This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Secon 2010 proceedings.

Divide and Conquer: Localizing Coverage holes in Sensor Networks

Harish Chintakunta
Electrical and Computer Engineering
North Carolina State University
hkchinta@ncsu.edu

Hamid Krim
Electrical and Computer Engineering
North Carolina State University
ahk@ncsu.edu

***Abstract*—Sensor Networks are inherently *complex networks*, and associated problems where analysis of some global features becomes more important than local ones, often arise. Localizing the holes in the overall coverage is one such problem. We present here, a distributed algorithm in a generalized combinatorial setting to localize holes in the coverage, with no a priori localization information for the nodes. We follow a divide and conquer approach, strategically dissecting the network so that the overall topology is preserved, while simultaneously minimizing the computational complexity. The detection of holes is enabled by first attributing a combinatorial object called a "Rips Complex" to each network segment, and by subsequently checking for the triviality of the first homology class of this complex. Our estimate approaches the location of the holes exponentially with each iteration leading to a very fast convergence coupled with optimal usage of valuable resources such as power and memory. We demonstrate the effectiveness of the presented algorithm with simulations.**

***Index Terms*—Networks, Sensors, Reliability, Topology, Graph Theory, Algorithms**

I. Introduction

The infrastructure of computing systems is rapidly transitioning from centralized systems to distributed and pervasive systems. A very important category of such systems are sensor networks which find applications in areas including Environmental monitoring, Health care and Military operations [9]. There has been considerable amount of research in this field in the past decade dealing with problems including node localization [10], distributed compression [11], probabilistic inference [12] and motion tracking. A unifying theme in the strategies is to glean consensus information by systematically combining the data collected at individual nodes in accordance to the structure of the network. The consensus information thus obtained characterizes the network, or the data in the network, as a whole and better represents the underlying phenomenon than can be inferred from the data in individual nodes. This reveals the fundamental nature of sensor networks, that they are essentially *complex networks* in which global patterns emerge from simple interactions between nodes. From an engineering perspective, the fundamental challenge in sensor network applications is coping with the limited resources; limited communication capability of nodes, i.e. nodes can communicate only with their neighbors, limited power and limited memory. Furthermore, sensor networks are often deployed in unaccessible locations and situations where maintenance is impractical; this makes careful use of exhaustible resources such as power, imperative. This unique combination motivates the use of techniques such as topological analysis, which directly extract global information without being overly dependent on the local structure and hence, also alleviating the need for excess recourses. In this work, we demonstrate the effectiveness of topological analysis by showing its exploitation to solve the hole localization problem. The use of topology for the coverage problem was introduced in [1], [2]. [3], [4] demonstrates distributed computation of homology groups and [5] attempts to localize the holes by posing the localization as an optimization problem. In this work, we further exploit the spatially constrained nature of the coverage holes and formulate a very effective "divide and conquer" algorithm by jointly invoking graph theory and Algebraic topology.

The balance of the paper is organized as follows; In Section II, we formulate the problem by a precise mathematical definition. Section III lays down the theoretical underpinnings of the algorithm presented and lays down the problem of hole localization in the context of Algebraic Topology. The algorithm along with its implementation details are presented in Section IV. In Section V, we demonstrate the algorithm with some simulations and in Section VI, we provide some concluding remarks along with some future directions

II. Problem Statement

We consider the scenario where N sensor nodes are randomly deployed in a region of interest. We denote the collection of all the nodes as the set $V = \{v_i\}$. Each node v_i can communicate with all the nodes within a circular neighborhood R_n^i and we denote these nodes as the set $N(v_i)$, the neighbors of v_i. A communication graph $G = (V,E)$ is thus formed as the collection of the set V together with the set of edges $E = \{(v_i, v_j)\}$ where $(v_i, v_j) \in E$ if and only if v_i and v_j can communicate with each other. We will make a important generalization of the notion of this communication graph to a simplicial complex in section III. The coverage area of the sensor on each node is assumed to be a circular neighborhood R_c^i centered at the node v_i. We make a reasonable assumption about the coverage areas R_c^i which will be necessary for a meaningful representation of the network by a simple simplicial complex, that is, the convex hull of the positions of nodes in a clique is contained in the union of their

978-1-4244-7151-5/10/$26.00 ©2010 IEEE

coverage areas. Let Q be a clique in G, then

$$conv(Q) \subseteq \bigcup_{v_i \in Q} R_c^i. \quad (1)$$

Let the union of areas enclosed by the outermost boundaries of connected components of the network be denoted by $\Re$. The problem is to first check whether the following relation holds

$$\Re \subseteq \bigcup_i R_c^i = R_c \quad (2)$$

where R_c is the total coverage space. The outermost boundary of a sensor network is to some extent in the control of the deployer and there are algorithms which can detect this boundary [8]. Therefore, as equation (2) suggests, we are mainly interested in the coverage of the region "inside" the network. Furthermore, if the relation (2) does not hold, we want to find the boundary $\partial(\Re \setminus R_c)$ of the uncovered region. We also assume that each node is given a unique integer as its ID. The setting presented here is similar to that in [5]. Note that we neither use any localization information of the nodes, nor do we invoke any directional information about their neighbors.

III. THEORETICAL FRAMEWORK

The tools presented here are mainly from the subject of *Topology* and *Algebraic Topology*. Topology [13] can loosely be construed to be the study of global organization of spaces without paying much heed to fine geometrical structure. For a space embedded in $\mathbb{R}^3$, this amounts to analyzing properties such as, is the space connected?, does the space wrap upon itself?, does the surface have holes in it? or does the surface enclose a three dimensional void? and so on. As such, the developed tools provide the right generalization to study organizational features of the network without expending resources on finer details. Defining a topology on a data space gives a notion of continuity bypassing the need to define metrics which are often difficult to find. This basic idea enables the characterization of *topologically equivalent* spaces or *homeomorphic* spaces.

