CHAPTER 9

Coverage and Connectivity Issues in Wireless Sensor Networks

AMITABHA GHOSH and SAJAL K. DAS

Department of Computer Science and Engineering, University of Texas at Arlington

9.1 INTRODUCTION

Wireless sensor networks [33,34] have inspired tremendous research interest in since the mid-1990s. Advancement in wireless communication and microelectromechanical systems (MEMSs) have enabled the development of low-cost, low-power, multifunctional, tiny sensor nodes that can sense the environment, perform data processing, and communicate with each other untethered over short distances. A typical wireless sensor network consists of thousands of sensor nodes, deployed either randomly or according to some predefined statistical distribution, over a geographic region of interest. A sensor node by itself has severe resource constraints, such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory; hence it can sense only a limited portion of the environment. However, when a group of sensor nodes collaborate with each other, they can accomplish a much bigger task efficiently. One of the primary advantages of deploying a wireless sensor network is its low deployment cost and freedom from requiring a messy wired communication backbone, which is often infeasible or economically inconvenient.

Wireless sensor networks ensure a wide range of applications [2], starting from security surveillance in military and battlefields, monitoring previously unobserved environmental phenomena, smart homes and offices, improved healthcare, industrial diagnosis, and many more. For instance, a sensor network can be deployed in a remote island for monitoring wildlife habitat and animal behavior [25], or near the crater of a volcano to measure temperature, pressure, and seismic activities. In many of these applications the environment can be hostile where human

Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions
Edited by Rajeev Shorey, Akkihebbal L. Ananda, Mun Choon Chan, and Wei Tsang Ooi
Copyright © 2006 John Wiley & Sons, Inc.

intervention is not possible and hence, the sensor nodes will be deployed randomly or sprinkled from air and will remain unattended for months or years without any battery replacement. Therefore, energy consumption or, in general, resource management is of critical importance to these networks.

Sensor nodes are scattered in a sensing field with varying node densities. Typical node densities might vary from nodes 3 m apart to as high as 20 nodes/m^3. Each node has a sensing radius within which it can sense data, and a communication radius within which it can communicate with another node. (We will discuss the models [52] for sensing and communication later.) Each of these nodes will collect raw data from the environment, do local processing, possibly communicate with each other in an optimal fashion to perform neighborhood data or decision fusion (aggregation) [23], and then route back those aggregated data in a multihop fashion to data sinks, usually called the *basestations*, which link to the outside world via the Internet or satellites. Since an individual node measurement is often erroneous because of several factors, the need for collaborative signal and information processing (CSIP) [49] is critical. Here the assumption is that the more a sensor network has access to the information scattered across different nodes, the greater the likelihood that it would be able to provide more reliable and correct information about the underlying stochastic process.

One important criterion for being able to deploy an efficient sensor network is to find optimal node placement strategies. Deploying nodes in large sensing fields requires efficient topology control [35]. Nodes can either be placed manually at predetermined locations or be dropped from an aircraft. However, since the sensors are randomly scattered in most practical situations, it is difficult to find a random deployment strategy that minimizes cost, reduces computation and communication, is resilient to node failures, and provides a high degree of area coverage [20]. The notion of area coverage can be considered as a measure of the quality of service (QoS) in a sensor network, for it means how well each point in the sensing field is covered by the sensing ranges. Once the nodes are deployed in the sensing field, they form a communication network, which can dynamically change over time, depending on the topology of the geographic region, internode separations, residual battery power, static and moving obstacles, presence of noise, and other factors. The network can be viewed as a communication graph, where sensor nodes act as the vertices and a communication path between any two nodes signifies an edge.

In a multihop sensor network, communication nodes are linked by a wireless medium, which is often unreliable and insecure. These links can be formed by radio, infrared, or optical media. Although infrared communication is license-free, cheap, and robust against interference from electrical devices, it requires line of sight between the sender and the receiver. "Smart dust" [21], which is an autonomous sensing, computing, and communication system based on optical media for transmission, also needs line of sight. Most of the current hardware for internode communication is based on radiofrequency (RF) circuit design in which securing the wireless communication links is of great concern because of the potential malicious users and eavesdroppers who can modify and corrupt data packets, insert rogue packets in the network, or launch denial-of-service (DoS) attacks. Therefore, designing proper authentication protocols and encryption algorithms for sensor networks is very important and a challenging task as well, especially because of severe resource constraints as mentioned earlier.

Routing protocols and node scheduling are two other important aspects of wireless sensor networks because they significantly impact the overall energy dissipation. Routing protocols involve primarily discovery of the best routing paths from source to destination, considering latency, energy consumption, robustness, and cost of communication. Conventional approaches such as flooding and gossiping waste valuable communication and energy resources, sending redundant information throughout the network. In addition, these protocols are neither resource-aware nor resource-adaptive. Challenges lie in designing cost-efficient routing protocols [39,37], which can efficiently disseminate information in a wireless sensor network using resource-adaptive algorithms. On the other hand, node scheduling for optimal power consumption requires identification of redundant nodes [40] in the network, which can be switched off at times of inactivity.

In this chapter, we discuss primarily the node deployment issues that are related to area coverage and network connectivity in wireless sensor networks. In Section 9.2, we introduce the notion of coverage and connectivity and state their importance with respect to different application scenarios. Section 9.3 describes the different models for sensing, communication, coverage, and other functions. We also introduce some mathematical notations and describe a few appropriate mobility models that will be applicable to mobile sensor networks. In Section 9.4 we describe the coverage algorithms based on exposure paths. In Section 9.5 we describe various deployment strategies and compare these strategies with respect to their goals, assumptions, complexities, and usefulness in practical scenarios. Section 9.6 discusses miscellaneous techniques based on node redundancy, which are used to optimize coverage and ensure connectivity. We provide a summary of our work and discuss open research problems and challenges in Section 9.7.

9.2 COVERAGE AND CONNECTIVITY

Optimal resource management and assuring reliable QoS are two of the most fundamental requirements in ad hoc wireless sensor networks. Sensor deployment strategies play a very important role in providing better QoS, which relates to the issue of how well each point in the sensing field is covered. However, due to severe resource constraints and hostile environmental conditions, it is nontrivial to design an efficient deployment strategy that would minimize cost, reduce computation, minimize node-to-node communication, and provide a high degree of area coverage, while at the same time maintaining a globally connected network is nontrivial. Challenges also arise because topological information about a sensing field is rarely available and such information may change over time in the presence of obstacles. Many wireless sensor network applications require one to perform certain functions that can be measured in terms of area coverage. In these applications, it is necessary to define precise measures of efficient coverage that will impact overall system performance.

