

# Coverage in Wireless Sensor Networks: A Survey

Raymond Mulligan

Wireless Sensor and Mobile Ad-hoc Networks (WiSeMAN) Research Lab

Department of Computer Science, Hofstra University

Hempstead, NY 11549, USA

RMULLI1@pride.Hofstra.edu

Habib M. Ammari

Wireless Sensor and Mobile Ad-hoc Networks (WiSeMAN) Research Lab

Department of Computer Science, Hofstra University

Hempstead, NY 11549, USA

Habib.M.Ammari@hofstra.edu

Received: November 24, 2009 Accepted: April 13, 2010 DOI: 10.5296/npa.v2i2.276

## Abstract

Wireless sensor networks are a rapidly growing area for research and commercial development. Wireless sensor networks are used to monitor a given field of interest for changes in the environment. They are very useful for military, environmental, and scientific applications to name a few. One of the most active areas of research in wireless sensor networks is that of coverage. Coverage in wireless sensor networks is usually defined as a measure of how well and for how long the sensors are able to observe the physical space. In this paper, we take a representative survey of the current work that has been done in this area. We define several terms and concepts and then show how they are being utilized in various research works.

**Keywords:** Connectivity, coverage, nodes, wireless sensor networks

## 1. Introduction

In recent years there has been increasing interest in the field of wireless sensor networks. A wireless sensor network consists of a number of wireless sensor nodes. These nodes are characterized by being very small in size with limited energy usually supplied by a battery. They communicate via built-in antennae over RF signals. These networks are typically used to monitor a field of interest to detect movement, temperature changes, precipitation, etc.

One of the most active research fields in wireless sensor networks is that of coverage. Coverage is usually interpreted as how well a sensor network will monitor a field of interest. It can be thought of as a measure of quality of service. Coverage can be measured in different ways depending on the application, we will address this later in the paper.

In addition to coverage it is important for a sensor network to maintain connectivity. Connectivity can be defined as the ability of the sensor nodes to reach the data sink. If there is no available route from a sensor node to the data sink then the data collected by that node can not be processed. Each node has a communication range which defines the area in which another node can be located in order to receive data. This is separate from the sensing range which defines the area a node can observe. The two ranges may be equal but are often different.

The paper is organized as follows: in the next section we address many of the issues that factor into how coverage is determined and how sensor networks are deployed. In section three, we cover several approaches to coverage and discuss how these have been integrated by researchers into their own methods. In section four, similarities and differences are discussed along with criticisms of the various works. Finally we give a summary of the information presented in the paper and discuss possible future research in coverage. At the end of the paper is a list of references.

## 2. Issues in Wireless Sensor Network Coverage

There are several factors that must be considered when developing a plan for coverage in a sensor networks. Many of these will be dependent upon the particular application that is being addressed. The capabilities of the sensor nodes that are being used must also be considered. Most researchers focus on a single deployment model but there are papers that attempt to develop a more general algorithm that can be used in many types of deployment.

### 2.1 Coverage Types

The first step in deploying a wireless sensor network is determining what it is exactly that you are attempting to monitor. Typically you would monitor an entire area, watch a set of targets, or look for a breach among a barrier. Coverage of an entire area otherwise known as full or blanket coverage means that every single point within the field of interest is within the sensing range of at least one sensor node. Ideally you would like to deploy the minimum number of sensor nodes within a field in order to achieve blanket coverage. This problem was addressed in [1] where the author proposes placing the nodes in a construct called an  $r$ -strip such that each sensor is located  $r$  distance away from the neighboring sensor where  $r$  is the radius of the sensing area. The strips can be then placed in an overlapping formation such that blanket coverage is achieved. The biggest problem with this solution is that it is impractical to try to deploy sensors in such a formation.

Target coverage refers to observing a fixed number of targets. This type of coverage has obvious military applications such as those covered in [2]. The authors in this paper did extensive tests to not only detect targets, but to classify and track them. The authors in [3], [4], [5], [6], and [7] attempt to maintain target coverage while conserving energy. The authors in discuss both blanket and target coverage in terms of energy efficiency.

Barrier coverage refers to the detection of movement across a barrier of sensors. This problem was defined as the maximal breach path in [9]. The authors in this study quantify the improvement in coverage when additional sensors are added to a network. Other papers such as [10] focus on algorithms in barrier coverage. A variation of barrier coverage known as sweep coverage is also discussed in [8] and [11]. Sweep coverage can be thought of as a moving barrier problem.

## *2.2 Deployment*

A sensor network deployment can usually be categorized as either a dense deployment or a sparse deployment. A dense deployment has a relatively high number of sensor nodes in the given field of interest while a sparse deployment would have fewer nodes. The dense deployment model is used in situations where it is very important for every event to be detected or when it is important to have multiple sensors cover an area. Sparse deployments may be used when the cost of the sensors make a dense deployment prohibitive or when you want to achieve maximum coverage using the bare minimum number of sensors.

In most of the work studying coverage it is assumed that the sensor nodes are static, they stay in the same place once they are deployed. Newer sensor nodes have the ability to relocate after they are deployed, these are known as mobile nodes. The algorithm in [12] has each sensor node determining the location it needs to move to in order to provide maximum coverage. The authors perform several experiments to determine how well the network covers the area and the deployment time of the algorithm. The key weakness in this algorithm is that each node must be within the sensing range of another node in order to determine the optimal location it needs to move to, if a node is not seen by any other nodes then that node cannot determine its relative location. In the deployment algorithm of [13], each node will communicate with its neighbors and tell them to move away until they are at a distance which maximizes coverage while maintaining connectivity. The simulations run by the authors show a very high degree of coverage can be obtained within minutes of deployment. Actual sensors may not perform as well if they are not able to calculate the distance of their neighbors with the same precision as the nodes in the simulation. The method introduced in [11] aims to maximize coverage while minimizing sensor movement. The simulations run by the authors show the method does achieve excellent coverage with low amounts of movement but it does require a complex algorithm be run which may tax the sensor nodes. The authors in [14] design three separate deployment protocols that provide a high level of coverage with minimal movement in a short time. The simulations show that the protocols hold up with a limited amount of sensors but there are questions about how scalable the protocols are with larger numbers of sensors.

Sensor network nodes are deployed in an area by either placing them in predetermined locations or having the nodes randomly located. Dropping sensors from a plane would be an example of random placement. It is easier to develop a coverage scheme for deterministic placement of sensor nodes than for random placement. However in many deployments, it is either impractical or impossible to deploy sensor nodes in a deterministic way. Examples of deterministic and random placement is shown in figures 1 and 2. The simple construct in [12] is an example of a deterministic placement. A more sophisticated deterministic deployment method is given in [15]. The authors propose to arrange the sensors in a diamond pattern which would correspond with a Voronoi polygon. The pattern achieves four way connectivity from each of the nodes with full coverage when the communication range divided by the sensing range is greater than the square root of two. The authors are able to mathematically prove the validity of their pattern, however the pattern is not practical for actual deployment. It assumes that the sensing and communication ranges of every node are a perfect circle as well as the ability to place the sensors in exact locations. Random deployments of sensor nodes are usually dense deployments as well since it is necessary to deploy additional sensors in order to achieve coverage if the sensor nodes are stationary. Networks with mobile sensors usually start out with a random deployment and utilize the mobility property in

order to relocate to the optimal location. Most research with random deployments of sensor nodes regards the ability to maintain coverage while minimizing the amount of energy expended. This will be covered more closely in another part of the paper.

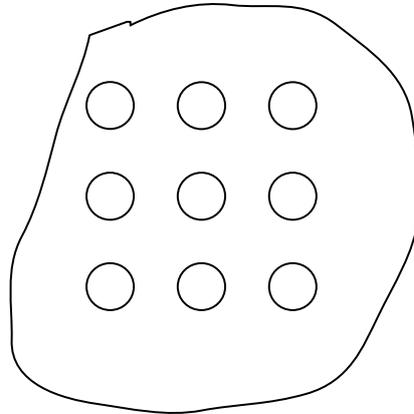


Figure 1: Deterministic Placement

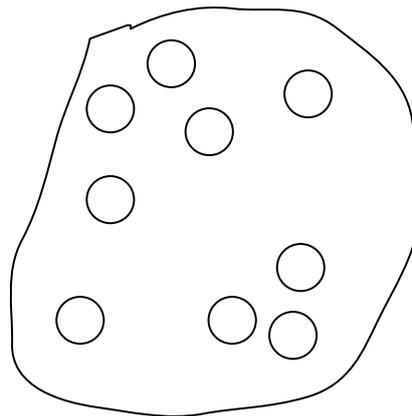


Figure 2: Random Placement

### 2.3 Node Types

The set of nodes that are selected for a sensor network can be either a homogeneous or heterogeneous group of nodes. A homogeneous group is a group in which all of the nodes have the same capabilities. A heterogeneous group is one in which some nodes are more powerful than other nodes. Usually you would have a smaller group of more powerful nodes known as cluster heads which would gather data from the less powerful nodes. Examples of homogeneous and heterogeneous nodes are given in figures 3 and 4.