Definition Two topological spaces X and Y are said to be *homeomorphic* if $\exists$ a bijective mapping $f : X \to Y$ such that f^{-1} is continuous.

The above definition implies that two homeomorphic spaces can be "smoothly" transitioned into each other. It can be shown that such spaces have the same topological characteristics as described above. This characterization is of particular interest to us because of the fact that if the coverage spaces $(\bigcup_i R_c^i)$ of two networks are homeomorphic, then they share the same coverage properties. Thus, our analysis is generalized to all networks with homeomorphic coverage spaces, fundamentally eliminating the need for localization information of the nodes. The above characterization can be further extended to spaces which are *homotopic* to each other by relaxing the requirement of a bijective mapping f and remarkably, homotopic spaces also share topological characteristics. This feature is exploited to obtain an elegant representation of the topological space by a simplicial complex in that the homology of the space is captured by that of the simplicial complex.

Algebraic topology [14] can be viewed as a generalization of Algebraic graph theory where we assign algebraic objects such as vector spaces and linear operators to combinatorial representations rather than just to a graph. This facilitates the computation and extraction of topological features. For example, the dimension of the first homology space computed for a topological space gives the number of holes. We should stress here that although we are working with vector spaces and linear operators, the theory can be developed using much less structured objects such as groups, rings, finite fields and homomorphisms defined on them . But our approach is to add as much structure as possible to simplify the computations.

A graph fundamentally is a combinatorial object in that it specifies relation between two vertices taken at a time. In fact, a graph $G = (V,E)$ together with the empty set ϕ can be viewed as $G \subseteq P(V)$, where $P(V)$ is the powerset of V with the properties that $x_1, x_2 \in G \Rightarrow x_1 \cap x_2 \in G$ and $y_1 \subset x_1 \Rightarrow y_1 \in G$. Note that these two properties are also satisfied by any collection of cliques in the graph, suggesting that the idea of a graph can be generalized. A particularly useful generalization to us is the simplicial complex.

Definition A simplicial complex $K = \{\sigma_i\}, K \subseteq P(V)$ is a subset of the power set defined on a set of vertices V. Its elements σ_i are known as simplices and the subsets of σ_i are known as faces of σ_i. The simplices have the following properties:

- $\sigma_i, \sigma_j \in K \Rightarrow \sigma_i \cap \sigma_j \in K$
- All faces of σ_i are in K.

A simplex with $k+1$ points is called a k dimensional simplex or a k-simplex (σ_i^k) and a k dimensional simplicial complex contains k-or lower dimensional simplices. Ex: A graph is a 1-dimensional simplicial complex. Each simplex can be assigned an orientation by specifying an order to its points and we can add some algebraic structure to these simplices by giving them a sign defined as

$$\begin{aligned} \sigma_i^k &= (v_1, \ldots, v_l, \ldots, v_m, \ldots, v_{k+1}) \\ &= -(v_1, \ldots, v_m, \ldots, v_l, \ldots, v_{k+1}) = -\sigma_i^k \end{aligned} \quad (3)$$

Two simplices σ_i^k and σ_j^k are said to be *upper adjacent* ($\sigma_i^k \frown \sigma_j^k$) if they are faces of the same $k+1$ dimensional simplicial complex and *lower adjacent* ($\sigma_i^k \smile \sigma_j^k$) if they have non-empty common faces.

Given a set of points V in a finite dimensional Euclidean space and a fixed distance ε, the *Vietoris-Rips* Complex or *Rips* Complex $R_\varepsilon(V)$ is a simplicial Complex in which each simplex is a set of points which are pairwise within Euclidean distance ε of each other. We will represent the coverage space using the Rips Complex to extract its properties, i.e., its homology.

A. *Homology Spaces*

A vector space C_k is obtained by considering all oriented k-simplices as basis vectors. A sequence of such vector spaces

$\{C_k\}$ is assigned to a simplicial complex. An element of the chain space C_k is called a *k-chain*. A *boundary operator*, $\partial_k : C_k \rightarrow C_{k-1}$ maps a k-simplex to a $k-1$-chain which "enclose" this simplex. For example, if we take a 2-simplex (a triangle), the operation of ∂_2 on this simplex will give us the sum of its three edges (1-simplices). Mathematically, the boundary operator is defined as

$$\partial_k(\sigma_i^k) = \sum_{j=0}^{k} (-1)^j (v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_{k+1}) \quad (4)$$

The structure of the definition (4) together with (3) results in a basic property of the boundary operators that

$$\partial_{k-1} \circ \partial_k = 0 \quad (5)$$

This property is the essence of *Chain Complexes* using which homology spaces can be defined. A chain complex C is the sequence of chain spaces $\{C_k\}$ together with linear operators called boundary operators which have the property $\partial_{k-1} \circ \partial_k = 0$. Pictorially, it is represented as:

$$0 \rightarrow C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \cdots C_0 \rightarrow 0 \quad (6)$$