Historically, three types of coverage have been defined by Gage [12]:

1. *Blanket coverage* — to achieve a static arrangement of sensor nodes that maxi mizes the detection rate of targets appearing in the sensing field
2. *Barrier coverage* — to achieve a static arrangement of sensor nodes that minimizes the probability of undetected penetration through the barrier
3. *Sweep coverage* — to move a number of sensor nodes across a sensing field, such that it addresses a specified balance between maximizing the detection rate and minimizing the number of missed detections per unit area

In this chapter, we will focus mainly on the blanket coverage, where the objective is to deploy sensor nodes in strategic ways such that an optimal area coverage is achieved according to the needs of the underlying applications. Here, it is worth mentioning that the problem of area coverage is related to the traditional *art gallery problem* (AGP) [30] in computational geometry. The AGP seeks to determine the minimum number of cameras that can be placed in a polygonal environment, such that every point in the environment is monitored. Similarly, the coverage problem basically deals with placing a minimum number of nodes, such that every point in the sensing field is optimally covered under the aforementioned resource constraints, presence of obstacles, noise and varying topography.

Before proceeding further, let us introduce the notion of the *degree* of coverage. In the simplest term, the *degree* of coverage at a particular point in the sensing field can be related to the number of sensors whose sensing range cover that point. It has been observed and postulated that different applications would require different degrees of coverage in the sensing field. For instance, a military surveillance application would need a high degree of coverage, because it would want a region to be monitored by multiple nodes simultaneously, such that even if some nodes cease to function, the security of the region will not be compromised, as other nodes will still continue to function, whereas some of the environmental monitoring applications, such as animal habitat monitoring or temperature monitoring inside a building, might require a low degree of coverage. On the other hand, some specific applications might need a framework, where the degree of coverage in a network can be dynamically configured. An example of this kind of application is intruder detection, where restricted regions are usually monitored with a moderate degree of coverage until the threat or act of intrusion is realized or takes place. At this point, the network will need to self-configure and increase the degree of coverage at possible threat locations. A network that has a high degree of coverage will clearly be more resilient to node failures. Thus, the coverage requirements vary across applications and should be kept in mind while developing new deployment strategies.

Along with coverage, the notion of connectivity is equally important in wireless sensor networks. If a sensor network is modeled as a graph with sensor nodes as vertices and the communication link, if it exists, between any two nodes as an edge, then, by a *connected network* we mean that the underlying graph is connected, that is, between any two nodes there exists a single-hop or multihop communication path consisting of consecutive edges in the graph. Similar to the notion of degree of coverage, we shall also introduce the notion of degree of network connectivity. A sensor network is said to have k connectivity or be k-node-connected if removal of any $(k-1)$ nodes does not render the underlying communication graph disconnected. In latter sections, we shall provide formal definitions of k connectivity and k coverage from graph theory perspectives. Like single degree of coverage, single-node connectivity is not sufficient for many sensor network applications because the failure of a single node would render the network disconnected. It should be noted that robustness and throughput of a sensor network are directly related to connectivity.

Area coverage and connectivity in wireless sensor networks are not unrelated problems. Therefore, the goal of an optimal sensor deployment strategy is to have a globally connected network while optimizing coverage at the same time. By optimizing coverage, the deployment strategy would guarantee that optimum area in the sensing field is covered by sensors, as required by the underlying application. By ensuring that the network is connected, it is also ensured that the sensed information is transmitted to other nodes and possibly to a centralized basestation that can make valuable decisions for the application.

9.3 MATHEMATICAL FRAMEWORK

In this section, we introduce the basic mathematical framework for sensing models, communication models, coverage models, mobility models, and graph-theory-based network connectivity models applicable to wireless sensor networks. These will be used in subsequent sections for describing and analyzing the existing algorithms on coverage and connectivity and to provide future research directions.

9.3.1 Sensing Model

Each node has a sensing gradient, whose radius, although ideally extending to infinity, attenuates gradually as the distance increases. The *sensitivity* S of a sensor s_i at point P is usually modeled as follows [26]

$$S(s_i, P) = \frac{\lambda}{[d(s_i, P)]^{\gamma}} \tag{9.1}$$

where λ and K are positive sensor-dependent parameters and $d(s_i, P)$ is the Euclidean distance between the sensor and the point. Typically the value of γ is dependent on environmental parameters and varies between 2 and 5. Since the sensitivity rapidly decreases as the distance increases, we define a maximum sensing range for each sensor. It is customary to assume a binary sensing model, according to which a sensor is able to sense from all the points that lie within its sensing range

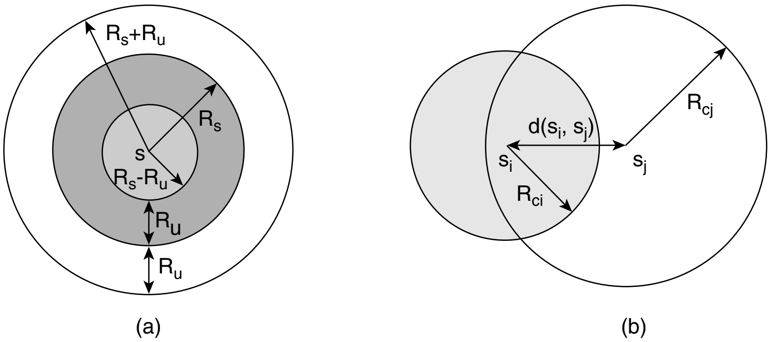

Figure 9.1 (a) Probabilistic sensing model; (b) communication model.

and any point lying beyond it is outside its sensing range. Thus, according to this model the sensing range for each sensor is confined within a circular disk of radius R_s. In a heterogeneous sensor network, the sensing radii of different types of sensors might vary, but in this chapter, to simplify the analysis of coverage algorithms, we assume that all the nodes are homogeneous and the maximum sensing radius for all of them is the same, R_s.