A homogeneous set of nodes is required for the algorithms in [1] and [15]. Each of these solutions require the nodes to be placed at a precise distance in relation to each other that is dependent on the sensing ranges of every node being identical. The authors in [12] assume homogeneous sensors but repeat their experiments with different different uniform sensing ranges in order to prove the efficacy of their algorithm. Several algorithms for best coverage using homogeneous nodes are presented in [16].

Any algorithm that will work for a heterogeneous network will also work with a homogeneous network. Several papers attempt to prove their theories first with a homogeneous deployment then

show that the findings will hold up for a heterogeneous deployment. In [10] the authors design a rectangular based coverage model using homogeneous sensors to monitor a barrier. The authors do this by assuming a maximum and minimum sensing range and substituting these values into the theorem that was previously proven for homogeneous networks. The authors in [17] build an energy efficient network by using homogeneous sensors. This is then extended for heterogeneous networks. They do this by using a weighted Voronoi diagram.

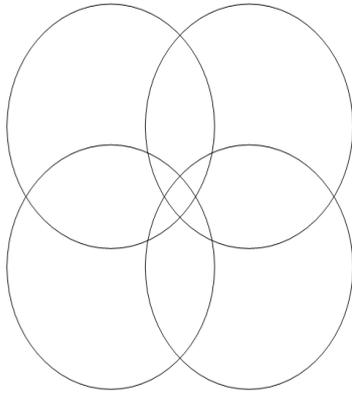


Figure 3: Homogeneous Sensors

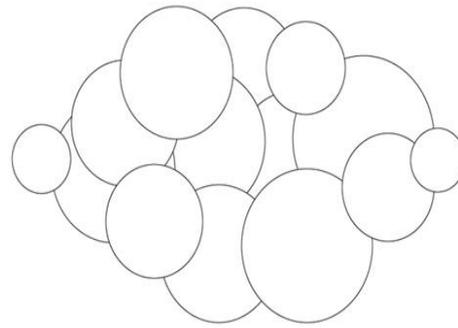


Figure 4: Heterogeneous Sensors

## 2.4 Constraints

Perhaps the most important factor to consider in the development of a coverage scheme is that of energy constraints. Sensor nodes usually depend upon a battery for their energy source and in most deployments battery replacement is not feasible. It therefore becomes very important to conserve energy and prolong battery life. There are several methods available to do this. Placing unneeded sensors into a low energy sleep mode is a popular method to conserve energy. Another method is to adjust the transmission range so that the sensor nodes only use enough energy to transmit to a neighbor node. When sensors are arranged in a hierarchical network then cluster heads can be used to aggregate data and reduce the amount of information sent up to the sink. This will relieve some of the burden on the nodes that are along the transmission path and increase their lifetimes. Improving the efficiency of data gathering and routing is also used to conserve energy. If multiple sensor nodes are collecting the same information the network is expending energy unnecessarily. Eliminating the redundancy will allow the network to be more efficient. Optimizing the routing so that data is sent along the shortest path to the sink using the least number of nodes will conserve energy by lightening the routing burden on some nodes. By using less energy for routing data, coverage is helped by having the nodes' lifetimes extended. There is a great deal of research in the optimization of sensor routing but it is not directly related to the issue of coverage and will not be discussed further in this paper.

Cardei and Wu present a summary of different approaches to energy efficient coverage problems in [8]. The authors state that most work done in this field was in the theoretical realm at the time of the survey. Chen, Kumar, and Lai extend a barrier coverage protocol to improve energy efficiency. When a node detects adequate  $k$ -coverage in the area it will put itself into sleep mode. It will enter wakeup mode after a random period of time and perform another check. If the node is not needed then it will find out from the other nodes when and factor that into its calculation as to when it should wakeup again. The authors in [4] conserve energy by turning off groups of nodes at a time. The authors in [18] introduce a new protocol in which the nodes can be in any of five different states. When a node wakes from the sleep state it will enter the listen state and wait for a beacon. After receiving the beacon the node determines if it should go back to sleep mode or go to the join state. From this state it will wait for its timer to expire and move to the active state unless it receives a message telling it to return to the sleep state. When the node is in the active state it is providing

coverage to the area, it will remain in this state until it becomes ineligible at which point it moves to the withdraw state. Once in the withdraw state the node sets a timer and returns to the sleep state unless it receives a message telling it to return to the active state. The authors in [17] utilize a redundancy protocol in which the redundant sensors with the lowest energy levels are turned off. There may be several rounds of computation and sensors turned off until the optimal configuration is achieved.

A sensor node's coverage area is usually modeled as a disk in two dimensions or a sphere in three dimensions. Any point within the area is assumed to be seen by the node. However in actual deployments the coverage area can be affected by obstacles. Examples of obstacles are walls and office equipment in indoor deployments, rocks and trees in outdoor deployments. Obstacles can absorb or reflect the RF signal put out by the node thereby rendering the area behind them invisible to the node. Examples of how an obstacle would affect one or two sensors are given in figures 5 and 6. The authors in [19] define different types of obstacles to be used in simulation environments. They specify several primitive shape such as circles, rectangles, and stripes that can be combined to simulate a real object in the sensing area. They run several experiments in a simulation environment to study the effects of obstacles on several routing protocols. They do not address the coverage problem in their work but their models or something similar could be used in coverage simulations. The effect of obstacles on coverage and connectivity is discussed by Wang, Hu, and Tseng in [20]. The authors assume a homogeneous set of nodes with a deterministic deployment and attempt to ensure coverage and connectivity with the minimum number of nodes. They divide the field of interest into smaller areas to determine where to deploy the sensors. The authors in [21] also consider forces exerted by obstacles. Knowledge of the terrain is needed to guide the initial random deployment in order to improve coverage.

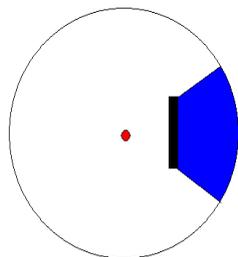


Figure 5: Effect of obstacle on one sensor

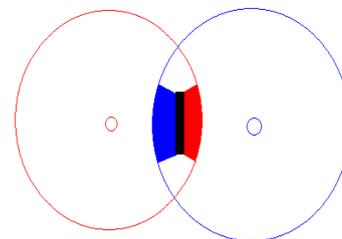


Figure 6: Effect of obstacle on two sensors

Depending on the application an area may require that multiple sensors monitor each point in the field of interest. This constraint is known as  $k$ -coverage in which the  $k$  represents the number of nodes that watch each point. Requiring a  $k$  value of more than one will add complexity to the coverage algorithm. An example of  $k$ -coverage is illustrated in figure 7. The  $k$ -coverage constraint is closely related to the energy constraint in that most of the research that has been performed attempts to preserve  $k$ -coverage while minimizing the energy expended in the sensor nodes. This is the goal of the authors of [22], [23], and [24]. The authors in [25] extend the problem to connected  $k$ -coverage which they define as the minimum number of sensors that must stay active so that each point in the field is  $k$ -covered and every one of these sensors must be connected to each other. Poduri and Sukhatme [13] examine the problem of  $k$ -coverage in a network with mobile sensors. Their algorithm will compel the sensors to move as far away from each other as possible in order to maximize coverage while maintaining  $k$ -coverage in the area.

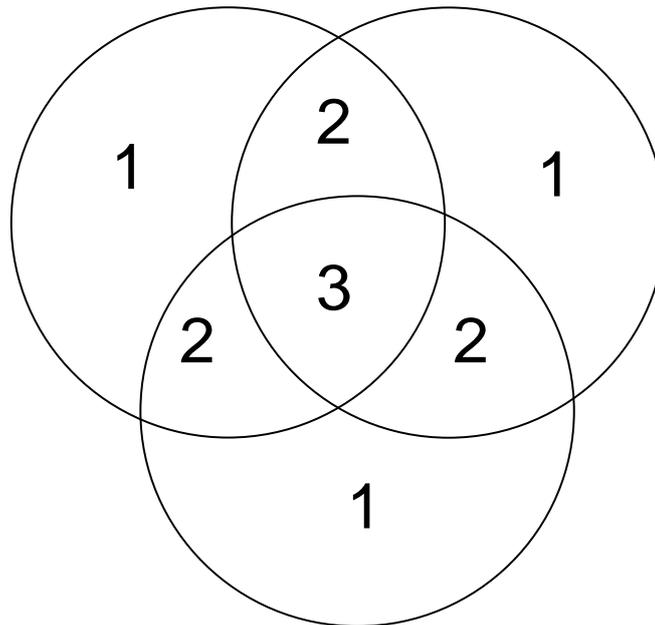


Figure 7: k-coverage

### 2.5 Centralized/Distributed Algorithms

Once sensors are deployed an algorithm is run to determine if sufficient coverage exists in the area. This can be either a centralized or distributed algorithm. A centralized algorithm is run on one or more nodes in a centralized location usually near the data sink. A distributed or localized algorithm is run on nodes throughout the network. Distributed algorithms involve multiple nodes working together to solve a computing problem while localized algorithms imply that many or all of the nodes run the algorithm separately on the information each has gathered. They both spread the workload out more evenly than the centralized algorithm, however since it is being run on many more nodes throughout the network the distributed/localized algorithms may be more complex than the centralized algorithms. Figures 8 and 9 demonstrate centralized and distributed strategies, the shaded sensors are the ones that are running part or all of the algorithm. Zhou, Das, and Gupta develop both centralized and distributed algorithms for connected k-coverage in [23]. They find that both centralized and distributed algorithms return a near optimal solution. The authors in [24] also present centralized and distributed versions of their algorithm. The experiments they run indicate that their algorithm runs faster, activates a near optimal number of sensors and consumes less energy than comparable algorithms.