Definition Given a simplicial complex K and a chain complex C defined on it, The k^{th} *homology space*, $H_k(K)$ is determined as the quotient space

$$H_k(K) = ker(\partial_k)/img(\partial_{k+1}) \quad (7)$$

where $ker(.)$ denotes the null space and $img(.)$ denotes the range space. Note that (5) implies that $img(\partial_k) \subseteq ker(\partial_{k-1})$ and therefore the above definition (7) for homology spaces is well defined. The chains which belong to the same equivalence class in the homology space are called *homologous* and differ from each other only by chains in $img(\partial_{k+1})$, i.e., chains which enclose higher dimensional simplices. For example, let us consider the first Homology space. The 1-chains (σ_i^1) in the space C_k can be equivalently viewed as paths on a graph and $\sigma_i^1 \in ker(\partial_1)$ implies that σ_i^1 is a closed path. The space $img(\partial_2)$ contains the 1-chains which are boundaries produced by the action of ∂_2 on 2-simplices (triangles) and their sums. Firstly, note that if there are no holes in the graph (considering triangles as filled), then any closed path can be expressed as a sum of boundaries of these triangles. Therefore, the first homology group will be trivial i.e., zero-dimensional since $ker(\partial_1) = img(\partial_2)$. If there is a hole in the graph, then the boundary σ^1 of the hole is a closed path and hence $\sigma^1 \in ker(\partial_1)$. Any other closed path which also encircles this hole but is not a boundary will differ from σ^1 by boundaries of some triangles and therefore will belong to the same equivalence class as σ^1. This suggests that the dimension of Homology space will be equal to the number of holes in that space. This can be rigorously shown and is true for any order (H_k) homology space. A 2-dimensional hole will correspond to a void in space and so on. The dimensions of the homology spaces are called *betty numbers* $\{b_k\}$. We are interested only in holes in the coverage space and we will restrict our focus to just the first homology spaces.

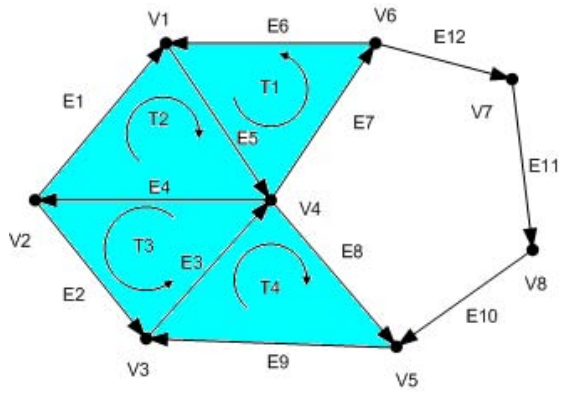


Fig. 1. A simplicial Complex. Directed edges are oriented 1-simplices. The orientation of 2-simplices (triangles) is clockwise or anticlockwise as shown

Example: Consider the simplicial complex (X) shown in Figure 1. The orientation of the simplices (1 and 2-dimensional) are chosen arbitrarily; the homology spaces assigned will be independent of this choice. Consider the 1-chains (paths): c_1, the outermost boundary, c_2, the closed path enclosing the triangles and c_3, the closed path enclosing the hole, all in clockwise directions. These chains in terms of the basis vectors (the simplices) are expressed as

$c_1 = E1 - E6 + E12 + E11 + E10 + E9 + E2$

$c_2 = E1 - E6 - E7 + E8 + E9 + E2$

$c_3 = E12 + E11 + E10 - E8 + E7$

Note that equation (3) states that a change in the sign of a simplex changes its orientation. Using equation (4) for the boundary operator, we can see that

$\partial_1(c_3) = \partial_1(E12) + \partial_1(E11) + \partial_1(E10) - \partial_1(E8) + \partial_1(E7) = V7 - V6 + V8 - V7 + V5 - V8 - (V5 - V4) + V6 - V4 = 0$

Similarly, any closed path (including c_1 and c_2) can be show to belong to $ker(\partial_1)$. Again using equation (4), the action of the ∂_2 operator, on $T1$ for example, is given as

$\partial_2(T1) = E6 + E5 + E7.$

It should be easy to verify that c_2 can be expressed as $c_2 = \partial_2(T4 - T3 + T2 - T1)$ and therefore, $c_2 \in img(\partial_2)$ Note also that $c_1 - c_3 = c_2$, i.e., c_1 and c_3 differ by a chain in $img(\partial_2)$ and are therefore, homologous. In other words, they encircle the same hole. To compute the $H_1(X)$, first observe that any closed path on X can be expressed as a sum of the closed paths surrounding the four triangles and that surrounding the hole. Therefore, $ker(\partial_1)$ is a 5 dimensional vector space, i.e., $ker(\partial_1) \cong \mathbb{R}^5$. Also, the four closed paths generated by the action of ∂_2 on the four triangles (the basis vectors for C_2) are linearly independent and therefore, $img(\partial_2) \cong \mathbb{R}^4$. From the definition (7) of homology space as a quotient space, we can see that $H_1(X) \cong \mathbb{R}$. The first homology space is one dimensional, corresponding to one hole in the complex.

B. Rips Complex and Coverage Space

In the previous section we described the homology spaces of a simplicial complex, a discrete combinatorial object. But the coverage space R_c which we have to analyze is embedded in the Euclidean space $\mathbb{R}^2$ and has an infinite number of points. The homology spaces are similarly determined as in the discrete case except that the chain spaces in the chain complex contain different dimensional manifolds embedded in this continuous space. For example, C_1 consists of all the curves embedded in the space which are analogous to 1-chains (paths) on the simplicial complex and closed curves $\in ker(\partial_1)$ analogous to closed paths. The chain spaces thus obtained would be of infinite dimension but remarkably, the homology spaces of a simplicial complex will be isomorphic to that of a related continuous space called the shadow of the simplicial complex.

Definition Given a simplicial complex K, the shadow $S(K)$ of K is the natural projection onto $\mathbb{R}^d$ obtained by mapping each simplex affinely onto the convex hull of its vertices where d is the dimension of K.

