm is developed where the nodes make local decisions on whether to sleep or join the set of active nodes. The two-dimensional and three-dimensional cases for selecting an optimum border cover of a given region are studied. A measure of optimality is also proposed to compare the performance of the border coverage of a given sensor network with the optimum coverage obtained in section 3.

The border coverage algorithm presented in this section has the following key features:

1. It is a decentralized algorithm that depends only on the local states of the sensing neighbors.
2. It provides guaranteed degrees of border coverage.
3. It could easily be extended to handle different degrees of coverage.
4. It handles different sensing ranges.

4.1 A 2D Distributed Border Cover Selection Algorithm

In order to solve the border coverage problem for a two-dimensional region of interest, it is assumed that the region to be monitored is a rectangle specified by its vertices $V_1, V_2, V_3,$ *and* V_4. It is also assumed that all the sensor nodes are aware of the location of the vertices, i.e. the sensor nodes are aware of the extent of coverage that is required. The border coverage algorithm can be applied to any shape of boundary but the region of interest is assumed to be a rectangular region for the sake of ease of presentation. We will start by giving some definitions and assumptions that will aid us in developing an algorithm to select a border cover.

Let B represent the boundary of the region to be covered. $B = \bigcup_{i,j=1, i \neq j}^{4} B_{i,j}$ where $B_{i,j}$ is the segment connecting vertices V_i and V_j. Without loss of generality, suppose the boundary edges are ordered as $B_{12}, B_{23}, B_{34}, and\ B_{41}$ as shown in figure 3.

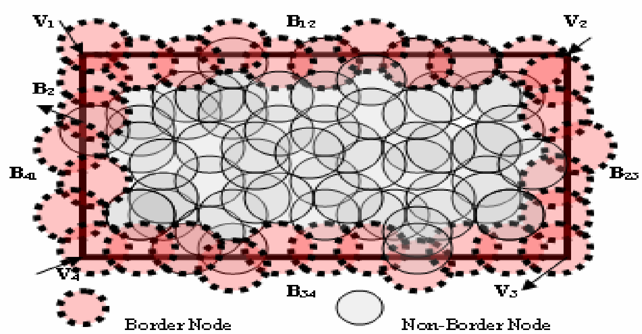


Figure 3: The border cover of a rectangular boundary.

Definition 4.1.1: An **intersection segment** is the portion of the boundary B covered by the sensing region of a sensor node and is represented by the closed interval [x, y] such that:

$[x, y]$ *is an intersection segment* $\Rightarrow \forall z \in [x, y]$,

$\exists\, i \in 1,..,n$ such that $z \in A_i$ *and* $x, y \in Cir_i$. □

A segment $Seg_i = [x_i, y_i]$ is represented by its start point x_i and end point y_i. Examples of intersection segments are shown in Figure 4.

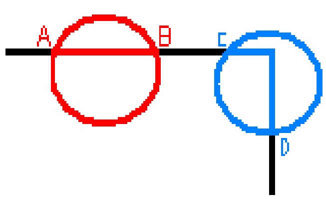


Figure 4: Intersection segments [A, B] and [C,D].

Since our algorithms depend on the concept of ordering, we define a mapping: $\varphi : \mathrm{B} \to [0,1]$ based on the distance metric from the nearest origin i.e.

$\forall x \in B_{ij}, \varphi(x) = \dfrac{d(x,V_i) + d(V_1,V_i)}{|P|}$ where |P| is the total length of the perimeter of the rectangular boundary and $d(V_i,V_j)$ is the distance along the boundary of the region i.e. for example $d(V_1,V_3) = |B_{12}| + |B_{23}|$. A special case should be taken for the sensing region covering the origin vertex v_1, where the resulting intersection segment is divided into 2 sub-segments each of which is mapped separately.

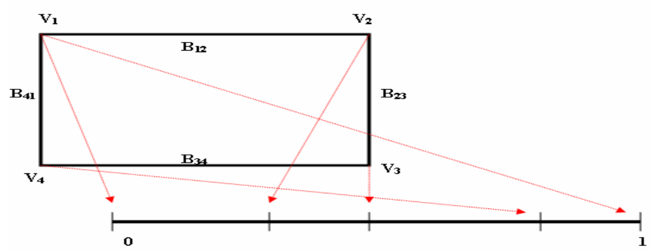


Figure 5: The ordering based on the mapping function.