This binary sensing model can be extended to a more realistic one and expressed in probabilistic terms [52]. This is illustrated in Figure 9.1a. Let us define a quantity $R_u < R_s$, such that the probability that a sensor would detect an object at a distance less than or equal to $(R_s - R_u)$ is 1, and at a distance greater than or equal to $(R_s + R_u)$ is 0. In the interval $((R_s - R_u), (R_s + R_u))$, there is a certain probability p, that an object will be detected by the sensor. The quantity R_u is a measure of uncertainty in sensor detection. This probabilistic sensing model reflects the sensing behavior of devices such as infrared and ultrasound sensors.

9.3.2 Communication Model

Similar to the sensing radius, we define a communication radius R_{c_i} (see Fig. 9.1b) for each sensor s_i. Two sensors, s_i and s_j, are able to communicate with each other if the Euclidean distance between them is less than or equal to the minimum of their communication radii, that is, when $d(s_i, s_j) \leq \min\{R_{c_i}, R_{c_j}\}$. This basically means that the sensor with smaller communication radius falls within the communication radius of the other sensor. Two such nodes that are able to communicate with each other are called *one-hop neighbors*. The communication radii might vary depending on the residual battery power (energy) of an individual sensor. In this chapter, we assume that the communication radii for all the nodes are the same, denoted by R_c.

9.3.3 Coverage Model

Depending on the sensing range, an individual node will be able to sense a part of the sensing field. From the probabilistic sensing model, we define the notion of probabilistic coverage [52] of a point $P(x_i, y_i)$ by a sensor s_i by the following equations:

$$c_{x_i y_i}(s_i) = \begin{cases} 0, & R_s + R_u \leq d(s_i, P) \\ e^{-\gamma a^{\beta}}, & R_s - R_u < d(s_i P) < R_s + R_u \\ 1, & R_s - R_u \geq d(s_i P) \end{cases} \tag{9.2}$$

Here, $a = d(s_i, P) - (R_s - R_u)$ and γ and β are parameters that measure the detection probabilities when an object is within a certain distance from the sensor. All points that lie within a distance of $(R_s - R_u)$ from the sensor are said to be 1-covered and all points lying within the interval $((R_s - R_u), (R_s + R_u))$ have a coverage value that exponentially decreases as the distance increases and is less than 1, as observed in Equation (9.2). Beyond the distance $(R_s + R_u)$, all the points have 0 coverage by this sensor. However, a point might be covered by multiple sensors at the same time, each contributing a certain value of coverage. In the following, we define the concept of *total coverage* [52] of a point.

Definition 9.1 (Total Coverage of a Point) Let $S = \{s_i, i = 1, 2, \ldots, k\}$ be the set of nodes whose sensing ranges cover the point $P(x_i, y_i)$. We define the total coverage of the point P as follows:

$$C_{x_i y_i}(S) = 1 - \prod_{i=1}^{k}(1 - c_{x_i y_i}(s_i)) \tag{9.3}$$

Since $c_{x_i y_i}(s_i)$ is the probabilistic coverage of a point as defined in Equation (9.2), the term $(1 - c_{x_i y_i}(s_i))$ is the probability that the point is not covered by sensor s_i. Now, since the probabilistic coverage of a point by one node is independent of another node, the product $\prod_{i=1}^{k}(1 - c_{x_i y_i}(s_i))$ of all such k terms will denote the joint probability that the point is not covered by any of the nodes. Hence, one minus this product would give the probability that point P is covered jointly by its neighboring sensors, and is defined as its total coverage. Clearly, the total coverage of a point lies in the interval [0,1].

9.3.4 Graph-Theoretic Perspective of Wireless Sensor Networks

9.3.4.1 Geometric Random Graph

Over the years, several natural phenomena have been modeled using different graph-theoretic abstractions, more specifically, using random graphs. Understanding the structural properties of such graphs provides valuable insights into the underlying physical phenomena. In this section, we provide some concepts related to graph theory that concern the notion of coverage and connectivity.

Under the particular sensing, communication and coverage models described in the previous sections, the structure of geometric random graphs (GRGs) provides the closest resemblance to wireless sensor networks. A number of probabilistic aspects related to setup of an ad hoc sensor network, such as sprinkling nodes randomly in a sensing field and simultaneous routing of information through different paths, motivates the study of GRGs in the networking community. Furthermore, it has been observed in practice that a sensor network cannot be too dense because of spatial reuse; specifically, when a particular node is transmitting, all other nodes within its transmission radius must remain silent to avoid collision and corruption of data. In this chapter, we consider the generic GRG model $G(n, r, l)$, where, instead of limiting the locations of the graph vertices within a unit square, we assume that the vertices are distributed according to a probability distribution function (pdf) in a d-dimensional space, having a length l for each dimension; and that an edge exists between any two vertices if the Euclidean distance between them is less than the communication radius. In this generic GRG model, the node density n/l^2 can converge to zero, to a constant $c > 0$, or diverge as $l \rightarrow \infty$, depending on the relative values of n, r, and l. Therefore, this model is applicable to both sparse and dense communication networks. Next, we provide a formal definition of GRGs.

Definition 9.2 (Geometric Random Graph) We define a generic *geometric random graph* as $G(n, r, l) = (V, E)$, where a total number n of vertices are distributed according to a pdf f, in a d-dimensional space $[0, l]^d$ to form the nodes in V and an edge $(u, v) \in E$ exists between any two nodes u and v if the distance between them is less than r, namely, $d(u, v) < r$, for some $0 < r \leq l$.

Some of the results from GRG can be applied to study the connectivity in ad hoc wireless networks. For instance, if we assume that a communication graph is induced on a wireless network, then the minimum common transmission range required for all the sensors, such that the communication graph that is connected is equal to the longest Euclidean edge of the minimum spanning tree built on the GRG [31].

These kinds of results from GRGs can be analyzed using the continuum percolation theory [3]. In the theory of continuum percolation, nodes are distributed according to a Poisson density λ. The main result of the theory states that there exists a finite, positive value of λ, say, λ_c, which is called the *critical density*, such that a phase transition occurs in the graph. This means that when the node density crosses a particular threshold λ_c, the detectability of an ad hoc network becomes 1; that is, an object moving within the sensor network can be detected with probability almost equal to 1.