Theorem 3.1: Given a simplicial complex K and its shadow $S(K)$, The homology spaces $H_k(K)$ and $H_k(S(K))$ are isomorphic [6].

We form our simplicial complex K by taking all the cliques in the communication graph G as simplices. Because of the way the communication graph was formed, K thus obtained would be a Rips Complex $R_\varepsilon(V)$ where ε is the distance within which the nodes can communicate. Now, given assumption 1 holds, the shadow $S(R_\varepsilon(V))$ will be homotopic to the coverage space R_c, i.e., R_c can be continuously deformed to $S(R_\varepsilon(V))$ and hence their homologies will be isomorphic (see figure 2). From this and theorem 3.1, we conclude that the homology of $R_\varepsilon(V)$ and R_c are isomorphic and it suffices for our purposes to analyze just the Rips complex $R_\varepsilon(V)$. In order to facilitate this analysis, we use Laplacian Operators.

C. Laplacian Operators

Section III-A describes how homology spaces can be computed by the quotient space of $ker(\partial_k)$ with respect to $img(\partial_{k+1})$. We contend that Laplacian operators defined on $R_\varepsilon(V)$ offer a more efficient method to determine the homology spaces and the corresponding Betty numbers. The k^{th} Laplacian operator $L_k : C_k \to C_k$ on $R_\varepsilon(V)$ is defined as

$$L_k = \partial_{k+1} \circ \partial_{k+1}^* + \partial_k^* \circ \partial_k \tag{8}$$

where ∂_k^* is the adjoint of ∂_k and given a basis, the matrix representation of ∂_k^* would be the transpose of that of ∂_k. It can be shown that the null space of the k^{th} laplacian operator is isomorphic to the k^{th} Homology space, i.e., $ker(L_k) \cong H_k(R_\varepsilon(V))$. In fact, observe that L_k is a sum of two non-negative definite operators and therefore, $c^k \in ker(L_k)$ implies $\partial_{k+1}^*(c^k) = 0 \Rightarrow c^k \perp img(\partial_{k+1})$ and $\partial_k(c^k) = 0, c^k \in ker(\partial_k)$. For example, in case of 1-chains (paths on the graph), $c^1 \in L_1$ means that c^1 is a closed path since it $\in ker(\partial_1)$ and since it is $\perp img(\partial_2)$, it does not contain the boundaries of any 2-simplices (triangles). Thus, it represents an equivalence class of H_1. By dealing with Laplacian operator, we are not only preserving the information required to extract homology, but we now have a single symmetric matrix which can be efficiently analyzed in a distributed fashion instead of two boundary operators.

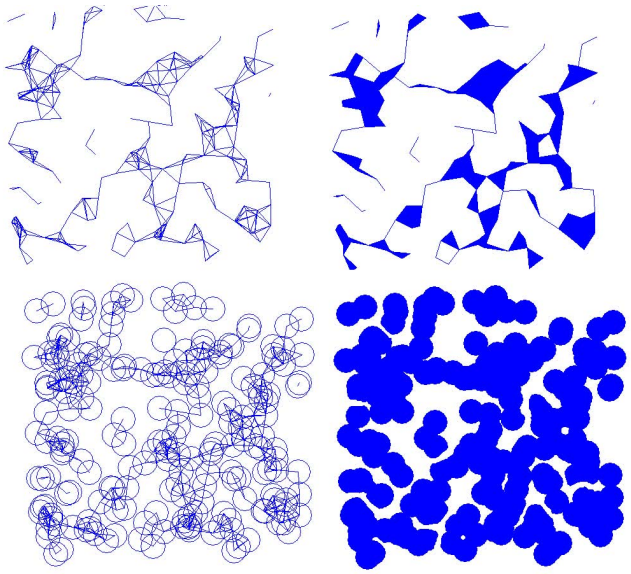


Fig. 2. Clockwise from top-left, (a)Communication graph G (b)Rips Shadow$S(R_\varepsilon(V))$, (c)Coverage Space R_c, (d)Individual Coverage spaces of nodes

D. Exact Sequence

An exact sequence is a sequence of vector spaces $\{A_k\}$ along with Linear operators $\{h_k\}$, pictorially represented similar to (6), such that $img(h_{k+1}) = ker(h_k)$. We will be using a particular exact sequence called the Meyer-Vietoris [14] exact sequence later in section IV-B to prove necessary and sufficient conditions for a segmentation to preserve the topology.

IV. Hole Localization

Each element in the first homology space H_1 represents an equivalence class of homologous closed paths encircling a hole in the coverage space. As such, *Localizing* the exact boundary of this hole is in essence a problem of finding the smallest closed path in a equivalence class. A very direct approach was taken in [5] where the authors formulate the localization as an optimization problem to find the sparsest chain in the H_1 space. Such an approach is effective but it leads to a very slow convergence and involves all the nodes in the network to participate in the optimization. Although the presence of holes in the coverage space is a global property, the boundary of the hole is constrained to a certain relatively small part of the network. Our ability to *detect* a hole in this region does not depend in any way on the configuration of nodes in other parts of the network. We exploit this idea to transform the problem of *identification* of boundary of holes to a much simpler problem of *detection* of holes.

We accomplish this transformation by iteratively dissecting each partition in the network into two smaller partitions and detecting the presence of holes in these smaller partitions. The nodes in the partition where no hole is detected go into *sleep* mode and no longer participate in the algorithm thus saving valuable power. The only active nodes will be in partitions with non-trivial homology (with holes in coverage) which get further dissected. Thus, we will be converging onto the exact boundary of the holes rapidly with each iteration. In the first iteration, each connected component of the network graph G is treated as a partition.