Definition 4.1.2: We call $Seg_j = [x_j, y_j]$ the **successor** of $Seg_i = [x_i, y_i]$ denoted by $Seg_i \succ Seg_j$ if the following conditions are satisfied:

- $Seg_j \cap Seg_i = Seg_{ji} \neq \varnothing$
- $x_j > x_i$ & $y_j > y_i$
- there is no other starting point in Seg_{ji}. i.e.

  $\forall\ p \in Seg_{ji}, p \neq x_k$ *for some* $k \neq i, j$ □

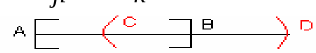


Figure 6: Example of a segment and its successor, $[A,B] \succ [C,D]$.

Theorem 4.1.1: Consider the set of segments $S = \{Seg_1, Seg_2, ..., Seg_m\}$ *where* $Seg_i = [x_i, y_i] \subset [0,1]$. Assume that no two segments are contained in each other i.e. $\forall Seg_i, Seg_j \in S, Seg_i \not\subset Seg_j, i \neq j$. A segment $Seg\ (=[x,y])$ is covered by $\bigcup_{i=1}^{m} Seg_i$ if and only if the following hold:

a. $\exists$ integers $1 \leq a,b,...,k \leq m \mid x \in Seg_a$ *and* $y \in Seg_b$

b. $Seg_a \succ Seg_b \succ ..... \succ Seg_k$. □

Proof: $Seg_a \succ Seg_b \Rightarrow Seg_a \cup Seg_b = [x_a, y_b]$.

Therefore, $\bigcup_{i=a}^{k} Seg_i = [x_a, y_b]$. Further from (a), $x_a < x$, and $y_k > y$. $[x,y] \subset [x_a, y_k] = \bigcup_{i=a}^{k} Seg_i \subset \bigcup_{i=1}^{m} Seg_i$. On the other hand, suppose that the segment [*x*, *y*] is covered by the segments $Seg_1,..., Seg_m$. Since the segment is covered, there exists some segment Seg_a such that $x \in Seg_a$. Similarly, there exists at least one segment '*k*' such that $y \in [x_k, y_k]$. Thus, condition (a) is easily satisfied. Now, if $y_a > y$, then $[x, y] \subset Seg_a$ and condition (b) is trivially satisfied. Otherwise, there exists a segment Seg_b such that $Seg_a \succ Seg_b$. If this was false, then it means that $y_a \geq y_k$ *for* $1 \leq k \leq m$. This would then imply that there exist points in the interval $(y_a, y]$ that are not covered, thereby contradicting the assumption that the segment [*x*, *y*] is covered. If $y_b > y$, then condition (b) is proved. Otherwise, repeating the process, we obtain integers *a…k* such that $Seg_a \succ Seg_b \succ ..... \succ Seg_k$ and $y_k > y$. Therefore, conditions (a) and (b) together imply that the segment [*x*, *y*] is covered by the collection of segments $Seg_1, Seg_2, ..., Seg_m$. □

Theorem 4.1.1 indicates that a sensor node is completely border redundant if each segment in the partitioning of its intersection segment by its neighbors' intersection segments has a successor and the end points are also covered. Therefore, to check if a sensor S_0 is a border redundant sensor and therefore could be deactivated without affecting the overall border coverage, one has to first find all the neighboring sensor nodes that lie on the border of the region of interest. For each neighboring sensor, find the resulting intersection segment (or segments) with the boundary lines and check if S_0 's portion of border coverage is completely covered by its neighboring sensor nodes. We can do that by using theorem 4.1.1. An algorithm is presented that illustrates the steps in this process.

<u>2D Distributed Border Coverage Algorithm</u>

For each node S_i, form the set of neighbors, $N(i)$ .Do the following:

<u>Step 1</u>: Find the intersection segments Seg_i

Find 'Seg_i' the intersection segment of S_i with the boundary of the region of interest and map it to [0, 1].