One example of a centralized method is given by Cardei, Thai, Li, and Wu in [3]. They employ a central data collector node known as a base station to collect data and determine which sensors to deactivate in order to conserve energy and preserve k-coverage. The authors in [4] also use a central data collector node to gather information from the other sensor nodes to decide which sensors to put into sleep mode.

Chen, Kumar, and Lai develop a localized algorithm for barrier coverage in [10]. Although individual sensors cannot confirm the lack of barrier coverage in a network, centralized algorithms for barrier coverage do not scale well. The need for a distributed algorithm is apparent so they introduce the Localized Barrier Coverage Protocol (LBCP) which is intended to maximize the network lifetime by determining when nodes should be put into sleep mode and woken back up.

The authors' simulations show that LBCP provides near optimal performance while providing barrier coverage most of the time. The protocol does break down as the width of the barrier increases. A distributed algorithm for target coverage is given by Zhang, Wang, and Feng in [6]. They propose the Distributed Optimum Coverage Algorithm (DOCA) which is designed to maximize the network lifetime by having the sensors periodically calculate their power to adjust their waiting time. When the waiting time expires they transition to an active state.

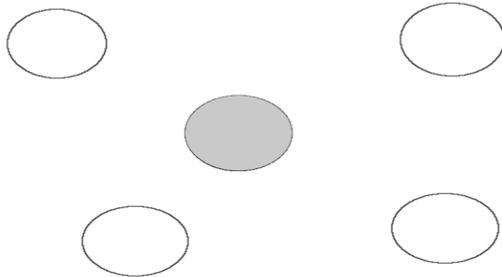


Figure 8: Centralized Algorithm

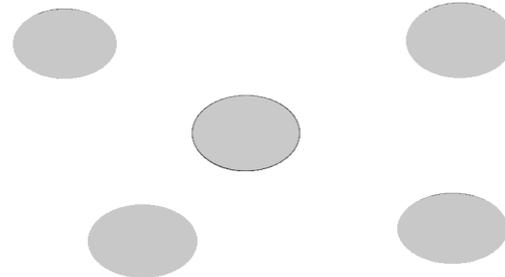


Figure 9: Distributed Algorithm

### 2.6 Three dimensional coverage

Most research that has been done on the coverage problem uses a two dimensional field as its model. It is a lot easier to develop algorithms for a two dimensional area than a three dimensional area but this is not sufficient for many real world environments. An example of three dimensional coverage is given in Figure 10.

One of the first papers on three dimensional coverage was [26] by C. Huang, Y. Tseng, and L. Lo. They assume that the sensors' coverage ranges are shaped in a sphere. They approach the coverage problem not by looking at the coverage of each point in the field of interest, rather they determine whether each sphere is covered. If each of the spheres have sufficient  $k$ -coverage then the entire field must be  $k$ -covered as well. If a sphere has at least  $k$  other spheres covering its entire area then the area within the sphere is  $k+1$  covered. They reason that if the area within the sphere is  $k$ -covered then the area bordering the sphere must be at least  $k$ -covered. An algorithm is then presented which determines whether a sphere is  $k$ -covered or not. This algorithm can be fully distributed so that each sensor can determine its own coverage. The authors state that if one sphere is located entirely within another sphere then those spheres will have no intersection. The smaller sphere would need to have its coverage increased by one but that does not seem to be covered in the algorithm. The algorithm does appear to mathematically feasible but it would appear that calculating the spherical caps which are the intersections of the sphere with decent precision could be very tricky for some sensor nodes. The authors did not run any simulations to test their theory so its implementation is dubious.

S. Alam and Z. Haas took on the problem of three dimensional coverage and connectivity in [27]. They attempted to determine what is the best way to place nodes in a three dimensional space to ensure coverage with the fewest number of nodes. They also wanted to determine the minimum ratio of transmission range to sensing range. They note the similarity between this problem and the sphere packing problem which was how to arrange non-overlapping spheres in a space so that the maximum number of spheres could fit. The Voronoi cell in 3D which is a polyhedron is used as a base model for the sensors' coverage. The ratio of the volume of that area that is actually being covered by the polyhedron to the maximum area that can be covered by a polyhedron is the volumetric quotient of the polyhedron. Finding the highest total value for the polyhedron is the key to the solution. The authors then compute the volumetric quotient using several different types of polyhedrons to show that a truncated octahedron is the best space filling shape. They then calculate the minimum transmission ranges for each polyhedron to show that the cube has the lowest

minimum transmission range while the truncated octahedron has the highest. A simulation is run to validate the findings. The paper succeeds as an excellent academic exercise however the practicality of this strategy appears to be limited. If the nodes are static then it would be very difficult to place them in the precise locations to conform to these shapes. Mobile sensor nodes could conceivably arrange themselves in such a formation but it appears to be a very complex task which may be better served by increasing node density in the area.

The three dimensional spaces that sensor nodes would be deployed in are the atmosphere, underwater, and space. The deployment of sensor nodes underwater is addressed by D. Pompili, T. Melodia, and I. Akyildiz in [28]. They examine the problem first in two dimensions where the sensors are anchored on the ocean floor, then in three dimensions where the sensors float in the water. In addition to finding the minimum number of sensors required for coverage the authors also consider node failures and how many redundant nodes are needed to compensate for node failures. For the two dimensional deployment they propose a triangular grid of equilateral triangles. They calculate the trajectory of sinking objects in order to increase the predictability of deployment. They propose three different deployment strategies for three dimensional coverage: sensors floating randomly in the ocean, sensors arranged randomly on the ocean floor, or sensors arranged in a grid formation at assigned depths. Their findings indicate that the sensors arranged in a grid pattern give the best coverage ratio with a fixed number of sensors. The findings of the authors are not a surprise as a deterministic deployment should have better coverage than any type of random deployment.

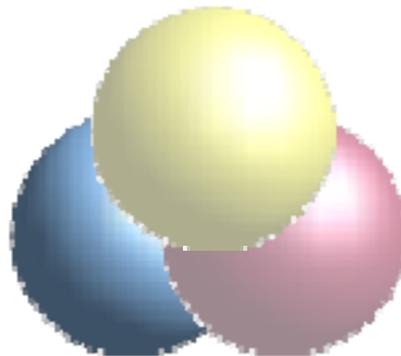


Figure 10: Three Dimensional Coverage

### 3. Approaches to Wireless Sensor Network Coverage

#### 3.1 Art Gallery problem

The art gallery problem is a problem that has been studied in computational geometry and is related to the concept of coverage. In this problem the room is modeled by a polygon and the guards are represented by points in the area. The goal is to ensure that every part of the room can be observed by at least one of the guards. This is similar to the coverage problem so solutions to the art gallery can be used as a base for solving coverage problems.

A more formal definition of the art gallery problem is found in [29]. A point  $x$  is visible by another point  $y$  (guard) if the entire straight line from  $x$  to  $y$  is within the polygon (area). If the entire polygon is visible from any of the guards then the polygon is considered covered. The polygons are classified according to the number of vertices  $n$  each contains. The proof states that  $n/3$  guards are sufficient to cover the polygon. A simple proof of this was given by Steve Fisk in 1978. He partitions the the polygon into triangles colors each vertex of the triangle in one of three colors. The set of any color vertex has visibility over every triangle in the polygon therefore the

color set with the fewest members represents the minimum number of guards needed for coverage.

The proof of the art gallery is only valid for a two dimensional plane. When the problem is extended to a three dimensional space it become NP-hard to determine the minimum number of guards required in order to guarantee coverage.

The authors in [12] note that the art gallery problem can be used as a base for a coverage algorithm only when the shape of the field of interest is known before deployment. It would usually only be used when a deterministic placement of the sensor nodes is being employed.