The dissecting strategy is to minimize the "size" of the resulting partitions and simultaneously preserving the overall topology. Note also that a very important aspect of the algorithm presented here, is that it can be implemented distributively in the network by passing messages between nodes $\{v_i\}$ and their neighbors $\{N(v_i)\}$. An overview of the algorithm is given in Table I which constitute the main parts described in the sections to follow.

TABLE I
HOLE LOCALIZATION OVERVIEW

for all partitions with non-trivial homolgy
\\ Dissection
step 1: Find the diameter nodes.
step 2: Find the boundary nodes and divide the partition.
\\ Detecting holes in each of the above two partitions
for the two partitions formed
step 3: Compute the Laplacian Matrix L
step 4: Check the rank deficiency of L
repeat.

A. *Finding Diameter Nodes*

Firstly, we elaborate on what we mean by "size" in the above description. The time required for completion of steps 1 and 2 directly depends on the *diameter* of the network partition. Step 4 involves a gossip algorithm whose convergence does depend on the diameter, but other factors could also come into play [15]. Therefore, a segmentation resulting in minimization of diameter of the smaller partitions is optimal for minimizing run time. This is facilitated by finding a pair of nodes called the *diameter nodes* defined as

$$(\bar{x},\bar{y}) = \arg\max_{(x,y)} d(x,y) \tag{9}$$

where $d(x,y)$ is the shortest path between nodes x and y in terms of hop count on the partition of interest. In general, such a pair will not be unique and ties are broken by a simple protocol which chooses the pair that has the node with the smallest ID. We find the boundary nodes in two stages; First, we find all the candidate nodes C_{dia} by assigning a scalar field $f(x)$ equal to the farthest distance for each node x on the current partition and then selecting the nodes with maximum f. Second, amongst the candidate nodes, we break the ties with the above mentioned criterion.

$$\begin{aligned} f(x) &= \max_y d(x,y) \\ C_{dia} &= \{x | f(x) = \max_y f(y)\} \end{aligned} \tag{10}$$

For computing f on G, we use a simplified version of the *Dijkstra's* algorithm. The simplification comes because unlike the setting for Dijkstra's, 1)we do not need the shortest paths but rather just the distances and 2)we do not need the shortest distance from a node x to all other nodes but rather just the *max* of these distances.

*1) **Computing** f:* In what follows, we first give a mathematical justification for the algorithm proposed and then describe the algorithm.

Let $A = \{a_{ij}\}$ be the adjacency matrix for G, then $A^n = a^n_{ij}, n > 0$ where a^n_{ij} is the number of paths of length n form i to j. For simplicity, we assume that i is the ID given to node v_i. The only way this assumption would affect the algorithm is that we use i in place of ID(i). Now, the shortest distance from v_i to v_j, $i \neq j$ is given by

$$d(v_i, v_j) = \arg\min_{n>0} a^n_{ij} > 0, i \neq j \tag{11}$$

The matrix A^n can be represented distributively in the network where node v_i computes and stores the i^{th} row. This can be computed iteratively as $A^{n+1} = A \cdot A^n$, and a^{n+1}_{ij} at node v_i is obtained as $a^{n+1}_{ij} = \sum_{v_k \in N(v_i)} a^n_{kj}$. This computation is enabled by all the nodes broadcasting their row to their neighbors. From equation (11), we conclude that if m is the smallest integer such that $a^m_{ij} > 0$, then $\forall n > m, a^n_{ij}$ are insignificant implying that each node v_i can broadcast information about a node v_j to its neighbors only once. At the n_{th} iteration, a node v_i discovers all the nodes $\{v_j\}$ such that $d(v_i, v_j) = n$, broadcasts this set $\{v_j\}$ to all its neighbors and stores the set in a table. The table used, acts as a reference to avoid transmitting duplicate information to its neighbors. Here, it appears that the memory required at each node will be equal to the number of nodes in the partition as all the nodes will be eventually discovered. We contend that it suffices to store a node in the table for only two iterations.

Theorem 4.1: A node v_i storing the information about the node v_j for two iterations guarantees no duplicate information is broadcasted.

Proof: By contradiction.

Duplicate information will be broadcasted if a node v_i discovers a node v_j at iterations m and $m+t, t > 2$. This means that there are two paths $P = (j, p_1, \ldots, p_{m-1}, i)$ and $Q = (j, q_1, \ldots, q_{m+t-1}, i)$. At the m^{th} iteration, node v_i will start a broadcast which propagates along Q in the reverse direction and meets the message coming along Q at node $q_{m+t_1} = q_{m+t-t_1}$. $\exists t_1 = (t-1)/2 or (t-2)/2$ whichever is an integer such that $(m+t-t_1)-(m+t_1) = 1 or 2$. The message from v_j would take the path Q only if node q_{m+t_1} broadcasted it thus violating the rule because v_j is already in the table at that instant. ■

TABLE II
FINDING DIAMETER NODES

At each Node i in the segment
\\ **Computing** f
\\ Initialization: Discover itself
add v_i to table and broadcast to $N(i)$
\\ run time
at iteration n:
\\check for new nodes discovered
if found new nodes
broadcast new nodes to $N(i)$
add new nodes to table
clear values of n-2 iteration
else
$f(v_i) = n$
stop.

TABLE III
FINDING BOUNDARY NODES

At each Node i in V_X
\\Initialization
if v_i is a diameter node
broadcast i to $N(i)$. stop.
(i will serve as the segment ID)
else
wait until reception
if received two distinct IDs
broadcast the lowest received ID to $N(i)$.
v_i = boundary node.
wait one time interval
if received two distinct IDs overall
v_i = boundary node.
stop.