If $Seg_i = \varnothing$ or a point then S_i is a non border node.

Else go to step 2.

<u>Step 2:</u> Non containment property

Let $\overline{Seg_i}$ be the set of segments covering Seg_i and is initially set to $\varnothing$.

For every pair of nodes S_j , S_k in $N(i)$

- Find the common intersection segments $Seg_j = [x_j, y_j]$ and $Seg_k = [x_k, y_k]$ resp.
- If the end points appear in increasing order as x_j, x_k, y_k, y_j i.e. $Seg_k \subseteq Seg_j$ and can be ignored.
- Update $\overline{Seg_i}$ to include Seg_j i.e. $\overline{Seg_i} = \overline{Seg_i} \cup \{Seg_j\}$

<u>Step 3</u>: Check for endpoints coverage

Check that, $\exists\, Seg_f = (x_f, y_f)$ *and* $Seg_l = (x_l, y_l)$ in $\overline{Seg_i} \mid x_f \le x_i \le y_f$ *and* $x_l \le y_i \le y_l$.

If true go to step 4.

<u>Step 4</u>: Check for successor

Check that, for each element $Seg_m = (x_m, y_m)$ in $\overline{Seg_i} - Seg_l$, $\exists\, Seg_n = (x_n, y_n) \mid Seg_m \succ Seg_n$ and $m \ne n$.

If this condition is satisfied, the boundary intersecting segment Seg_i of the given sensor is completely covered and S_i is declared as a border redundant sensor node and can be deactivated without affecting the overall border coverage. The algorithm guarantees that every point on the boundary of the target region is covered by at least one sensor. The optimal set of sensor nodes is also selected.

The computational complexity of the redundancy selection algorithm developed in this section depends on $N = \left(\max_{i=1}^{n} \left| N(i) \right| \right)$,the maximum number of nodes in the neighbor set of any sensor in the network and n , the total number of sensor nodes in the network. The complexity of the border coverage algorithm is therefore $O(N^2)$. Since we have 'n' sensor nodes to be checked, then the complexity is $O(n.N^2)$. For large networks, the number of neighbors of any sensor is small compared to the size the network $(N \ll n)$ so the computational complexity of the algorithm for such large networks is of order 'n' ($O(n)$) where n is the total number of sensor nodes in the network. The key to our algorithm is that it is distributed and low in computational complexity.

4.2 A 3D Distributed Border Cover Selection Algorithm

The three dimensional optimal sensor border coverage problem is different than the two dimensional case. We will approach it from a different angle and try to transform it to optimal complete coverage of the sensor nodes in a 2D plane. The following lemma would be the key to addressing the three dimensional border coverage problems.

<u>Lemma 4.2.1:</u> The problem of 3D border coverage of a cube by sensor nodes modeled as 3D balls is equivalent to the problem of complete coverage of a 2D plane by sensor nodes modeled as circles. □

<u>Proof:</u> According to the definition of border coverage, each point on the border should be covered by at least one sensor. The border B(R) of the cubical region **R** is represented by 6 faces (2D planes). First, if each face of the cube $B_a \in B(R)$ *,a=1,2,3,4,5 and 6* is completely covered by a set of circles $Cir_a = \{Cir_{a1}, ..Cir_{an}\}$ and if D_i is the disc bounded by the circle Cir_i , then $\forall p \in B_a, p \in D_{ai}$ *for some* $Cir_{ai} \in Cir_a$. The 3-dimensional border coverage is transformed to finding the spheres whose intersections with the boundary are these circles. $\forall p \in B(R), p \in A_i$ *for some* $Cir_i = A_i \cap B(R)$. Now, if we have a set of sensor nodes that border cover a 3D cubical region, taking the intersection of the spherical sensing regions of the sensor nodes with each face of the cube will result in the formation of circles which completely cover the 2D plane. So, the 3-dimensional border coverage problem is transformed to the 2-dimensional full coverage problem. □

Lemma 4.2.1 indicates that a sensor node S_0 is border covered if its circle of intersection with the boundary plane is completely covered by its neighbors. To check if a circle is completely covered by the neighboring circles, we could use our previous results developed in [4, 5]. Therefore, to check if S_0 is border redundant; one has to first find all the circles obtained by the intersection of $S_0 \cap B_m, m = 1..6$. For each circle Cir_k , find all the intersection points that lie within D_k . If all these intersection points are covered, then the circles Cir_k are covered. Then, by lemma 4.2.1, S_0 is border redundant and can be deactivated without affecting the overall cubical border coverage.

