Summary of Coverage Problems in Wireless Ad-hoc Sensor Networks

Laura Kneckt
21th of March, 2005

Abstract

This paper presents a brief summary of the article about Coverage Problems in Wireless Ad-hoc Sensor Networks. The referred article presents an algorithm which suits best for testing the deployment of the sensor nodes and evaluating the Quality of Service based on the coverage.

1 Introduction

1.1 Goal: Sensor Network Coverage

Coverage can be considered as the measure of Quality of Service of a sensor network. In coverage problems, the most significant factors are the ability of a network to observe a given area and what are the changes that it detects in given time frame. The article introduces two algorithms for calculating the Quality of Service. The first algorithm, worst-case coverage, finds areas of lower observability from sensor nodes and detects breach regions. The second algorithm, best case coverage, is analogous to the worst case coverage algorithm. The best-case coverage algorithm tries to find areas of high observability from sensors and identify the best support regions.

The highlight of the article is said to be the optimal polynomial time worst and average case algorithm for coverage calculation. The continuous geometric problem is transformed into a discrete graph problem and solved with solved with the help of Voronoi diagram and Delaunay triangulation.

This summary paper is organized as follows:
In the section 2 we look at the prerequisites to the algorithms and discuss about ad-hoc networks, particularly the power consumption and costs. The third section defines the difference between deterministic and stochastic coverage and in fourth section we give the details about the coverage algorithms. The section 5 consist the performing of algorithm and experimental results. Section 6 points out some criticism about the applicability of the algorithm to the ad-hoc networks.

2 PRELIMINARIES

The models of sensor behaviour varying with different degrees of complexity share one thing in common. Sensing ability in the models is directly dependent on the distance in a sense that

$$sensor\_coverage = \frac{1}{(distance\_from\_sensors)}$$

The coverage algorithms relie on geolocation information of the sensor nodes by considering only nodes that have valid location information. Some of the sensor nodes, called beacons, are asssumed to know their coordinates in advance, either from satellite information (GPS) or pre-deployment. Nodes approximate their neighbour distance from the signal strength information. Each node locates itself by trilateration and becomes a beacon by hearing at least three beacon neighbours.

<u>The Voronoi diagram</u>, used in the maximal breach path algorithm, can be built into the plane with randomly placed discrete set of points (sites). For all pairs of neighbour sites, the Voronoi diagram draws a line equidistant from both sites (see figure 1). The lines are cut as they cross with another thus the lines become an edges. The edges produce convex polygons around all sites except the sites at boundaries of the net. The minimal distance between edge and closest sites of the edge is maximized by this definition.

<u>The Delaunay triangulation</u> connects the neighbour sites producing triangles. The smallest angle of each Delaunay triangle is maximized compared to all pos-

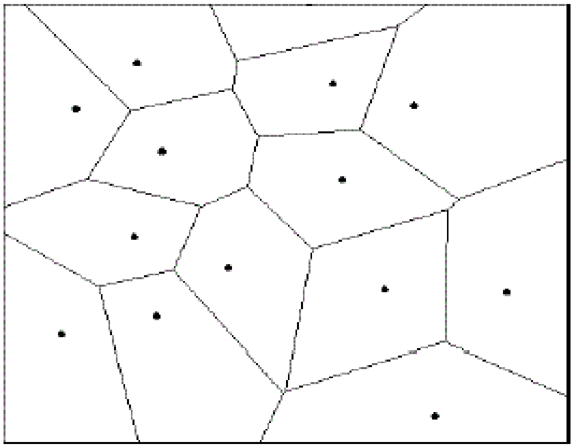

Figure 1: Voronoi diagram of a set of randomly placed points in a plane

sible triangulations.

The number of hops is determined by reducing the power consumption. The requied communication between two arbitrary nodes is defined by

$E = B \cdot d^y$, where
d is the distance between the two nodes,
$y > 1$ is the path lost exponent and
B is proportionality constant describing the overhead per bit.

The equation shows that energy is strongly dependent on the distance between the two nodes. Even though the equation gives an idea to use infinite number of hops over the smallest possible distances by reducing the power consumption, we need to take account of:

- The number of intermediate hops is limited by the number of nodes between the two communicating nodes
- Each radio needs energy for receiving a bit and readying a bit for retransmission

The costs of transmission can be reduced by finding a balance between computation - communication, meaning that whether trying to compress the message using the energy of transmitting sensor node or sending an uncompressed message causing energy consumption to each retransmitter.