If n_{max} is the largest distance for which a node v_i discovers a new node, then we set $f(v_i) = n_{max}$. A summary of this algorithm is given in table II.

*2) **Diameter nodes**:* Once f is computed, candidate diameter nodes are found by consensus for maximizing f on the network by a simple gossip algorithm. There are many algorithms in the literature for computing such aggregates on the network, for example [16]. The essence of such algorithms is that at each iteration, if a node "discovers" a new max value, it broadcasts this discovered value to all its neighbors. Similarly, the diameter nodes are obtained from the candidate nodes by consensus for minimum of node IDs.

B. Finding Boundary Nodes

As the physical positions of the nodes do not change, we form a virtual segmentation by finding boundary nodes $B = \{b_i\}$ within a partition which stop messages from passing through. This separates the given partition into two parts whose nodes cannot intercommunicate. For the set B to behave like a boundary [1], it has to satisfy certain properties:

Definition Let $X = (V_X, E_X) \subseteq G$ be a connected sub-graph. The set of nodes B is said to be a boundary in X if and only if $\exists$ two disjoint sets $V_{X1}, V_{X2} \subset V_X$ such that there is a node $b_i \in B$ in any path $(v_i, \ldots, v_j)$ where $v_i \in X_1$ and $v_j \in X_2$. Furthermore, $V_{X1} \cup V_{X2} \cup B = V_X$.

If every path from V_{X1} to V_{X2} contains a boundary node, this means there is no path along which a message from V_{X1} can reach V_{X2}, thus virtually separating both. This justifies the above definition for the boundary. The boundary nodes identify their neighbors as belonging to V_{X1} or V_{X2} and do not transmit messages from one to the other.
To minimize the diameter of the resulting partitions ($S_1 = V_{X1} \cup B$ and $S_2 = V_{X2} \cup B$), we choose the boundary nodes to be equidistant from the determined diameter nodes. This way, the boundary nodes bisect the diameter of X. These equidistant nodes are obtained using a simple flooding algorithm which is presented in Table III. The basic idea is to start a flood from both the diameter nodes and determine the boundary nodes where these floods meet.

Every node will either belong to S_1 or belong to S_2 because X is connected and therefore, $S_1 \cup S_2 = X_1 \cup X_2 \cup B = X$. Let the diameter points be x_1 and x_2 and let $v_1 \in X_1$ and $v_2 \in X_2$. This implies v_1 and v_2 received a single ID, $\mathrm{ID}(x_1)$ and $\mathrm{ID}(x_2)$ respectively. Then for any path $p = (v_1, \ldots, v_2), \exists v_i \in p$ such that v_i received both the IDs and hence belong to B. This shows that the nodes obtained as in table III indeed satisfy the definition of the boundary.

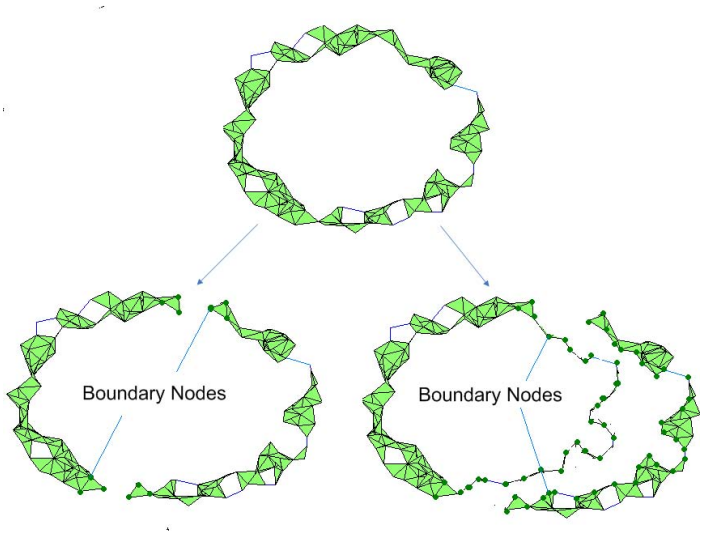


Fig. 3. The partitioning on the left does not preserve topology whereas that on the right does.

There is an additional very important property such a boundary partitioning should satisfy, i.e., it should preserve the topology (see figure 3). Specifically, if X has no holes in its coverage, then neither of S_1 nor S_2 should, and if X has a hole, then it should be preserved in either one of the partitions. Theorem 4.2 shows that a sufficient condition for preserving topology is that the Rips complex obtained from

[1]This definition of boundary should not be confused with the conventional notion which represents the closure of a graph or a region. The particular definition we are using will be clear from the context

induced subgraph on the boundary nodes B is contractible. The Rips complex for X is obtained by taking all the cliques as simplices and similarly for the subgraphs induced on S_1, S_2 and B.

Theorem 4.2: Let X be a Simplicial complex and $A,B \subset X$ be sub complexes such that $A \cap B$ forms a boundary on the underlying graph. Then $H_1(X) = H_1(A) \oplus H_1(B)$ if $A \cap B$ is contractible, i.e., $H_0(A \cap B) = \mathbb{R}$ and $H_1(A \cap B) = 0$.