Given the region to be monitored, one could easily find the number of sensor nodes required and their location for border coverage. However, if the sensor nodes are already deployed and a subset of these sensor nodes selected to keep active, then the *measure of optimality* is a measure of excess energy spent in monitoring the region as compared to an optimum deployment of the sensor nodes. A network with a lower '*measure of optimality*' would result in lesser expenditure of energy in monitoring the region.

V. SIMULATION RESULTS

The theoretical developments in sections 2-4 are validated through numerical examples in this section. The case of random deployment of sensor nodes is studied and compared to the optimum deployment for border coverage. Both 2D and 3D cases are considered and the number of sensor nodes required for border coverage is studied. The number of sensor nodes required to cover a 2D region of size 10 units by 10 units (or a 3D region of size 10x10x10) is considered. Random deployment, optimal deployment and optimal selection of the nodes for border coverage are studied for different values of the sensing radius. The optimization to our border algorithm is also tested and resulting border coverage lifetime of the network is analyzed.

In the 1st experiment, the optimum 2D coverage algorithm is used to find the optimum border cover of region 10x10 units when sensor nodes are randomly deployed. The nodes have a sensing radius of 1 unit and initially different numbers of nodes are randomly deployed in this region using a uniform distribution. It can be seen that the average *optimality measure* of our border selection algorithm is 1.228 and the nodes that were active in the optimum border cover resulted in average savings of 98.4% (Figure 8). In Figure 9, the required number of sensor nodes with different radii using random deployment, optimal 2D Border deployment and 2D Border selection algorithm are compared.

In the 2nd experiment, we evaluate the border coverage percentage of the region when the sensor nodes are randomly deployed and the border coverage selection algorithm is applied. As we vary the number of deployed nodes, we evaluate the border coverage of the region using the border cover obtained (Figure 10). It is noticed that after a specific threshold value, the border coverage percentage is always one. The reason is that random deployment of the sensor nodes does not guarantee border coverage of the region below that threshold.

In the 3rd experiment, we evaluate the system life time. The metrics used in evaluating system lifetime is the border *coverage lifetime.* The overall border coverage lifetime is the continuous operational time of the system before the border coverage drops below its specified threshold (for example 0.9). In Figure 11, the system lifetime is evaluated assuming the each sensor node has a limited energy supply (300 Joules) and when it runs out of energy it is deactivated. The node deployment densities are 300 and 600 respectively. The power consumption of Tx (transmit), Rx (receive), Idle and Sleeping modes are 1400mW, 1000mW, 830mW, 130mW respectively. As time passes, sensor nodes nodes will be deactivated due to lack of energy and will leave some coverage holes in the border of the region. If 300 sensor nodes were deployed, after approximately 1600 seconds, the border coverage percentage using the original network will drop below 0.9. However, using the border selection algorithm it needs about 2300 seconds to drop below the threshold. If we increase the number of deployed nodes to 600, the cost for calculating the border cover will increase and thus after approximately 1690 seconds the border coverage percentage will go below 0.9. In both experiments, the border coverage life time of the network using our border selection algorithm is much better that that using the original network. In Figure 12, the degree of border coverage is analyzed before and after running the algorithm. We divide the border in 1000 uniformly distributed grid points and check how many sensor nodes cover each center of each grid cell. It can be seen that the degree of border coverage is decreased and a minimum subset of nodes is only activated to cover the border of the region.