### 3.2 Voronoi diagram and Delaunay triangulation

The Voronoi diagram has been used a model in several coverage algorithms. The Voronoi diagram for a sensor network is a diagram of boundaries around each sensors such that every point within a sensor's boundary is closer to that sensor than any other sensor in the network. A formal definition of the Voronoi diagram is given in [29]:

Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of points in a plane

A Voronoi region  $V(p_i)$  is the set of points that are as close to  $p_i$  as any other point:

$$V(p_i) = \{x: |p_i - x| \leq |p_j - x| \text{ for all } j \neq i\}$$

An example Voronoi diagram is shown in figure 11. So and Ye present algorithms using generalized Voronoi diagrams to solve two coverage problems in [30]. The first problem is to determine whether a specified level of k-coverage exists in the field of interest. The second problem is to determine the highest level of k-coverage provided by the sensor nodes. They produce an algorithm for the first problem that is proven to run in  $O(n \log n + nk^2)$  time. They then extend the solution to work with heterogeneous nodes for an area requiring coverage of  $k = 1$ . Finally the authors provide a solution for the second problem that will run in  $O(n^3)$  time for two dimensional spaces and  $O(n^4)$  time for three dimensional spaces. The only drawback to the authors' solutions are that they are centralized so they may not scale well for larger networks.

In [11] the authors utilize Voronoi diagrams as part of a fuzzy logic systems used to control the movement of sensors from a random deployment. A fuzzy logic system is one which is designed to take continuous or analog input values and output a discrete or digital value. The input value is analyzed against a set of rules by an inference engine. The engine uses these rules to compute the output. The Voronoi diagram is used to help calculate an ST-FACTOR for the node. The ST-FACTOR is used by the inference engine to determine if the node should move itself. The authors experiments show that the algorithm will provide coverage of 93% or greater after three iterations. One shortcoming of this algorithm is that a node must be able to communicate with other nodes in order to be able to move. If a node on the outer edge of the coverage area had only one neighbor and that neighbor died then the network would not move another node to re-establish communication with the outlying node. Another problem with the algorithm is that it does not provide k-coverage for  $k > 1$ . The lack of redundancy of the deployed network limits its usefulness in areas that are not easily redeployed.

The movement of sensor nodes in random deployments is a problem that is also addressed by Wang, Cao, and La Porta in [14]. The authors use Voronoi diagrams to find gaps in coverage and propose three separate deployment protocols to control the sensor movement. The Vector-based algorithm (VEC) compels two sensors that are too close to each other to move away if there is a coverage hole in either one of their Voronoi polygons. The Voronoi-based algorithm (VOR) will pull a sensor towards a coverage gap. The movement is limited to one half of the communication range in order to avoid communication problems with the neighboring nodes. Finally the Minimax algorithm works similar to VOR except that it further limits the maximum movement of a node. The idea behind this algorithm is to create a move even Voronoi diagram so that the area covered by

each sensor is more uniform. The experiments show that Minimax provides the best coverage with the fewest sensors while VEC performs the worst. Minimax does require the most sensor movement while VEC requires the least. The protocols all perform acceptably for the problem for which they were designed. However, these protocols suffer from the same shortcomings as the fuzzy logic system described previously.

Carbunar, Grama, and Vitek utilize Voronoi diagrams as a means of detecting and eliminating redundancy while preserving coverage in [17]. The authors also tackle the problem of detecting the boundaries of coverage in a sensor network. The authors use what they call a Multiplicative Weighted Voronoi Diagram (MWVD) in which the sites are assigned weights which are used instead of the Euclidean distance in determining how the closest site to a point. The addition of weights allows the Voronoi diagram to be used with heterogeneous sensor nodes. They define the redundant sensor elimination (RSE) solution which selects sensors to deactivate. The simulations run by the authors show that RSE detects all redundant sensors. To find the boundaries of coverage in a sensor network, the sensors whose Voronoi cell is not covered by its sensing range must be found. If there are points in the Voronoi cell not covered by the sensing range of a node that implies there are no other sensors that cover those points. Those points exist in a coverage hole and the sensors that border that coverage hole are boundary sensors. The results of the simulations show that RSE is able to detect all of the sensor boundary nodes in the network. The RSE protocol shows very promising results in the simulations but a true test would be having it implemented in an actual deployment.

The Delaunay triangulation is closely related to the Voronoi diagram. A Delaunay triangulation is defined as a triangulation of an area such no points in any triangle are located within the circumscribed circle of any other triangle in the area. A Delaunay triangulation can be built from a Voronoi diagram simply by drawing edges that connect the sensors which border one another. An example of a Delaunay triangulation is given in Figure 12. The Delaunay triangulation can be used to determine which two sites are closest to each other by finding the shortest edge in the triangle. The Delaunay triangulation is used by the authors in [9] in order to find the maximal support path. Neither the Voronoi diagram nor the Delaunay triangulation can be constructed with localized algorithms. Distributed algorithms for their construction have been found to be inefficient.

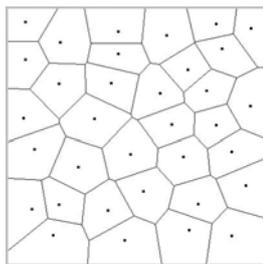


Figure 11: Voronoi diagram

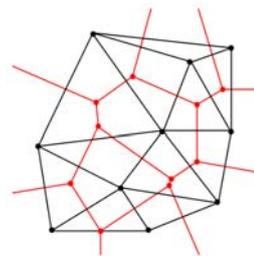


Figure 12: Delaunay triangulation

### 3.3 Worst or Best Case Coverage

Coverage in many sensor network applications can be viewed from either a best case or worst case point of view. When looking at coverage from the best case point of view you are trying to determine the areas high coverage as opposed to the worst case point of view in which you are looking for areas of lower coverage. Looking at coverage from both views is helpful to solving different problems.

The most complete work on worst and best case coverage was provided by Megerian, Koushanfar, Potkonjak, and Srivastava in [31]. The authors describe worst case coverage in terms

of an agent trying to avoid the sensors. The closest distance that an agent must come to any sensor when traveling along the path is considered the worst case. The best case coverage can be described in terms of an agent trying to remain close to the sensors. The furthest distance that an agent must travel from the nearest sensor along the path is considered the best case. The authors employ the Voronoi diagram to determine the worst case or maximal breach path. Since the Voronoi diagram maximizes the distance between the sensors the maximal breach path must lie upon its segments. It is possible to have multiple worst case paths but only one needs to be found for this application. The segments are assigned weights and then binary and breadth-first searches are performed to find the maximal breach path. In order to find the best case or maximal support path the authors employ Delaunay triangulation. To validate this method the authors provide a proof by contradiction to show that the edge with the point farthest away from the sensors lies on the Delaunay triangulation. As in the worst case, weights are assigned to the edges and searches are executed to find the maximal support path. The authors use a simulator to run one hundred random deployments. They find a strong relationship and predictable between the number of nodes deployed and the level of coverage. The authors do a very good job of illustrating their algorithms and perform a sufficient number of tests to validate their results. The biggest problem with the authors' algorithms is that they are centralized. Centralized algorithms would not scale well in a large deployment such as the ones being simulated which contain anywhere from one hundred to one thousand sensors.

The best coverage problem was also covered by Li, Wan, and Frieder in [16]. The authors first define breach and coverage distance as well as the best and worst case coverage. They then present a localized algorithm for best case coverage. Given a set of sensors  $S$ , starting point  $s$ , and ending point  $t$ , a brief summary of the algorithm is as follows:

Find the closest sensor node to  $s$  ( $u_s$ ) and  $t$  ( $u_t$ )

The sensor nodes  $u_s$  and  $u_t$  construct edges ( $uv$ ) to each of their neighbors

Each edge is assigned the weight of  $\frac{1}{2} \|uv\|$

The shortest path algorithm is run to find the path with the minimum weight path connecting  $u_s$  and  $u_t$

The path selected is the smallest weight path connecting  $u_s$  and  $u_t$ , plus the edges from the sensors to those two points

The authors extend the problem to include energy conservation or travel distance in the algorithm. They provide proofs for each of the lemmas used in the construction of the algorithm. The major weakness of this paper is that no simulations or deployments were performed to judge the performance of the algorithms. The algorithms are shown to be theoretically correct but it is unknown how difficult it would be to implement these protocols.

### 3.4 Probabilistic Sensing

Most of the research done in coverage assumes that the sensing ability within a sensing area is deterministic, every point within the sensor's range will be seen by the sensor. This is not always the case with real sensors. A real sensor would be more likely to detect an event that is physically closer to the sensor than one that is near the edge of its sensing range due to attenuation of the RF signal. A probabilistic coverage model takes into consideration the effect of distance on the sensing ability of a node.