Proof: For any simplicial complex X, and $A,B \subset X$, $\exists$ the following exact sequence called the Meyer-Vietoris Exact sequence.
$H_1(A \cap B) \xrightarrow{\phi} H_1(A) \oplus H_1(B) \xrightarrow{\Psi} H_1(X)$
where ϕ and ψ are linear operators. Now,
$H_1(A \cap B) = 0 \Rightarrow img(\phi) = 0 \Rightarrow ker(\psi) = 0 \Rightarrow \psi$ is injective since it is linear. Therefore, $H_1(A) \oplus H_1(B) \subseteq H_1(X)$
Let $c \in ker(\partial_1(C_1^X))$ be a chain in the null space of the first boundary operator acting on the first chain space of X (c is a closed path), such that it contains $v_1 \in A$ and $v_2 \in B$. $\exists b_1, b_2 \in c, b_1 \neq b_2$ such that b_1, b_2 also $\in A \cap B$ since $A \cap B$ is a boundary. Now, since $A \cap B$ is connected, $\exists$ a chain corresponding to the path $b_1 \rightarrow b_2$. Consider the two chains corresponding to closed paths $c1 := (v_1 \rightarrow b_1 \rightarrow b_2 \rightarrow v_1)$ and $c2 := (v_2 \rightarrow b_2 \rightarrow b_1 \rightarrow v_2)$. It immediately follows that $c1 + c2 = c$. Therefore, any chain in $ker(\partial_1(C_1^X))$ can be expressed as a sum of chains in $ker(\partial_1(C_1^A))$ and $ker(\partial_1(C_1^B))$, $\Rightarrow H_1(X) \subseteq H_1(A) \oplus H_1(B)$.
$\Rightarrow H_1(X) = H_1(A) \oplus H_1(B)$. ■

The first part of the theorem states that no new holes are created by the partitioning, and second states that all the holes are preserved. If the boundary nodes obtained by algorithm given in Table III are not connected, we can form a tree by joining different connected components by a shortest path between them. This shortest path can be discovered by a simple flooding in the network originating at the connected components. The boundary obtained is also usually contractible, however, there is one rare exception. As shown in Figure 4, this happens exactly when $d(x_1,v_1) = d(x_1,v_2) = d(x_2,v_3) = d(x_2,v_4)$ in the given configuration where x_1 and x_2 are the diameter nodes. In this case, all the nodes v_1, v_2, v_3, v_4 will be made boundary nodes. But, note that we use the contractibility condition only to prove that no new holes are created and clearly it is not so in this case.

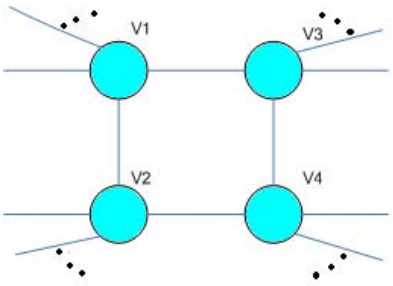


Fig. 4. Exception case when $B \ni v_1, v_2, v_3, v_4$ is not contractible

C. Detecting Holes

After each segmentation, the task is to detect holes in each segment. If a hole is detected, the segment will be further dissected or else, put into sleep mode. As stated in Section III-C, the kernel of L_1 is isomorphic to H_1, therefore, it suffices to check for its rank deficiency to detect holes. The Laplacian operator can be computed distributively within the network [4]. The first step is to discover all the simplices in the network. Since we are only interested in L_1, we only need to discover all the 2-simplices and this can be done by each node passing two messages. Once the simplices are discovered, L_1, as a matrix where the basis vectors are the simplices, can be computed using some simple rules based on the adjacencies of the simplices. Furthermore, we need to update L_1 only on the boundary nodes after each segmentation. Theorem 4.3 can be used to check its rank deficiency.

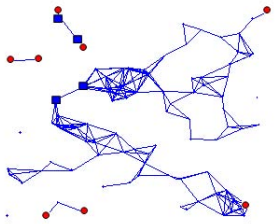
Theorem 4.3: Let L_1 be a symmetric non-negative definite matrix with spectrum $\sigma(L_1)$ and spectral radius $\rho(L_1)$. Then, L_1 is rank deficient if $\rho(\rho(L_1)I - L_1) = \rho(L_1)$

Proof: Let x be an eigenvector corresponding to the eigenvalue $\lambda \in \sigma(L_1)$. Then $(\rho(L_1)I - L_1)x = (\rho(L_1) - \lambda)x$. $\Rightarrow x$ is also an eigenvector of $(\rho(L_1)I - L_1)$ and its eigenvalue is $\rho(L_1) - \lambda$. Furthermore, L_1 is non-negative definite $\Rightarrow \lambda \geq 0 \Rightarrow \rho(\rho(L_1)I - L_1) \leq \rho(L_1)$ If L_1 is of full rank, then $\lambda > 0$. $\Rightarrow \rho(\rho(L_1)I - L_1) < \rho(L_1)$. ■

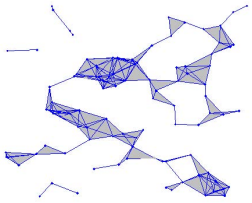
The spectral radius of L_1 can be computed using the power iteration method searching for the largest eigenvalue and it can be distributively executed in the network [7]. The convergence of the power iteration method (for eigenvector) is slow when the difference between the largest and second largest eigenvalue (δ) is small but the eigenvalue itself quickly comes very close to the actual value. An exception where we might detect a hole when there is none, is the case when the smallest eigenvalue is very close to zero, but this problem is unlikely to happen in successive iterations. In any case, it always helps to precondition the matrix for faster convergence. Since the preconditioning should also be distributively carried out, it is not straightforward and will be a part of our future work.