3 Deterministic vs. Stochastic Coverage

In deterministic coverage, the static network is constructed by deploying the nodes according to a predefined locations. In stochastic coverage, the sensor nodes are randomly distributed to the environment, using uniform, Gaussian, Poisson or some other distribution. The coverage algorithm can be used in both, deterministic and stochastic coverage.

4 The Coverage Algorithms

4.1 Worst Case Coverage - Maximal Breach Path

Given: A field A instrumented with sensors S where for each sensor $s_i \in S$, the location (x_i, y_i) is known. Areas I and F corresponding the initial and final locations of an agent.
Problem: Identify P_B, the Maximal Breach Path in S, from I to F.
P_B is defined as a path through the field A, with endpoints I and F and with the property that for any point p on the path P_B, the distance from p to the closest sensor is maximized, thus the P_B must lie on the line segments of the Voronoi diagram.

The lines at the boundaries of the Voronoi diagram extend to infinity. Since we are dealing with a finite area A we clip the Voronoi diagram to the boundaries of A, inserting the boundaries of A to the Voronoi diagram. By this construction we are dealing with bounded Voronoi diagram.

The algorithm first generates the Voronoi diagram corresponding to the sensors in S. By creating a node for each vertex and an edge corresponding to each line segment (in the Voronoi) diagram and given each edge a weight equal to its minimum distance from the closest sensor in S, the algorithm constructs the weighted, undirected graph G. A binary search is performed between the smallest and largest edge weights in G.

In each step, the graph G' is initialized with all the vertex found in G and taking only account the edges with weights larger than or equal of the search criteria (*breach_weight*). The Breadth-First-Search is used to check wether there is a path between I and F. If the path exists, the search criteria (*breach_weight*) is increased. If the path is not found, the *breach_weight* is decreased.

After performing, the algorithm has found the Maximal Breach Path through the graph starting from I and ending in F and with the highest weighted edges. The Maximal Breach Path may not be unique, depending of the search criteria. The algorithm solves Maximal Breach Path with at least one edge equal to *breach_weight* and rest of edges larger than the *breach_weight*

The Worst Case Coverage algorithm:

Generate Bounded Voronoi Diagram for S with vertex set U and line segment set L. Initialize weighted undirected graph G(V,E)
FOR each vertex $\mathbf{u_i} \in U$
Create duplicate vertex v_i in V
FOR each $l_i(u_j, u_k) \in L$
Create edge $e_i(v_j, v_k)$ in E
Weight(e_i)=min distance from sensor $s_i \in S$ for $1 \leq i \leq S$
Initialize the variables
min_weight = min edge weight in G
max_weight = max edge weight in G
range = (max_weight - min_weight) / 2
breach_weight = min_weight + range
WHILE (range ¿ binary_search_tolerance)
Initialize graph G'(V',E')
by inserting every vertex
FOR each $v_i \in V$
Create vertex v_i' in G'
and insert the edges with weights large enough
FOR each $e_i \in E$
IF Weight(e_i) $\geq$ breach weigth
Insert edge e_i' in $\overline{G}'$
Update the binary search variable
range = range / 2
IF BFS(G',I,F) gives a path between I and F
breach_ weight = breach_weigth + range
ELSE
breach_ weight = breach_weigth - range
END IF

4.2 Best Case Coverage - Maximal Support Path

The best-case coverage algorithm is similar to the worst-case algorithm and can be stated as:

Given: A field A instrumented with sensors S where for each sensor $s_i \in S$, the location (x_i, y_i) is known; areas I and F corresponding to initial and final locations of the agent.
Problem: Identify the path of maximal support in S, starting in I and ending in F.

P_S is defined as a path through the field A, with endpoints I and F and with the property that for any point p on the path P_S, the distance from p to the closest sensor is minimized.

While minimizing the distance from sensors, the path must lie on the straight lines connecting sensor nodes with minimum distance between the sensor nodes. This path is found by the algorithm which uses Delaunay triangulation. The support algorithm has the following exceptions compared to the breach algorithm:

- The Voronoi diagram is replaced by the Delaunay triangulation
- The edges in graph G are assigned weights equal to the length of the corresponding line segments in the Dalaunay triangulation
- The search parameter $breach_weight$ is replaced by the new parameter *support_weight*.