Zou and Chakrabarty authored one of the first papers utilizing probabilistic coverage in [21]. Their model is a randomly deployed, heterogeneous cluster based sensor network. The cluster heads run what is called the Virtual Force Algorithm (VFA). VFA determines if two sensors are too close to each other and if they are then it tells them to exert negative forces on each other to push them away until they are at the optimal distance. The simulation run by the authors shows an

improvement in coverage and energy efficiency as opposed to random deployment. The major weakness of this algorithm is related to the cluster head model. If the cluster head fails for any reason then the other nodes may lose connectivity with the network if another cluster head is not within reach. Since this is a homogeneous network the number of cluster heads is limited so if they are not sufficiently spread out in the random deployment then the redundancy of the network may be limited.

Ahmed, Kanhere, and Jha address probabilistic coverage in [32]. The algorithm they present is fully distributed and based on computational geometry. It calculates the perimeter coverage in order to determine the area coverage. Their equation for path loss computes a geometric loss of detection strength that is a factor of the reference distance plus the path loss rate with a random variable added. The rate of probability loss over distance is a smooth downward curve. The algorithm they propose is named the Probabilistic Coverage Algorithm (PCA) and can be briefly summarized as follows:

Each node calculates a sorted list of distances to each of its neighbors

The node detects if it is on a region boundary and if it is it marks points on the perimeter outside the boundary

The probability of a neighbor detecting points with each region is calculated

The cumulative probabilities are computed to see if each area is sufficiently covered

The authors run a simulation that shows PCA returns a more accurate picture of coverage than deterministic models. The major weakness of this article is the computation of probabilistic coverage. Different sensor nodes and different sensor node types may have more or less path loss than the formula given. With the multitude of different sensor nodes and the variation of performance from the same type of sensor node it is unlikely that a precise simulation model can be built for general usage.

Hefeeda and Ahmadi propose a coverage protocol that works for both deterministic and probabilistic sensing models in [33]. They also consider connectivity and energy efficiency while designing the protocol. They call it the Probabilistic Coverage Protocol or PCP. They define an area as having probabilistic coverage if the probability of every point in the area being covered is greater than or equal to a given threshold parameter. The least covered point is defined as the point with the lowest probability of being covered. The PCP protocol will build a triangular structure that ensures that the probability of the least covered point being sensed equal to or greater than the threshold value. It does this by computing the maximum separation possible between sensors and ensuring that the edges do not exceed this value. The authors prove that PCP will generate a connected network. They then prove that the protocol will converge according to a given formula with every point having a probability of being sensed at least equal to the given threshold. The authors run multiple simulations to validate the operation of their protocol.

### *3.5 Disjoint Sets*

Dense sensor deployments usually have more sensors deployed in an area than are needed for the required coverage. In this case the network lifetime can be extended if unneeded sensors can be turned off or sensors can be alternately turned off and on in order to maintain the necessary coverage while conserving battery life of the sensors. One way to accomplish this goal is to divide the sensors into groups or sets. Each set must be capable of covering the field of interest. Several papers have examined this issue.

The authors in [34], [3] and [4] explore energy efficiency for stationary target using disjoint set covers. The disjoint set cover is defined as a subset of the sensors that is capable of covering the entire area by itself. Each set cover is activated and put to sleep in turn in order to preserve the

energy on all the sensors. Slijepcevic and Potkonjak are the first to present a heuristic called the most constrained – minimally constraining heuristic to compute the maximum number of set covers in [34]. Their heuristic attempts to build disjoint set covers using the fewest number of sensors first. The sensors would have to cover larger areas. It then builds succeeding set covers using a greater number of sensors which cover smaller areas. They perform simulations using their heuristic which show that a larger number of set covers being generated. Cardei and Du present another heuristic, MC-MIP, to compute the maximum number of disjoint set covers in [3] and [4]. A simulation is then performed to validate the heuristic and analyze the results against a Slijepcevic and Potkonjak's heuristic. The results indicate that MC-MIP produces a larger number of set covers while executing in an acceptable time. The heuristic is shown to scale well over hundreds of sensors. The major flaw of using disjoint set covers is that it is limited to stationary targets. The set covers are computed only once by a central node so if the targets move then the set covers are invalid.

Cheng, Ruan, and Wu attempt to refine the disjoint set cover approach in [35]. They point out a potential problem with the earlier work is that the number of sensors in a disjoint set is unlimited. When the sensors need to report their data to the base station a bandwidth bottleneck may result. Some of the sensors will not be able to report their data to the base station during the reporting cycle. Assuming the existence of a flat network, the only way to resolve this problem is to factor in bandwidth constraints when creating the sets. They utilize disjoint set covers in order to solve the minimum breach problem. The minimum breach problem is defined as dividing the sensors into disjoint sets so that each set has no more than a given number of sensors while the overall breach does not exceed a certain threshold. The authors develop two heuristics to solve this problem and run simulations to test the relative performances. The simulations show that increasing sensors in a network while keeping the bandwidth constant does not improve coverage. When bandwidth is increased along with the number of sensors the breach rate is decreased. This paper does a good job of identifying a flaw in the earlier work and helping to correct it, however their algorithm would likely lead to fewer disjoint set covers being computed so the sensors would likely be active more often than in the earlier methods.

### *3.6 Coverage with Connectivity*

The areas of coverage and connectivity are closely related. Each are necessary conditions for a functional wireless sensor network. This being the case, attempts have been made to combine the two into a single algorithm. An important principal to consider is that if the communication range of the sensors is at least twice that of the sensing range then coverage of an area implies connectivity. This rule was useful in developing the following protocols.

Xing, Wang, et al. present the Coverage Configuration Protocol in [18]. Their protocol attempts to maximize the number of nodes that can be put into sleep mode while guaranteeing  $k$ -coverage and connectivity. They use a Voronoi diagram to prove the assertion that coverage implies connectivity when  $R_c \geq 2R_s$ . They extend this by proving if the area is  $k$ -covered by the sensors then it is also  $k$ -connected. They also prove that in a convex region, the connectivity is  $2K_s$  if  $R_c \geq 2R_s$ . To prove  $k$ -coverage the authors focus on the boundaries of the sensing range. If every boundary is  $k$ -covered then the entire region must be  $k$ -covered. The nodes in CCP can be in one of three states: SLEEP, LISTEN, or ACTIVE. Each node will periodically send out HELLO packets with its location and status. From this the nodes will compile a list of each of its neighbors when it is in the LISTEN state. If its entire sensing area is covered by its neighbors then it will transition into SLEEP mode. They will remain there until the sleep timer expires and then they will re-evaluate coverage. The CCP protocol does not guarantee connectivity when  $R_c < 2R_s$ . In order to accomplish this the authors decided to integrate CCP with another connectivity protocol SPAN. In this case a node will not enter the ACTIVE state unless it satisfies the eligibility rules for both CCP and SPAN. The ACTIVE node will withdraw if it cannot meet the eligibility rules of either CCP or SPAN. In SPAN a node does not become eligible unless one pair of its neighbors cannot reach each other through no

more than two active nodes. The authors implement three sets of simulations to verify the performance of CCP against other protocols. The simulation results support the authors' assertions about the ability of CCP to provide coverage and connectivity on its own and when combined with SPAN.

Another protocol that attempts to combine coverage and connectivity is the Optimal Geographical Density Control (OGDC) developed by Zhang and Hou in [36]. The authors of this paper are also trying to minimize the number of active nodes in the network. Like the previous work on CCP the authors prove that coverage implies connectivity when  $R_c \geq 2R_s$ . The nodes in OGDC can be in any of three states: ON, OFF, or UNDECIDED. They quantify time into rounds which are comprised of a node selection phase and a steady state phase. The nodes start off as UNDECIDED and then transition to either ON or OFF for the steady state phase. Nodes with greater power will volunteer to be active during the node selection phase. This will lead to more uniform energy depletion among the nodes.

These two protocols are very similar in their design and they approach the same problem. The authors of the OGDC protocol compare their work to CCP and include CCP in their simulations. They find that their protocol requires many fewer active nodes than CCP. They also find that their protocol extends the nodes' lifetimes much better than CCP. Each of the protocols are fully localized so they should scale well. Both protocols however are useful primarily for dense sensor networks. If there is not a high degree of redundancy then the overhead of implementing either of them would not be worth the benefits.

Some recent work in this field was done by Ammari and Giudici in [25]. They approach the problem of connected coverage in sensor networks with heterogeneous sensing ranges. They introduce Helly's theorem to help determine coverage of an area. The theorem states that given a number of convex sets  $n$  in  $d$  dimensional space, if the intersection of every  $d+1$  collection of sets is nonempty then the entire collection of sets must not have an empty intersection. They also utilize the Reuleaux triangle. This is another construct helpful in modeling coverage areas. This is a curve of constant width which is constructed by drawing three circles of equal size with the edge of each circle intersecting with the center of the other two circles. The Reuleaux triangles are tiled to provide the necessary  $k$ -coverage. The authors find a relationship between communication range and sensing range for heterogeneous sensors. Specifically, connectivity is implied if  $R_c \geq \sqrt{3}R_s$ . They define the sensor states as READY, WAITING, and RUNNING. A sensor that is WAITING has its radio turned off and stays in that state for a fixed interval until it switches to the READY state. When a sensor is in READY state is waiting to be woken up by an AWAKE message from a RUNNING node. The RUNNING nodes are able to communicate and sense. They implement both centralized and distributed connected  $k$ -coverage protocols. They run simulations to test the performance of each against existing protocols and each other. The authors plan to extend the work to cover probabilistic sensing areas and three dimensional coverage.