V. Simulations

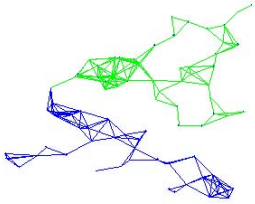
We generated a random graph with 100 nodes in the $\mathbb{R}^2$ plane and formed the communication graph G (Figure 5(a)) by connecting each node with all the nodes within a radius. The Rips Complex $R_\varepsilon(V)$ extracted from G is shown in Figure 5(b). Note that simplices only up to dimension 2 are shown for clarity; 1-simplices by the edges, 2-simplices by triangles. The diameter nodes obtained in the first iteration are highlighted in Figure 5(a) by circles and the boundary nodes by squares. These boundary nodes split the connected component they are in into two partitions. These partitions will be active only if they enclose holes in them. Figure 5(c) shows the active partitions after the holes are detected for the first time. That the smaller components which did not have any holes are not shown. The boundary nodes are colored as one of the partition,



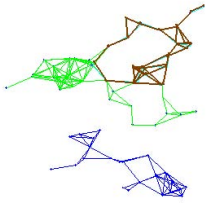
(a) Communication Graph G. Diameter nodes as circles and Boundary nodes as squares



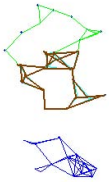
(b) Rips Complex extracted from the communication graph



(c) Active nodes after first segmentation. Segments differentiated by colors.



(d) Active nodes after two iterations



(e) Active nodes after three iterations (f) Active nodes after four iterations

Fig. 5. Simulations showing convergence to hole locations

but note that they actually belong to both the partitions. Figures 5(d) and 5(e) show the resulting active partitions after further successive partitions and in figure 5(f), the only active nodes are those which exactly surround the hole. The extra 2-simplices which survive can easily be put down since the nodes already discovered all the simplices during computation of the laplacian operator.

Note only 4 segmentations were required to converge to the hole location. Each iteration has simple steps and barring the computation of the spectral radius of laplacian operator, all other steps have deterministic computation time upper bounded by the diameter of the partitions. The power iteration was deemed to converge if the $L1$ norm between the old and the updated vector was less than 0.01. The average number of iterations for this convergence was 12 in our experiment, i.e., each segmentation takes 12 iterations on an average. Overall, this is a very significant improvement in contrast to 4000 iterations required for optimization as in [5]. Furthermore, we also achieve a tremendous savings in power as nearly half the active nodes are put to sleep after each segmentation.

VI. Conclusion

We presented here an elegant integration of Topological Analysis, Graph theory and distributed computation techniques to solve the problem of localization of holes in the coverage space. Although the theoretical setting is abstract, the computations and the messages required are very simple. Our algorithm converges very quickly and our estimation of the hole location improves with each iteration. Furthermore, the number of nodes participating in the algorithm reduces exponentially with each iteration, thus saving power. Although the problem of focus in this paper is coverage space, our analysis method is fairly general in that it applies to any topological space which can be defined on a network.

There are some rare exceptions where we might miss a hole due to numerical factors, therefore improving this accuracy will be part of our future work. There are many steps in each iteration, therefore a careful analysis of reliability issues is imperative which will also be a part of our future work.

References

[1] R. Ghrist and A. Muhammad, *Coverage and hole-detection in sensor networks via homology* in Proceedings of the Fourth Inernational Symposium on Information Processing in Sensor Networks, Los Angeles, CA, April 2005, pp. 254260.

[2] V. de Silva and R. Ghrist, *Coordinate-free coverage in sensor networks with controlled boundaries via homology* The International Journal of Robotics Research, vol. 25, no. 12, pp. 12051222, 2006.

[3] A. Muhammad and M. Egerstedt, *Control using higher order Laplacians in network topologies* in Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006, pp. 10241038.

[4] A. Muhammad and A. Jadbabaie, *Decentralized computation of homology groups in networks by gossip* in Proceedings of American Control Conference, New York, NY, 2007, pp. 34383443.

[5] Alireza Tahbaz-Salehi and Ali Jabdabaie, *Distributed Coverage Verification in Sensor Networks Without Location Information* in proceedings of the 47th IEEE conference on Decision and Control. 2008

[6] E. W. Chambers, V. de Silva, J. Erickson, and R. Ghrist, *Rips complexes of planar point sets* under Preperation. [Online]. Available: http://www.cs.uiuc.edu/∼jeffe/pubs/rips.html

[7] D. Kempe and F. McSherry, *A decentralized algorithm for spectral analysis* in Proc. of the thirty-sixth annual ACM symposium on Theory of computing, pp. 561568, 2004.

[8] Yue Wang, Jie Gao and Joseph S.B. Mitchell,*Boundary recognition in sensor networks by topological methods*. Proceedings of the 12th annual international conference on Mobile computing and networking. Pages: 122 - 133. 2006

[9] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. *Wireless sensor networks: a survey*

[10] Jehoshua Bruck,, Jie Gao, Anxiao (Andrew) Jiang. *Localization and routing in sensor networks by local angle information*. ACM Transactions on Sensor Networks (TOSN). 2009

[11] Naoto Kimura and Shahram Latifi. *A Survey on Data Compression in Wireless Sensor Networks*. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC05)

[12] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. *Generalized Belief Propagation*. 2000

[13] William F. Basener. *Topology and its applicaitons*. Wiley publications. 2006.

[14] Allen Hatcher. *Algebraic Topology*. Cambridge University Press. 2002

[15] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar and Devavrat Shah *Randomized Gossip Algorithms*. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 52, NO. 6, JUNE 2006

[16] Jen-Yeu Chen, Gopal Pandurangan and Dongyan Xu*Robust Computation of Aggregates in Wireless Sensor Networks: Distributed Randomized Algorithms and Analysis*. 2005.