9.3.4.2 Graph Connectivity

In the previous sections, we introduced the concept of degree of coverage and connectivity; here we provide formal definitions for those concepts in terms of node degree and connectivity in a graph [45].

Figure 9.2 A 3-connected graph and a disconnected graph.

Definition 9.3 (Node Degree) Let $G(V, E)$ be an undirected graph. The degree deg(u), of a vertex $u \in V$ is defined as the number of neighbors of u. The minimum node degree of G is defined as $\delta(G) = \min_{\forall u \in G} \{\deg(u)\}$.

Definition 9.4 (*k*-Node Connectivity) A graph is said to be connected if for every pair of nodes, there exists a single-hop or a multihop path connecting them; otherwise the graph is called *disconnected*. A graph is said to be k-connected if for any pair of nodes there are at least k mutually independent (node-disjoint) paths connecting them. In other words, there is no set of $(k - 1)$ nodes, whose removal would render the graph disconnected or result in a trivial graph (single vertex).

Definition 9.5 (*k*-Edge Connectivity) In a similar fashion, the notion of k-edge connectivity is defined when there are at least k edge-disjoint paths between every pair of nodes. In other words, there is no set of $(k - 1)$ edges whose removal will result in a disconnected graph or a trivial graph.

It can be proved [45] that if a graph is k-node-connected, then it is also k-edge-connected, but the reverse is not necessarily true. In this chapter, we shall use the term *connectivity* to mean node connectivity. In Figure 9.2, a 3-connected and a disconnected graph are shown.

Mapping these graph connectivity definitions to the wireless sensor networks scenario, we say that the communication graph formed by the sensor nodes is connected, if between every pair of nodes there exists a single-hop or multihop communication path. A sensor network would be k-connected if at least k other nodes fall within the transmission range R_c of each node. The connectivity problem in sensor networks has been approached from different angles in the literature. One such way is to assign different transmission ranges to the sensors such that the network is connected. This problem has been defined as the *critical transmission range* (CTR) assignment problem [36], which can be formulated for the case of homogeneous sensor network as follows. Given a total number *(N)* of nodes to be deployed in an area A, what is the minimum value of the transmission range to be assigned to *all* the sensors, such that the network ensures global connectivity?

We are now ready to describe various techniques that are used to ensure optimal network coverage and connectivity. In the following sections, we classify these approaches into three main categories and analyze them in terms of their goals, assumptions, algorithm complexities, and practical applicability:

1. Coverage based on exposure paths
2. Coverage based on sensor deployment strategies
3. Miscelleneous strategies

9.4 COVERAGE BASED ON EXPOSURE PATHS

Approaches to solve the coverage problem in wireless sensor networks using exposure paths is basically a combinatorial optimization problem. Two kinds of optimization viewpoints exist in formulating the coverage problem: worst-case and best-case coverage.

In the worst-case coverage, usually the problem is tackled by trying to find a path through the sensing region, such that an object moving along that path will have the least observability by the nodes. Hence, the probability of detecting the moving object would be minimum. Finding such a worst-case path is important because if such a path exists in the sensing field, a user can change the locations of the nodes or add new nodes to increase the coverage and hence observability. Two well-known methods of approaching the worst-case coverage problem are *minimal exposure path* [26] and *maximal breach path* [24,27].

On the other hand, in the best-case coverage, the goal is to find a path that has the highest observability, and hence an object moving along that path will be most probable to be detected by the nodes. Finding such a path can be useful for certain applications, including those that require the best coverage path in regions where security is of highest concern, or those that would like to maximize some predefined benefit function from the nodes while traversing the sensor field. An example of the latter kind is a solar-powered autonomous robot traversing in a light detecting sensor network so as to accumulate the most light within a certain timeframe. By using the best coverage path, the solar powered robot can gain the maximum amount of light within its limited time. Two approaches to solve the best-case coverage problem are *maximal exposure path* [42] and *maximal support path* [27]. In the following text, we describe several methods to calculate the worst-case and best-case coverage paths and the algorithms that use the concept of exposure to derive analytical results.

9.4.1 Minimal Exposure Path: Worst-Case Coverage

Exposure is directly related to the area coverage problem in sensor networks. It is a measure of how well a sensing field is covered with sensors. Informally stated, it can be defined as the *expected average ability* of observing a target moving in the sensing field. The *minimal exposure path* provides valuable information about the worst-case coverage in sensor networks. Let us first explain the notion of *exposure*, which is defined as an integral of a sensing function that is inversely proportional to the distance from the sensors, along a path between two specified points during a certain time interval [28,42]. We can state this formally as follows.

Definition 9.6 (Exposure) The exposure of a moving object in a sensing field during time interval $[t_1, t_2]$ along a path $p(t)$ is defined as the integral:

$$E(p(t), t_1, t_2) = \int_{t_1}^{t_2} I(F, p(t)) \left| \frac{dp(t)}{dt} \right| dt \tag{9.4}$$

where the sensing function $I(F, p(t))$ is a measure of sensitivity at a point on the path by the closest sensor or by all the sensors in the sensing field.

In the first case, it is called the *closest sensor field intensity*, defined as $I_C(F, P(t)) = S(s_{\min}, P)$, where the sensitivity S is given by Equation (9.1) and $s_{\min}$ is the sensor closest to point P. In the latter case, it is called the *all-Sensor field intensity*, defined as $I_A(F, P(t)) = \sum_1^n S(s_i, P)$, where the n active sensors, $s_1, s_2, \ldots, s_n$, contribute a certain value of sensitivity to point P depending on their distance from it. In Equation (9.4), the quantity $|dp(t)/dt|$ is an arc element of the path. If the path is defined in parametric coordinates as $p(t) = (x(t), y(t))$, then

$$|dp(t)/dt| = \sqrt{(dx(t)/dt)^2 + (dy(t)/dt)^2} \tag{9.5}$$

This definition of exposure as given by Equation (9.4) makes it a path-dependent value. Given two endpoints A and B in the sensing field, different paths between them, as shown in Figure 9.3a, are likely to have different exposure values. The