#### 4. Discussion

We have defined a taxonomy to classify the issues and implementations in wireless sensor networks. While this taxonomy is useful it is not perfect as many of the approaches given in the cited research overlap several categories in the taxonomy. In this section, we will look at each of the different research papers and the categories that each of them fall into. We will also discuss the advantages and disadvantages of each.

The Voronoi diagram is the basis for much of the work in coverage. The authors of [30] give a good overview of how the Voronoi diagram may be used to solve coverage problems. They present algorithms for both heterogeneous and homogeneous nodes and various levels of  $k$ -coverage. The authors do not run any simulations to test their algorithms so they are not validated in a deployment.

“Coverage Problems in Wireless Ad-Hoc Sensor Networks” [9] was one of the earliest papers on

the coverage problem. The authors attempt to define the problem from several points of view and present worst and average case algorithms for coverage calculation. They define best and worst case coverage and then use Voronoi diagrams with Delaunay triangulations to develop their algorithms. Simulations are performed to validate the authors' conclusions. The approach the authors used is very useful as a foundation for future work but it is limited by several assumptions made by the authors. The nodes are assumed to be deployed in a deterministic method, the authors claim that the algorithm is valid for randomly deployed nodes but do not prove that in the simulation. The paper also assumes that the nodes are homogeneous but the algorithm is centralized. This is not an optimal solution as some of the nodes would be doing substantially more work than the other nodes. This would lead to a situation in which these nodes drain their energy source more quickly and shorten the network lifetime.

Many of the same authors of [9] collaborated on “Exposure in Wireless Ad-Hoc Sensor Networks” [37]. Exposure is defined as how well a moving object can be observed by the network over a period of time. The major advantage to this paper is that it defines and utilizes a probabilistic sensing model which is much more realistic than the disk based model used in most research. The equation (1) for the sensing model  $S$ , for sensor  $s$  at a point  $p$  is given as:

$$S(s, p) = \frac{\lambda}{[d(s, p)]^K} \quad (1)$$

The constants  $\lambda$  and  $K$  are dependent on the sensors. The Euclidean distance is represented by  $d(s, p)$ . This model is then used to generate the equation (2) for exposure:

$$S(s(0,0), p(x, y)) = \frac{1}{d(s, p)} = \frac{1}{\sqrt{x^2 + y^2}} \quad (2)$$

The algorithm is helpful in helping to calculate detection of an object with probabilistic methods however it is limited in that it addresses a very narrow issue in the field of coverage. It uses the earlier paper's Voronoi diagram as a model and builds upon it.

A self-deployment algorithm is described in [12]. This paper assumes that mobile sensor nodes are to be deployed one at a time with each node using information from the previously deployed nodes to determine its location. The paper uses what it defines as a greedy algorithm, one which seeks to maximize the coverage area for each node. The algorithm given assumes homogeneous nodes but it appears that expanding the algorithm for heterogeneous nodes would not be too difficult. The major problem with this algorithm is that deploying nodes one at a time is impractical in many circumstances. For example if the nodes are to be dropped from a plane or manually dropped by somebody who has to travel a long distance to the field of interest it would be cost prohibitive. If a large number of sensor nodes are to be utilized it would be time prohibitive. The algorithm does not support k-coverage and is therefore not robust. The nodes depend on line of sight so if the nodes around a working sensor node fail then that node is rendered useless. The algorithm is helpful in that it illustrates the advantages of mobility for increasing coverage in sensor networks.

Kar and Banerjee address the issue of optimal node placement in [1]. They deploy sensors in what they call an r-strip. This is a line of homogeneous sensors with sensing radius  $r$  placed such that the distance between two of the sensors is exactly  $r$ . The strips are overlaid to cover an entire area. The algorithm does provide connected coverage with the minimum number of sensors for the field of interest but is impractical for actual deployments. Placing sensors such that they are exactly  $r$  distance apart is usually not possible. Actual sensors do not have exactly the same sensing ranges.

The paper is useful in that it provides a theoretical solution to a narrowly defined problem.

The authors in [16] attempt to develop a localized algorithm for the best case coverage problem defined in [9]. The authors utilize the Voronoi diagram with the Delaunay triangulation in their algorithm. The authors assume a uniform sensing model instead of a probabilistic model. The authors present a mathematical proof of the correctness of their algorithm. The biggest issue with this paper is the lack of a simulation or deployment of the algorithm. That combined with the use of the uniform sensing model make the solution given more suited to the theoretical realm than the practical one.

The issue of  $k$ -coverage is the main focus of Huang and Tseng in [38]. They present algorithms to determine  $k$ -coverage in networks with a disk based sensing model. Algorithms are presented for both homogeneous and heterogeneous sensors. They examine the coverage of the perimeter of the sensing range of each sensor. This is a more efficient approach than attempting to look at each region. They show how the algorithm can be utilized to discover insufficiently covered regions, conserve power by shutting off sensors, and how it can be modified for hot spots which require greater  $k$ -coverage. The authors leave as an open problem the extension of the algorithm for irregular sensing ranges. The algorithm has the advantage of being very efficient in determining  $k$ -coverage. The paper does not include a simulation of the algorithm. The algorithm suffers from not having support for the probabilistic sensor model. The presence of obstacles in the sensing area is not accounted for by the authors.

The authors of “Stochastic Coverage in Heterogeneous Sensor Networks” extend the concept of  $k$ -coverage to randomly deployed networks of nodes with heterogeneous sensing ranges. They use a set intersection model instead of the Voronoi diagram used in many research papers. The authors assume a uniform sensing range instead of a probabilistic range but it may be possible to extend their work to support probabilistic sensing ranges.

Perimeter coverage of nodes with probabilistic sensing ranges is explored in [32]. The authors use a computed variable, path loss, to determine the probability of event detection within the sensor range. They divide the sensing range into a number of concentric circles which each have a predefined probability value. The major problem with this algorithm is that it assumes that the probability of detection decreases almost uniformly over the increasing distance of the sensing range. Actual nodes may exhibit different behavior.

The issue of  $k$ -coverage is further explored in [22]. The approach utilized by the authors in this paper is to partition the sensors into multiple sets. Each set is capable of covering the entire sensing range. The sets are then turned off and on in turns in order to conserve energy in the network. They present randomized, distributed greedy, and centralized greedy algorithms to do this. The simulations show that the randomized algorithm works the best. By performing multiple experiments using several variations in the algorithm the authors present a strong case for their design. The authors of [34], [3] and [4] extend the use of covers by creating an algorithm utilizing disjoint sets. The use of disjoint sets for the covers is important in that it ensures that a sensor is not used by multiple sets that would drain its energy more quickly. They prove that the problem is solvable and present an algorithm that does that. They run simulations to measure the algorithm's performance. The set model for  $k$ -coverage is also utilized by Zhou, Das, and Gupta in [23]. The algorithms presented by the authors not only ensure  $k$ -coverage but connectivity among the nodes. Hefeeda and Bagheri propose an approximation algorithm in [24]. Their algorithm can be implemented in either a centralized or distributed method. The problem with set coverage is that it is useful only in very dense deployments. You need to have a large number of covers for this algorithm to be effective. It is possible that there would be more effective ways to conserve energy when the number of sets in an area is limited to two.

The use of set covers is also illustrated in [5] by Cardei, Wu, Lu, and Pervaiz. They extend the problem by using sensors with adjustable sensing ranges. They focus on covering a set of targets while using the adjustable sensing ranges to create a maximum number of set covers. They present

a number of methods to do this and run simulations to test each of them. The major problem with their approach is that it is completely centralized, all sensor nodes must be able to communicate directly with the base station. This limits the scope of the network severely. Another criticism would be that connectivity is not ensured, you may be able to detect events but have no way of gathering that data.

The authors of [6] and [7] present a distributed algorithm for target coverage. Each node runs the algorithm to determine when its turn to monitor the target comes up. The protocol is successful in extending the lifetimes of the sensor nodes. The nodes have to recalculate their sensing time before each round, this task may be better suited for a cluster node. The protocol can only monitor stationary targets, not moving targets.