4.3 Complexity

The complexities of the subalgorithms are

- for generating the Voronoi diagram, $O(n \cdot \log(n))$, where n is the number of vertex
- for BFS $O(\log(m))$ where m is the number of edges
- for binary search $O(\log(range))$

According to the referred article, the worst case complexity of the algorithm is $O(n^2 \cdot \log(n))$ and in practice the networks are sparse and the overall complexity is $O(n \cdot \log(n))$ dominated by the Voronoi diagram.

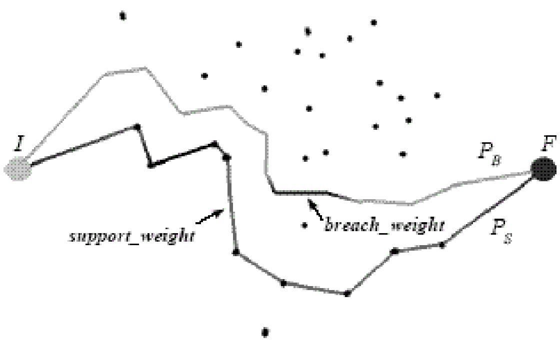

Figure 2: Sensor field with Maximal Breach Path (P_B) and Maximal Support Path(P_S)

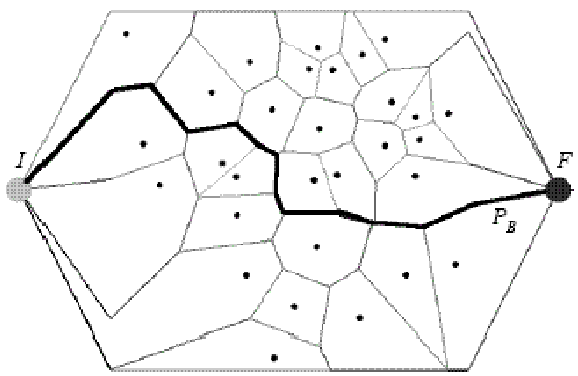

Figure 3: Sensor field with weighted Voronoi diagran and Maximal Breach Path (P_B)

5 EXPERIMENTAL RESULTS

5.1 Experimentation Platform - Sample Results

This section presents the sampling results. In figure 2, 30 sensor nodes are deployed randomly and coverage algorithms are run to find the support-path and breach-path. The figure also locates the edges, which corresponds the value of *breach_weight* and *support_weight*.

Figure 3 shows the bounded Voronoi diagram together with the resolved breach path. Extra edges with weight equal to zero connects the I and F regions to the structure so that all possible paths can be considered in the search algorithm. Figure 4 depicts the

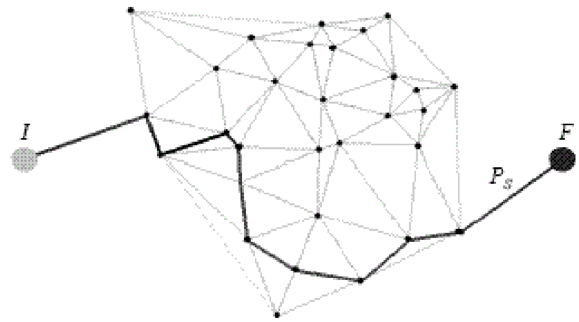

Figure 4: SEnsor field with weighted Delaunay triangulation and Maximal Support Path (P_S)

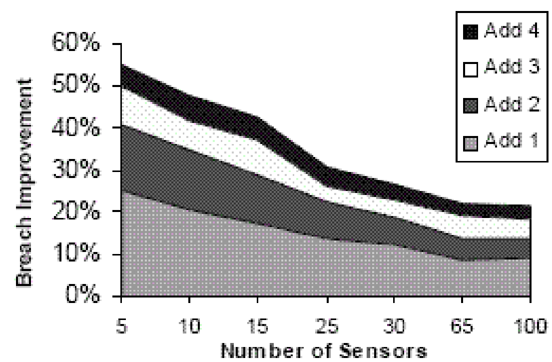

Figure 5: Average breach coverage improvement by additional sensor deployment.

corresponding Delaunay triangulation with introducing only two extra edges to connect I and F to the closest sensors in the structure.