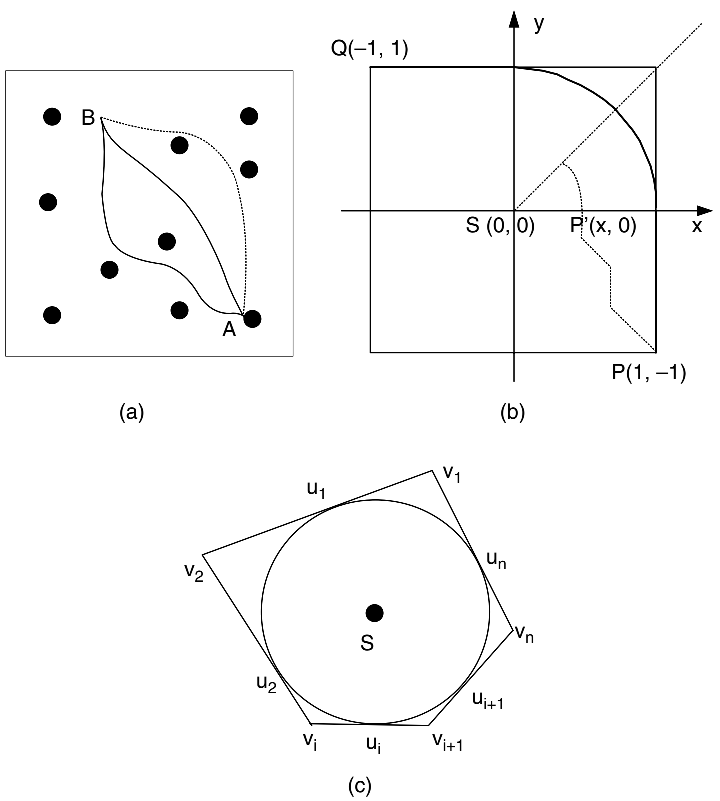

Figure 9.3 (a) Different paths between A and B have different exposures; (b) minimal exposure path for single sensor in a square sensing field; (c) minimal exposure path for single sensor in a sensing field bounded by a convex polygon.

problem of minimal exposure path is to find a path $p(t)$ in the sensing field such that the value of the integral $E(p(t), t_1, t_2)$ is minimum. In the following, we describe a few strategies to calculate the minimal exposure path.

As an example, illustrated in Figure 9.3b, it can be proved [28] that the minimal exposure path between two given points $P(1, -1)$ and $Q(-1, 1)$ in a sensing field, restricted within the region $|x| \leq 1$, $|y| \leq 1$ and having only one sensor located at (0,0), consists of three segments: (1) a straight-line segment from P to (1,0), (2) a quarter-circle from (1,0) to (0,1), and (3) another straight-line segment from (0,1) to Q. The basis of the proof lies in the fact that, since any point on the dotted curve is closer to the sensor than any point lying on the straight-line segment along the edge of the square, the exposure is more in the former case. Also, since the length of the dotted curve is longer than the line segment, the dotted curve would induce more exposure when an object travels along it, given that the time duration is the same in both cases. The calculations show that the exposure along the arc of the quarter-circle in Figure 9.3b is $\pi/2$.

This method can be extended in the following way to more generic scenarios when the sensing region is a convex polygon $v_1, v_2, \ldots, v_n$ and the sensor is located at the center of the inscribed circle, as illustrated in Figure 9.3c. Let us define two curves between points v_i and v_j of the polygon as

$$\Gamma_{ij} = \overline{v_i u_i} \circ \overbrace{u_i u_{i+1}} \circ \overbrace{u_{i+1} u_{i+2}} \circ \cdots \circ \overbrace{u_{j-2} u_{j-1}} \circ \overbrace{u_{j-1} v_j}$$

$$\Gamma'_{ij} = \overline{v_i u_{i-1}} \circ \overbrace{u_{i-1} u_{i-2}} \circ \overbrace{u_{i-2} u_{i-3}} \circ \cdots \circ \overbrace{u_{j+1} u_j} \circ \overbrace{u_j v_j}$$

where $\overline{v_i u_i}$ is the straight-line segment from point u_i to v_i, $\overbrace{u_i u_{i+1}}$ is the arc on the inscribed circle between two consecutive points u_i and u_{i+1}, $\circ$ denotes concatenation, and all $\pm$ operations are modulo n. It can be shown that the minimum exposure path between vertices v_i and v_j is either of the curves Γ_{ij} or Γ'_{ij}, whichever has less exposure.

Next, we extend the preceding two methods of calculating minimum exposure path under the scenario of many sensors. To simplify, the problem can be transformed from the continuous domain into a tractable discrete domain by using an $m \times n$ grid [28]. The minimal exposure path is then restricted to straight-line segments connecting any two consecutive vertices of a grid square. This approach transforms the grid into an edge–weighted graph and computes minimal exposure path using Djikstra's single-source shortest-path algorithm (SSSP) or Floyd-Warshal's all–pair shortest-path algorithm (APSP). The SSSP algorithm complexity is dominated by the grid generation process, which has a time complexity $O(n)$, where n is the total number of gridpoints. On the other hand, the APSP algorithm is dominated by the shortest-path calculation process, which has a time complexity $O(n^3)$.

A different approach based on variational calculus, due to Euler and Lagrange, has been used [42] to find a closed-form expression for minimal exposure path in case of a single sensor. In the following, we state the fundamental theorem of variational calculus and briefly describe the method from the paper by Veltri et al. [42] to derive an analytic solution for minimal exposure path. Informally stated, variational calculus is an approach to solving a class of optimization problems that seek a functional (y) to make some integral function (J) an extreme. The fundamental theorem of variational calculus states the following [11] theorem.

Theorem 9.1 Let $J[y]$ be a function of the form $J[y] = \int_b^a F(x, y, y')dx$ defined on the set of functions $y(x)$, which have continuous first-order derivatives in $[a, b]$ and satisfy the boundary condition $y(a) = A$ and $y(b) = B$. Then a necessary condition for $J[y]$ to have an extremum for a given function $y(x)$ is that $y(x)$ satisfies the Euler–Lagrange equation:

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left(\frac{\partial F}{\partial y'}\right) = 0 \tag{9.6}$$

Assuming the sensitivity of a sensor at a point P as given by $S(s_i, P) = 1/d(s_i, P)$ [$\lambda = 1$ and $\gamma = 1$ in Eq. (9.1)], the minimal exposure path between two arbitrary points A and B can be expressed in the following form using Equation (9.6) in polar coordinates $\rho(\theta) = ae^{\{[\ln(b/a)]/c\}\theta}$, where the constant a is the distance from sensor s_i to A, b is the distance from sensor s_i to B, and c is the angle formed by $\angle ASB$, as shown in Figure 9.4a. The function F in this case is given by $F = (1/\rho)\sqrt{\rho^2 + (d\rho/d\theta)^2}$, after the transformation $x = \rho \cos \theta$ and $y = \rho \sin \theta$.