In “Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks” [35] the set cover problem is extended to account for bandwidth limitations. A set cover must be able to send its collected data to the base station while it is still active otherwise the data is lost. The authors develop heuristics that limit the number of sensors in each cover in order to ensure that sufficient bandwidth exists for each cover. Their simulations show that bandwidth needs to be increased if the number of sensors in a set cover is increased. The authors do a good job of pointing out a limitations in existing work but do not provide a complete solution for the coverage problem. There are other approaches to the bandwidth constraint problem such as limiting the size of communication packets and aggregating data through cluster heads.

“Constrained Coverage for Mobile Sensor Networks” [13] combines the issues of mobility and k-coverage. The nodes are considered to exert a repulsive force on each other which pushes them away in order to maximize coverage. There is also an attractive force between the nodes which maintains k-coverage in the field of interest. The authors run simulations to test their algorithm and show that the result is symmetrically tiled deployment of sensors. One major advantage of this algorithm is that it can be extended to provide automatic reconfiguration for a failed node. The major problem with the algorithm is that it may be impractical for actual sensors. The ability to precisely find the distances among each other may not be possible for real sensors. To implement these features into wireless sensors may be cost prohibitive at the present time.

The use of mobile sensors is further explored in “Movement-Assisted Sensor Deployment” [14]. The authors utilize Voronoi diagrams to model the sensing network and have the sensors move to fill any coverage hole. They present three deployment strategies and run simulations to compare them. The major problem with this approach is that it only works for k-coverage of one. If a sensor node fails then the other nodes must readjust their positions in order to fill the coverage hole. Events may not be detected while this process is taking place so if node failure is a frequent occurrence in the network the reliability of the network is questionable.

Barrier coverage is the deployment of sensors to detect movement over a predefined border. This application is the focus of [24]. The biggest issue with barrier coverage is that full barrier coverage must be determined from a central location, not by the sensor nodes themselves. The authors here develop a protocol that is distributed and will provide coverage almost all of the time. The simulations run by the authors show that the protocol is valid for thin borders. The protocol is useful if used as described by the authors but can only run in a specific environment.

The issue of barrier coverage with an actual deployment is addressed in [2]. This is a military application in which not only are intrusions detected but classified as well. They use an influence field for classifying intrusions. The target size, speed, and ferro-magnetic content among other factors are used to determine its type and classify it. The authors conducted extensive experiments to test the performance of their solution. The authors give a thorough description of the sensors and software along with the implementation method. The biggest problem is that the classification is limited to predefined categories. Depending on the deployment environment there can be a large number of different objects which enter a monitored zone and distinguishing among them may not be practical or useful for many applications.

The use of a three dimensional coverage area adds a great deal of complexity to coverage algorithms. This was addressed by the authors in [26]. In this paper the authors use a sphere shaped sensing area as the model. They look at the coverage of the spherical cap in order to determine coverage of the area. The authors manage to produce an efficient algorithm but their solution is not without problems. The algorithm assumes a deterministic sensing model. This is problematic in three dimensions just as it is in two dimensions. It also can fail with heterogeneous sensor nodes. If a node is completely within the sensing range of another node without overlap then the algorithm will fail to detect this node as being covered. The algorithm is a good first step in the development of coverage for three dimensional spaces. The issue of three dimensional coverage was also addressed in [27] by Alam and Haas. They use a Voronoi tessellation for optimum coverage and connectivity. It specifies a deterministic placement which may be possible with mobile nodes but which is very difficult with current sensor node technology. Finally, three-dimensional coverage is examined in [28] which uses an ocean model for deployment. The sensors are anchored to the ocean floor and float at a specified depth. This deployment strategy will provide adequate coverage but appears to be cost prohibitive.

Wang, Hu, and Tseng address the issue of obstacle in the sensing range in [20]. Their approach is to partition a field into smaller regions and address coverage in each of these regions separately. Sensors are deployed in rows if there are no obstacles in an area. If an obstacle exists then extra sensors may be placed at the edge of the obstacle to ensure coverage. The simulations run by the author prove that the algorithm will provide coverage in the presence of obstacles. The biggest problem with the authors' approach is that it assumes a deterministic placement of the nodes which may not be practical and is hard to accurately deploy. It also assumes homogeneous sensors with a uniform sensing range. This is not supported by actual sensors.

The issue of connectivity, while not being the focus of this paper is just as important as coverage. The authors of [18] present CCP which attempts to provide both connectivity and coverage. As discussed earlier CCP will guarantee coverage when  $R_c \geq 2R_s$ . The protocol needs to be combined with SPAN if that condition is not met. One limitation of this approach is that it assumes uniform sensing ranges instead of probabilistic sensing ranges. If the  $R_c \geq 2R_s$  condition is not met then you will need to use two protocols instead of one. This adds complexity which may not be necessary since SPAN alone could be used. The authors of [36] also explore the issue of connected coverage and introduce the Optimal Geographical Density Control (OGDC) protocol. This is similar in many respects to CCP but differs in that it attempts to minimize the number of sensors in a set. The authors in [39] develop a distributed algorithm based on connected dominating set. The authors of [17] present a more efficient method of determining redundancy with distributed algorithms using Voronoi tessellations.

The authors of "Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks" [33] introduce the Probabilistic Coverage Protocol (PCP) which provides connected coverage for heterogeneous and homogeneous sensor networks. Their simulations show that energy conservation for PCP is better than CCP or SPAN. The major problem with PCP is that it does not support k-coverage where  $k > 1$ .

## 5. Conclusion

Coverage in a wireless sensor network can be thought of as how well the wireless sensor network is able to monitor a particular field of interest. Ensuring sufficient coverage in a sensor network is essential to obtaining valid data. In this paper we have attempted to give a broad overview of the work that has been done to address the coverage problem in wireless sensor networks. The coverage problem can be approached in many different ways. The needs of a particular deployment will heavily influence the coverage scheme chosen. The hardware and deployment methods that are available and within budget are major factors used when planning how coverage is achieved in the network. The issues faced when designing a coverage protocol include deterministic or random

deployment, heterogeneous or homogeneous sensor nodes, and centralized or distributed algorithms. Many papers focus on a specific problem while others attempt to provide more general solutions that can be used for many deployment types. The research into the coverage problem is ongoing and new work is being published on an ongoing basis. However, there are still many fundamental problems that must be solved before wireless sensor networks can reach their potential. This paper can be used as a starting point or a summary into what has been done so far.

The table below gives a summary of each of the issues discussed in the paper. The columns represent each issue and the rows represent the papers referenced. Each of the referenced works typically address several of the issues that we have discussed in this survey and the table can be used for comparison and for a quick reference. We have tried to cover as many papers as possible to give a current overview of the state of the research into wireless sensor network coverage.

Table 1. Issues discussed in each research paper

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	X	X			X			X													X
2			X			X		X				X									
3		X			X			X	X			X									
4		X				X			X											X	
5		X				X			X			X	X								
6		X							X				X								
7		X							X				X								
8	X	X	X		X	X			X		X	X	X		X	X	X	X		X	X
9			X		X	X		X				X			X	X	X				
10			X		X	X			X			X	X								
11	X		X	X	X	X							X								
12	X			X				X					X		X						
13	X			X		X	X	X			X		X								X
14	X			X									X			X					
15	X				X		X	X								X					X
16							X						X			X	X	X	X		
17	X					X	X		X		X		X			X	X				
18	X							X	X		X		X			X			X	X	X
19										X											
20				X		X		X		X						X					X
21		X		X		X	X		X			X			X				X		
22	X								X		X	X	X							X	
23	X					X			X		X	X	X								X
24	X								X		X	X	X								

25	X					X	X		X		X	X	X						X	X	
26	X					X						X		X							
27	X				X			X					X		X						X
28	X	X		X	X								X								X
29														X	X	X					
30							X	X							X						
31			X		X	X		X				X				X	X	X			X
32		X				X						X							X		
33	X								X		X		X						X		X
34	X					X			X										X	X	
35			X			X			X											X	
36	X					X	X	X	X				X								X
37		X				X													X		
38	X							X			X										
39	X							X	X			X	X								X
40						X	X				X									X	
41	X			X					X		X	X	X						X	X	X
42					X	X					X								X		
43	X							X	X				X							X	X
44	X					X	X		X		X	X	X							X	X

**Legend**

- A - Area coverage
- B - Target coverage
- C - Barrier coverage
- D - Mobile nodes
- E - Deterministic deployment
- F - Random deployment
- G - Heterogeneous nodes
- H - Homogeneous nodes
- I - Energy constraint (power savings)
- J - Obstacle constraint
- K - k-coverage
- L - centralized algorithm
- M - distributed algorithm
- N - three-dimensional coverage
- O - Art Gallery problem
- P - Voronoi diagram
- Q - Delaunay triangulation
- R - Worst or best case coverage
- S - Probabilistic sensing
- T - Disjoint or Connected Dominating sets
- U - Coverage with connectivity

**Acknowledgments**

The authors gratefully acknowledge the insightful comments of the anonymous reviewers which helped improve the quality and presentation of the paper significantly. This work is partially supported by the US National Science Foundation (NSF) grant 0917089 and a New Faculty Start-Up Research Grant from Hofstra College of Liberal Arts and Sciences Dean's Office.