In the 4th experiment, we do the same comparison that was done in the 3rd experiment however with self healing optimizations added (Figure 11). The idea behind our self healing enhancement to our algorithm is that a border sensor node will send a HELP message to its neighboring border nodes (i.e. neighbor nodes that also intersect the border of the region) when its energy level goes below a specific value (it is about to die). The neighboring nodes, that will receive the HELP message, will activate them selves and will cover as much border coverage as possible. We notice that the system life time (border coverage life time) is much better than the case if we had started with the original set of deployed nodes. The strength of our developed algorithm is evident through this simulation result where the algorithm allows the sensor network to adaptively reconfigure and repair itself in order to improve its own performance. As we increase the number of deployed nodes, the self healing border coverage algorithm performs better since activating a border substitute set will result in better percentage of border coverage and therefore the border coverage lifetime of the network is increased.

In Figure 13, 2000 nodes were deployed, and after running the border selection algorithm, 1974 nodes were deactivated resulting in savings of 98.7%. Minimizing the number of sensor nodes active to border cover a region of interest will result in minimizing the energy consumed by the whole sensor network and thus increasing the life time of the network as demonstrated in the simulation results.

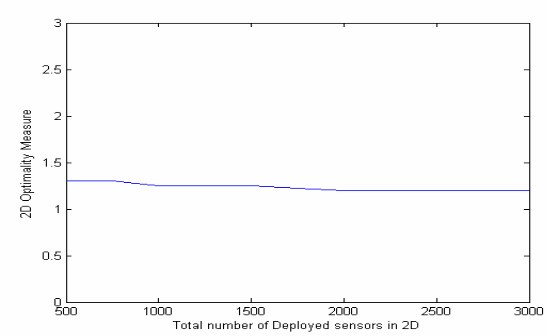


Figure 8: The optimality measure of the border selection algorithm.

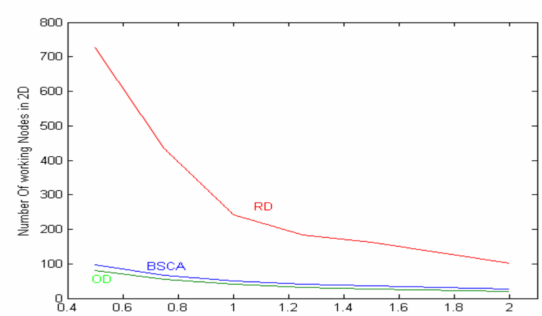


Figure 9: Comparison between Random Deployment, Optimal Deployment, and BSCA (Border Selection Cover Algorithm).

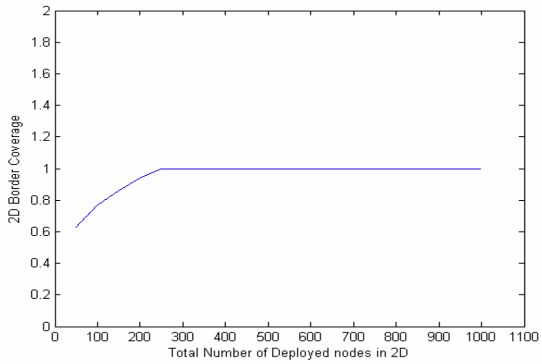


Figure 10: Varying the number of deployed nodes and the resulting border coverage after running the selection algorithm.

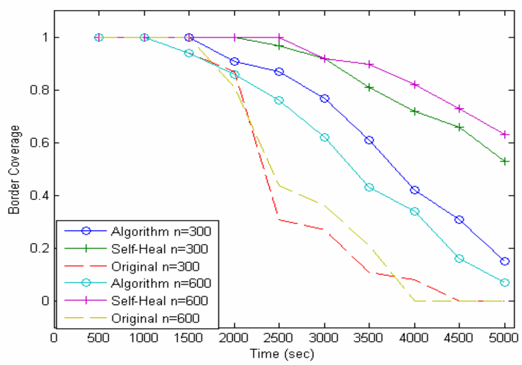


Figure 11: The border coverage life time of the region as time passes.