For the case of multiple sensors, a grid-based approximation algorithm [42] using the Voronoi diagram can be applied. In this approach, the gridpoints are placed along the Voronoi edges and gridpoints that are part of the same Voronoi cell are connected via an edge. The weight of such an edge is determined by the single sensor minimal exposure path weight between the two points. Each node exchanges a set of messages to find topological information and uses it in the localized Voronoi-based approximation algorithm to calculate the minimal exposure path.

In addition to the methods of calculating minimum exposure path, the solution to the *unauthorized traversal* (UT) problem [8] is relevant, which is to find a path P

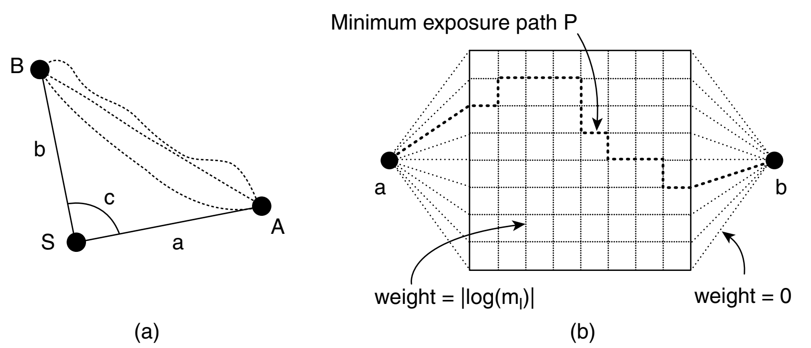

Figure 9.4 (a) Exposure path in single sensor scenario (b) Unauthorized Traversal problem.

that has the least probability of detecting a moving target, given that n sensors are deployed in the sensing field. According to the coverage model described in Section 9.3.3, the probability of failure to detect a target at a point u by a sensor s is $(1 - c_u(s))$. If the decision about a target's presence is taken by a collaborative group of sensors using value fusion or decision fusion, then we can replace $c_u(s)$ by $D(u)$, where $D(u)$ is the probability of consensus target detection using value fusion or decision fusion. Thus, the net probability $G(P)$, of failure to detect a target moving in the path P, is given by

$$G(P) = \prod_{u \in P}(1 - D(u)) \Rightarrow \log G(P) = \sum_{u \in P} \log(1 - D(u)) \tag{9.7}$$

Let us briefly describe the method of calculating a minimal exposure path in the UT algorithm. The algorithm divides the sensor field into a fine grid and assumes that the target moves only along the grid. Then finding the minimum exposure path on this grid is to find a path P that minimizes $|\log G|$.

Consider two consecutive gridpoints, v_1 and v_2. Let m_l denote the probability of failure to detect a target traveling between v_1 and v_2 along the line segment l. Then we have $\log m_l = \sum_{u \in P} \log(1 - D(u))$. Each segment l is assigned a weight $|\log m_l|$, and two fictitious points a, b and line segments with zero weights are added from them to the gridpoints as illustrated in Figure 9.4b. Thus the minimum exposure path in this configuration is to find the least-weight path from a to b, which can be identified using Dijkstra's shortest-path algorithm.

9.4.2 Maximal Exposure Path: Best-Case-Coverage

Earlier, we introduced the notion of maximal exposure path by relating it to the highest observability in a sensing field. In this section, we shall further explain the concept and state a few methods to calculate such a path. A maximal exposure path between two arbitrary points A and B in a sensing field is a path following which the total exposure, as defined by the integral in Equation (9.4), is maximum. It can be interpreted as a path having the best-case coverage. It has been proved [42] that finding the maximal exposure path is NP-hard because it is equivalent to finding the longest path in an undirected weighted graph, which is known to be NP-hard. However, there exist several heuristics to achieve near-optimal solutions under the constraints that the object's speed, pathlength, exposure value, and time required for traversal are bounded. Given these constraints, any valid path that can reach the destination before deadline is contained within an ellipse with the starting and ending points as the foci. This greatly reduces the search space for finding the optimal exposure path. In the following we describe each of the heuristics briefly [42].

1. *Random Path Heuristic*. This is the simplest heuristic to approximately calculate the maximal exposure path. In this method, a random path is created according to the rule that a node on the shortest path from source A to

destination B is selected at certain times, and a random node is selected at other times. Nodes on the shortest path are selected because of the time constraint, and random nodes are selected to collect more exposure. This approach does not depend on the network topology and is computationally inexpensive.

2. *Shortest-Path Heuristic*. In this approach, first a shortest path is calculated between the two endpoints A and B, assuming that certain topographical knowledge is available. Then, to achieve maximal exposure, an object must travel at maximum speed along this path and stop at the point with the highest exposure. However, it might not yield a good approximation because no other path, which might have more exposure, is allowed to be explored.
3. *Best-Point Heuristic*. This heuristic superimposes a grid over the ellipse and then finds the shortest path to each gridpoint from A and B. Next the total exposure of the two paths having a common gridpoint is calculated. The path that gives the maximal exposure is the optimal exposure path. The quality of the optimal path depends on the granularity of the grid; however, this approach is computationally expensive.
4. *Adjusted Best-Point Heuristic*. This method improves the best-point heuristic by considering paths that consist of multiple shortest paths. Performing one or more of the path adjustments such as moving, adding, or deleting a node on the shortest path iteratively, the optimal solution can be found.