---

**References**

- [1] K. Kar, S. Banerjee, “Node placement for connected coverage in sensor networks”, in: Proceedings of the Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03), Sophia Antipolis, France, 2003.
- [2] A. Arora, P. Dutta, et. al, “A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking, **Computer Networks: The International Journal of Computer and Telecommunications Networking**”, Volume 46 Issue 5, December 2004
- [3] M. Cardei, M. Thai, L. Yingshu, W. Weili, “Energy-efficient target coverage in wireless sensor networks”, INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Volume: 3, On page(s): 1976- 1984 , March 2005.
- [4] M. Cardei and D. Du, “Improving Wireless Sensor Network Lifetime through Power Aware Organization”, *Wireless Networks* , Volume 11 Issue 3, May 2005
- [5] M. Cardei, J. Wu, M. Lu, and M. Pervaiz, “Maximum network lifetime in wireless sensor networks with adjustable sensing ranges”. In Proceedings of the IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob), 2005.
- [6] H. Zhang, H. Wang, and H. Feng, “A Distributed Optimum Algorithm for Target Coverage in Wireless Sensor Networks”, 2009 Asia-Pacific Conference on Information Processing.
- [7] H. Zhang, “Energy-Balance Heuristic Distributed Algorithm for Target Coverage in Wireless Sensor Networks with Adjustable Sensing Ranges”, 2009 Asia-Pacific Conference on Information Processing.
- [8] M. Cardei and J. Wu, “Energy-efficient coverage problems in wireless ad-hoc sensor networks”, *Computer Communications* Volume 29 Issue 4 pp. 413-420, February 2006.
- [9] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Coverage Problems in Wireless Ad-Hoc Sensor Networks”, *IEEE Infocom* 3 (2001) 1380-1387.
- [10] A. Chen, S. Kumar, and T.-H. Lai, “Designing Localized Algorithms for Barrier Coverage,” in *MOBICOM*. ACM, Sep. 2007.
- [11] A. Osmani, M. Dehghan, H. Pourakbar, and P. Emdadi, “Fuzzy-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks”, 2009 First International Conference on Computational Intelligence, Communication Systems and Networks.
- [12] A. Howard, M. J. Mataric, and G. S. Sukhatme, “An incremental self deployment algorithm for mobile sensor networks,” *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, Sep. 2002.

- [13] S. Poduri and G. Sukhatme, “Constrained Coverage for Mobile Sensor Networks”, IEEE International Conference on Robotics and Automation, pages 165-172 April 26-May 1, 2004, New Orleans, LA, USA
- [14] G. Wang, G. Cao, and T. La Porta, “Movement-Assisted Sensor Deployment,” in Proc. of the 23rd IEEE INFOCOM, 2004.
- [15] X. Bai, Z. Yun, D. Xuan, T. Lai, and W. Jia, “Optimal Patterns for Four-Connectivity and Full Coverage in Wireless Sensor Networks”, IEEE Transactions on Mobile Computing, 2008.
- [16] X.-Y. Li, P.-J. Wan, and O. Frieder, “Coverage in Wireless Ad-hoc Sensor Networks”, IEEE Transactions on Computers, Vol 52 (2002), pp 753-763.
- [17] B. Cărbunar, A. Grama, J. Vitek, O. Cărbunar, “Redundancy and coverage detection in sensor networks”, ACM Transactions on Sensor Networks (TOSN), Volume 2, Issue 1, Pages: 94 –128, February 2006.
- [18] Guoliang Xing , Xiaorui Wang , Yuanfang Zhang , Chenyang Lu , Robert Pless , Christopher Gill, “Integrated coverage and connectivity configuration for energy conservation in sensor networks”, ACM Transactions on Sensor Networks (TOSN), v.1 n.1, p.36-72, August 2005
- [19] Chatzigiannakis, G. Mylonas and S. Nikolettseas, “Modeling and Evaluation of the Effect of Obstacles on the Performance of Wireless Sensor Networks”, The 39th Annual ACM/IEEE Simulation Symposium (ANSS 06), Huntsville, USA, April 2006.
- [20] Y. Wang, C. Hu, and Y. Tseng, “Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks”, WICON Proceedings of the First International Conference on Wireless Internet, pp. 114 – 121, 2005.
- [21] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. accepted for publication in ACM Transactions on Embedded Computing Systems, 2003.
- [22] Z. Abrams, A. Goel , S. Plotkin, “Set k-cover algorithms for energy efficient monitoring in wireless sensor networks”, Proceedings of the 3rd international symposium on Information processing in sensor networks, April 26-27, 2004, Berkeley, California, USA
- [23] Z. Zhou, S. Das, and H. Gupta, “Connected k-coverage problem in sensor networks,” in Proc. of International Conference on Computer Communications and Networks (ICCCN’04), Chicago, IL, October 2004.
- [24] M. Hefeeda and M. Bagheri. “Randomized k-coverage algorithms for dense sensor networks”. In Proc. of IEEE INFO-COM 2007 Minisymposium, pages 2376–2380, Anchorage, AK, May 2007.
- [25] H. Ammari and J. Giudici, “On the Connected k-Coverage Problem in Heterogeneous Sensor Nets: The Curse of Randomness and Heterogeneity”, 29th

- [26] C. Huang, Y. Tseng, and L. Lo, “The coverage problem in three-dimensional wireless sensor networks”, Global Telecommunications Conference GLOBECOM '04, November-December 2004.
- [27] S. Alum and Z. Haas, “Coverage and Connectivity in Three-Dimensional Networks”, International Conference on Mobile Computing and Networking, Proceedings of the 12th annual international conference on Mobile computing and networking, Los Angeles, CA, USA. 2006.
- [28] D. Pompili, T. Melodia, and I. Akyildiz, “Deployment Analysis in Underwater Acoustic Wireless Sensor Networks”, International Conference on Mobile Computing and Networking, Proceedings of the 1st ACM international workshop on Underwater networks, Los Angeles, CA, USA, 2006.
- [29] J. O'Rourke, “Computational Geometry in C”, Cambridge University Press, March 25, 1994.
- [30] A. So and Y. Ye, “On solving coverage problems in a wireless sensor network using voronoi diagrams,” in Proc. of Workshop on Internet and Network Economics (WINE'05), Hong Kong, December 2005.
- [31] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Worst and Best-Case Coverage in Sensor Networks,” IEEE Trans. Mobile Computing, vol. 4, no. 1, pp. 84-92, Jan.-Feb. 2005.
- [32] N. Ahmed, S. Kanhere, and S. Jha, “Probabilistic coverage in wireless sensor networks,” in Proc. of IEEE Conference on Local Computer Networks (LCN'05), Sydney, Australia, November 2005, pp. 672–681
- [33] M. Hefeeda, “Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. XX, NO. XX, XXX, 2009.
- [34] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In IEEE Int'l Conf. on Communications (ICC), pages 472–476, 2001.
- [35] M. Cheng, L. Ruan, and W. Wu, “Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks”, IEEE INFO-COM 2005, Mar. 2005.
- [36] H. Zhang and J. C. Hou, “Maintaining Sensing Coverage and Connectivity in Large Sensor Networks”, NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks (Feb. 2004).
- [37] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, “Exposure in Wireless Ad Hoc Sensor Networks”, Procs. of 7th Annual International Conference on Mobile Computing and Networking (2001) 139-150.

- [38] C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” in ACM Int’l Workshop on Wireless Sensor Networks and Applications (WSNA), 2003, pp. 115–121.
- [39] Y. Zou and K. Chakrabarty, “A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks,” IEEE Transactions on Computers, vol. 54, no. 8, pp. 978–991, August 2005.
- [40] L. Lazos and R. Poovendran, “Stochastic Coverage in Heterogeneous Sensor Networks,” ACM Trans. on Sensor Networks, 2, 3, pp. 325-358, 2006.
- [41] J. Chen and X. Koutsoukos. “Survey on coverage problems in wireless ad hoc sensor networks”. In IEEE SouthEastCon, Richmond, VA, March 2007.
- [42] B. Wang, V. Srinivasan, and K. C. Chua, “Information Coverage for Wireless Sensor Networks”, IEEE Communications Letters, Vol. 9, No. 11, November 2005.
- [43] J. Carle, A. Gallais, and D. Simplot-Ryl, “Preserving Area Coverage in Wireless Sensor Networks by Using Surface Coverage Relay Dominating Sets”, Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005).
- [44] S. Yang, F. Dai, M. Cardei, J. Wu, and F. Patterson, “On Connected Multiple Point Coverage in Wireless Sensor Networks”, Journal of Network and Computer Applications, Volume 32 , Issue 4, Pp. 866-877, (July 2009). Final del formulario.