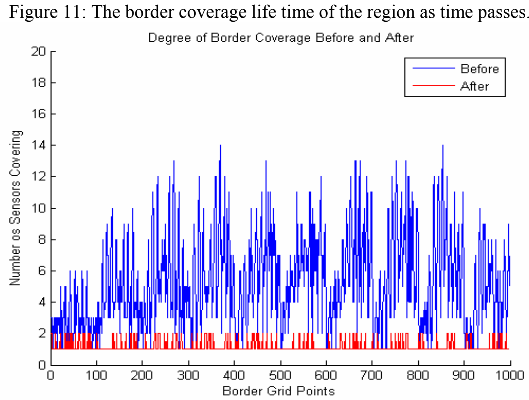


Figure 12: Degree of border coverage before and after running the algorithm

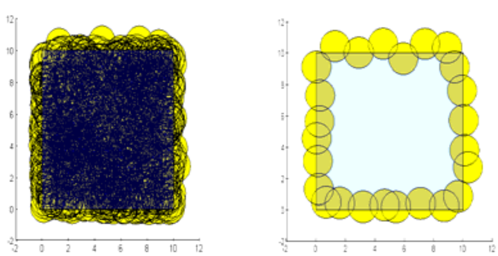


Figure 13: An example of the border selection algorithm. Active nodes, before and after running the algorithm are shown.

VI. Conclusions

In this paper, the border coverage problem in wireless sensor networks (WSNs) was formulated and analyzed. Algorithms were proposed to compute the minimum number of sensor nodes required for border coverage of a given region. A measure of optimality was also proposed that compares a given border deployment of WSN with optimum deployment. Part of our future work is to use the algorithms developed in this paper for tracking applications.

References

[1] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for self-Organization of a wireless sensor network," IEEE Personal Communication Magazine, pp. 16-27, October 2000.

[2] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," International Journal of High-Performance Computing Applications, vol. 16, no. 3, 2002.

[3] Extreme scale Wireless Sensor Networking. Technical report, http://www.cse.ohio-state.edu/exscal, 2004.

[4] M.Watfa, S.Commuri, "Optimal 3-Dimensional Sensor Deployment Strategy," Consumer Communications & Networking Conference, Las Vegas, 2006.

[5] M.Watfa, S.Commuri, "A Coverage Algorithm in 3D Wireless Sensor Networks," International Symposium on Wireless Pervasive Computing, Thailand, 2006.

[6] H. Gupta, S.R. Das, and Q. Gu, "Connected sensor cover: Self-Organization of sensor networks for efficient query execution," Proceedings of MobiHoc 03, Annapolis, Maryland, USA, pp. 189-200, June 2003.

[7] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," Journal of Interconnection Networks, vol. 3, pp. 213-229, 2000.

[8] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," Proceedings of IEEE International Conference on Communications, vol. 2, pp. 472-476, 2001.

[9] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," Proceedings of the 20th IEEE INFOCOM, pp. 1380-1387, 2001.

[10] K. Lieska, E. Laitinen, and J. Lahteenmaki, "Radio coverage optimization with genetic algorithms," Proceedings of IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), vol.1, pp.318-322, 1998.

[11] D. W. Gage, "Command control for many-robot systems," Proceedings of the 19th Annual AUVS Technical Symposium, pp. 22-24, 2002.

[12] Robert Ghrist and Abubakr Muhammad, "Coverage and hole-detection in sensor networks via homology," 4th International Conference on Information Processing in Sensor Networks (IPSN'05), Los Angeles, CA, April 25-27, 2005.

[13] B. Carbunar, A. Grama and J. Vitek. "Distributed and dynamic voronoi overlays for coverage detection and distributed hash tables in ad-hoc networks," 10th International Conference on Parallel and Distributed Systems, pp. 549-559, Newport Beach, 2004.

[14] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," Ad Hoc & Sensor Wireless Networks, an International Journal, vol. 1, UIUCDCS-R-2003-2351, June 2003.

[15] R. Kershner, "The number of circles covering a set," *Amer. J. Math.* , vol.61 pp.665-671, 1939.

[16] S. Verblunsky, "On the least number of unit circles which can cover a square," J. London Math. Society vol.24, pp.164-170, 1949.