9.4.3 Maximal Breach Path: Worst-Case Coverage

In Section 9.4.1, we discussed several methods to find a minimal exposure path in a sensing field under a single-sensor as well as multiple-sensor scenarios. We observed that finding a minimal exposure path is equivalent to finding a worst-case coverage path, which provides valuable information about node deployment density in the sensing field. A concept very similar to finding the worst-case coverage paths is the notion of maximal breach paths [27]. A maximal breach path through a sensing field starting at A and ending at B is a path such that, for any point P on the path, the distance from P to the closest sensor is maximum. The concept of the Voronoi diagram [29], a well-known construct from computational geometry, is used to find a maximal breach path in a sensing field. In two dimensions, the Voronoi diagram of a set of discrete points (also called *sites*) divides the plane into a set of convex polygons, such that all points inside a polygon are closest to only one point. In Figure 9.5a, 10 randomly placed nodes divide the bounded rectangular region into 10 convex polygons, referred to as *Voronoi polygons*. Any two nodes s_i and s_j are called *Voronoi neighbors* of each other if their polygons share a common edge. The edges of a Voronoi polygon for node s_i are the perpendicular bisectors of the lines connecting s_i and its Voronoi neighbors.

Since by construction, the line segments in a Voronoi diagram maximizes the distance from the closest sites, the maximal breach path must lie along the Voronoi edges. If it does not, then any other path that deviates from the Voronoi edges would

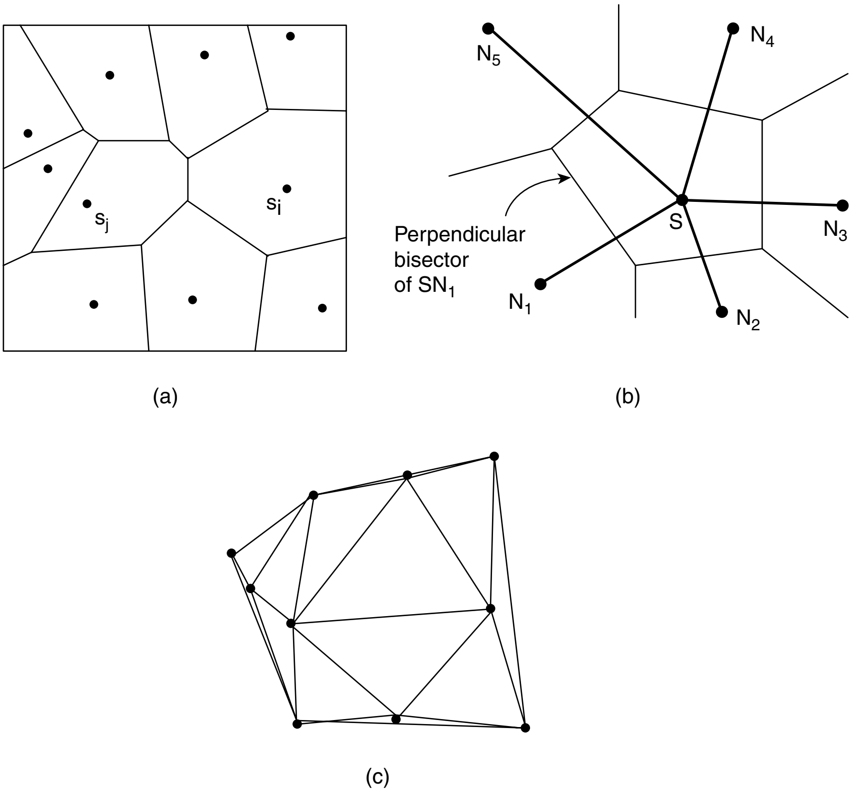

Figure 9.5 (a) Voronoi diagram of 10 randomly deployed nodes; (b) Voronoi polygon for node *S*, constructed by drawing perpendicular bisectors of the lines connecting *S* and its neighbors; (c) Delaunay triangulation for the same set of nodes.

be closer to at least one sensor, thus providing more exposure. Having said that the maximal breach path between two endpoints *A* and *B* will lie along the Voronoi edges, we now describe an algorithm that finds such a path. First a geolocation-based approach is used to determine node locations, and a Voronoi diagram based on that information is constructed. Then a weighted, undirected graph *G* is constructed by creating a node for each vertex and an edge corresponding to each line segment in the Voronoi diagram. Each edge is given a weight equal to the minimum distance from the closest sensor. The algorithm then checks the existence of a path from *A* to *B* using breadth-first search (BFS) and then uses binary search between the smallest and largest edge weights in *G* to find the maximal breach path. It should be noted that the maximal breach path is not unique. It can be proved that the worst-case time complexity of the algorithm is given by $O(n^2 \log n)$, and for sparse networks it is $O(n \log n)$.

Furthermore, the maximal breach path algorithm finds a path such that at any given time, the exposure is no more than some particular value that it tries to minimize. On the other hand, the minimal exposure path does not focus on exposure at one particular time, but rather tries to minimize the exposure acquired throughout the entire time interval in the network.

9.4.4 Maximal Support Path: Best-Case Coverage

A maximal support path through a sensing field starting at A and ending at B is a path such that for any point P on that path, the distance from P to the closest sensor is minimized. This is similar to the concept of maximal exposure path. However, the difference lies in the fact that a maximal support path algorithm finds a path at any given time instant, such that the exposure on the path is no less than some particular value that should be maximized. In contrast, the maximal exposure path does not focus on any particular time; rather, it considers all the time spent during an object's traversal.

A maximal support path in a sensing field can be found by replacing the Voronoi diagram by its dual, Delaunay triangulation as shown in Figure 9.5b, where the edges of the underlying graph are assigned weights equal to the length of the corresponding line segments in the Delaunay triangulation. (A *Delaunay triangulation* [29] is a triangulation of graph vertices such that the circumcircle of each Delaunay triangle does not contain any other vertices.) Similar to the maximal breach path approach described earlier, this algorithm also checks for the existence of a path using breadth-first search and applies binary search to find the maximal support path. The worst-case and average-case complexities for this algorithm are $O(n^2 \log n)$ and $O(n \log n)$, respectively.