5.2 Sensor Deployment Heuristics

By deploying additional sensor along the edge in the breach path closest to the sensors, one can improve the overall coverage. Similarly, by deploying a sensor node along the edge of the maximum length in the support path, one can improve the support coverage.

Figure 5 shows the average improvement in breach coverage when up to 4 additinal sensors are introduced in the randomly deployed network. The breach path algorithm is performed after each deployment (from 1 to 4), showing the breach region the next sensor to be deployed to. After deployed 100 sensors ran-

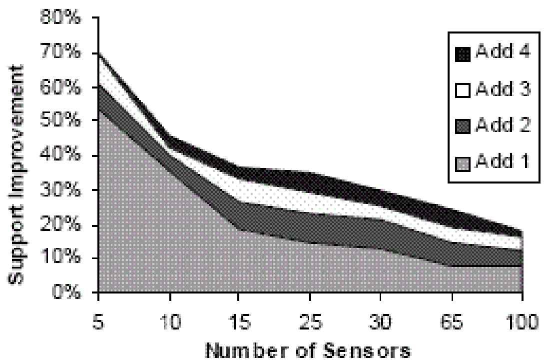

Figure 6: Average support coverage improvement by additional sensor deployment.

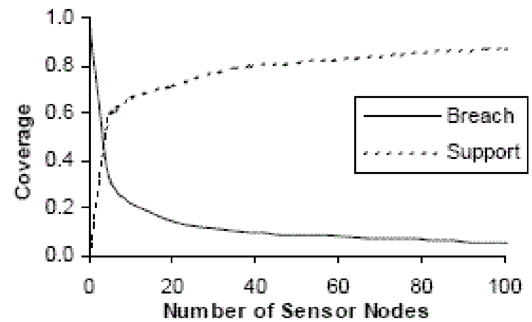

Figure 7: Normalized breach and support coverage as a function of number of sensor nodes

domly, breach coverage can be improved about 10% by deploying one more sensor.

The mid-point of the edge corresponding the *support_weigth* (the edge of maximal length found from the support path), shows the best deployment region while improving the support coverage. Figure 6 shows the average improvement of the coverage of the network while adding one to four sensors based on the support path. Figure 7 shows that 50% improvement can be achieved in support coverage by adding one additional sensor while 5 sensors have already been randomly deployed.

5.3 Asymptotic Behaviour

The graph in Figure 7 shows how the coverage of randomly placed sensor nodes in a field varies as changing the number of sensors. Both, breach and support indicate better coverage of the sensor field. The asymptotic behaviour of the graph suggests that by analyzing a given field and selecting proper number of sensor nodes, certain levels of coverage can be expected even if sensor deployment cannot be performed according to an exact plan.

6 Criticism

The algorithm presented in the article solves the proposal for deploying the new sensor nodes. After performing the worst case algorithm, the next node is deployed along the edge closest to the original nodes. By doing this, the algorithm is forced to find a new path at the next performing round. In best case, the new node is deployed in the middle of the longest edge found at the Maximal Support Path, which similarly forces the algorithm to find a new path. In Ad-hoc networks, the transmissions disturb other transmissions and a message ment to a particular node can be heard by neighbour node, which is not recommended. In my opinion, the new nodes should be deployed for instance the center of the biggest circle at the network area which do not include any sensors. The algorithm deployes a node close to other nodes improving the coverage of the support path or breach path, but do not detect the breaches elsewhere the network.

The usability of the algorithm is strictly restricted because the algorithm can be used only for networks where the locations of the nodes is known beforehand. If the locations are not known, but some beacon nodes can be found, the locations of almost all nodes can be found by trilateration based on the signal strength information. The researchers assume the nodes to be on a flat ground with nothing disturbing the signal, so the strength of the signal can be considered to be dependent only the distance between the nodes. To find a practical example where the algorithm presented in the article could take place is not easy. Researchers leave this problem to the future research.

In general, the presented algorithm has different approach than the practical communication problems. By improving the coverage and presenting the experimental results, the reseachers do not tell whether they have evaluated the coverage from only one region to another or tried to solve the coverage by finding many paths. Finding many paths might give more real overall coverage results, because deploying one sensor could improve just one path but the transversal coverage path might stay the same.

References

[1] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava. Coverage Problems in Wireless Ad-hoc Sensor Networks. IEEE INFOCOM 2001.