So far we have described several methods to derive worst-case and best-case coverage paths exploiting the concept of exposure to detect targets in a sensing field. Now we will see that exposure paths can also be used to find the optimal number of sensors (critical node density) required for complete coverage with very high target detectability [1]. Since the sensing task is inherently probabilistic, the method for critical density calculation takes into account the nature and characteristics of both the sensor and the target. We consider the path-based exposure model as described in Equation (9.4) and that the target moves in a straight line with constant speed away from the sensor at a distance δ. Assuming the probabilistic sensing model as described in Section 9.3.1, typical values are calculated for the quantities $(R_s - R_u)$ and $(R_s + R_u)$, which are termed as *radius of complete influence* (denoted by R_{ci}) and *radius of no influence* (denoted by R_{ni}), respectively. It can be proved that for a typical threshold exposure E_{th}, the values for radius of complete influence and no influence are given by the following equations [1]

$$E_{\mathrm{th}} = \frac{\lambda}{vR_{ci}}\left(\frac{\delta}{\delta + R_{ci}}\right) \tag{9.8}$$

$$E_{\mathrm{th}} = \frac{2\lambda}{vR_{ni}} tan^{-1}\left(\frac{\delta}{2R_{ni}}\right) \tag{9.9}$$

and that to cover an area A with random deployment, the number of nodes required is of the order of $O(A/R_{ni}^2)$.

9.5 COVERAGE BASED ON SENSOR DEPLOYMENT STRATEGIES

The second approach to the coverage problem is to seek sensor deployment strategies that would maximize coverage as well as maintain a globally connected network graph. Several deployment strategies have been studied for achieving an optimal sensor network architecture that would minimize cost, provide high sensing coverage, be resilient to random node failures, and so on. In certain applications, the locations of the nodes can be predetermined and hence can be hand-placed or deployed using mobile robots, while in other cases we need to resort to random deployment methods, such as sprinkling nodes from an aircraft. However, random placement does not guarantee full coverage because it is stochastic in nature, hence often resulting in accumulation of nodes at certain areas in the sensing field but leaving other areas deprived of nodes. Keeping this in mind, some of the deployment algorithms try to find new optimal sensor locations after an initial random placement and move the sensors to those locations, achieving maximum coverage. These algorithms are applicable to only mobile sensor networks. Research has also been conducted in mixed-sensor networks, where some of the nodes are mobile and some are static; and approaches are also proposed to detect coverage holes after an initial deployment and to try to heal or eliminate those holes by moving sensors. It should be noted that an optimal deployment strategy should not only result in a configuration that would provide sufficient coverage but also satisfy certain constraints such as node connectivity and network connectivity [32].

As mentioned in the introduction, the problem of sensor deployment is related to the traditional *art gallery problem* (AGP) [30] in computational geometry. The AGP seeks to determine the minimum number of cameras that can be placed in a polygonal environment, such that the entire environment is monitored. In a similar way, an optimal deployment strategy tries to deploy nodes at optimal locations, such that the area covered by the sensors is maximized. In the following, we briefly describe several sensor deployment algorithms targeted for static, mobile, and mixed-sensor networks, that aim to provide optimum sensing field architecture.

9.5.1 Imprecise Detections Algorithm (IDA)

Dhilon et al. [9] propose a grid coverage algorithm that ensures that every gridpoint is covered with a minimum confidence level. They consider a *minimalistic* view of a sensor network by deploying a minimum number of sensors on a grid that would transmit a minimum amount of data. The model assigns two probability values p_{ij} and p_{ji} for every pair of gridpoints (i,j), where p_{ij} is the probability that a target at gridpoint j is detected by a sensor at gridpoint i and p_{ji} is the probability that a target at gridpoint i is detected by a sensor at gridpoint j. In absence of obstacles, these values are symmetric: $p_{ij} = p_{ji}$. From this, a miss probability matrix M is generated where $m_{ij} = (1 - p_{ij})$. The obstacles are modeled as static objects, and the value of p_{ij} is set to zero if an obstacle appears in the line of sight between two gridpoints (i,j) (as illustrated in Fig. 9.6).

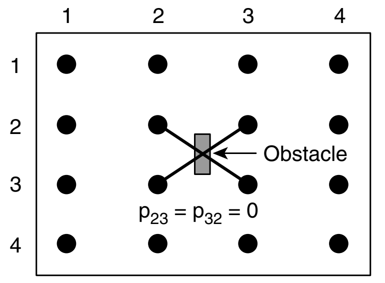

Figure 9.6 Two probability values, p_{ij} and p_{ji}, are assigned for every pair of gridpoints (i, j) under line-of-sight static obstacle modeling.

The algorithm as described by Dhilon et al. [9] takes three inputs: (1) *M, M**, $M_{\min}$, where M is the miss probability matrix as mentioned above; (2) $M^* = (M_1, M_2, \ldots, M_N)$, such that M_i is the probability that a gridpoint i is not *collectively*[1] covered by the set of sensors; and (3) $M_{\min} = 1 - T$, which is the maximum value of the miss probability that is permitted for any gridpoint. The algorithm is iterative and uses a greedy heuristic to determine the best placement of one sensor at a time. It terminates when either a preset upper limit on the number of sensors is reached or sufficient coverage of the gridpoints is achieved.

The time complexity of the algorithm is $O(n^2)$, where n is the total number of gridpoints in the sensor field. It attempts to evaluate the global impact of an additional sensor by summing up the changes in the miss probabilities for the individual gridpoints. However, the algorithm models the obstacles depending on whether they appear in the line of sight of the target and the sensor, which is applicable for infrared cameras, for example, but not for sensors that do not require line of sight, such as acoustic and temperature sensors. Also, since a complete knowledge of the terrain is assumed, the algorithm is not very applicable in cluttered environments, such as interior of buildings, because modeling obstacles becomes extremely difficult in those scenarios.

9.5.2 Potential Field Algorithm (PFA)

In contrast to static sensor networks, nodes in mobile sensor networks are capable of moving in the sensing field. Such networks are capable of self-deployment starting from an initial configuration. The nodes would spread out such that coverage in the sensing field is maximized while maintaining network connectivity. A potential field-based deployment approach using mobile autonomous robots has been proposed to maximize the area coverage. [18,32]. Poduri and Sukhatme augment the scheme such that each node has at least K neighbors. The potential field technique using mobile robots was first introduced in 1986 [22]. In the following we describe the concept of potential field and the algorithm proposed by Poduri and Sukhatme [32].

[1]The notion of collective or total sensor coverage of a point is expressed in Equation (9.3)

The basic concept of potential field is that each node is subjected to a force $\mathbf{F}$ (a vector)[2] that is the gradient of a scalar potential field U; that is, $\mathbf{F} = -\nabla U$. Each node is subjected to two kinds of force: (1) $\mathbf{F}_{\text{cover}}$, which causes the nodes to repel each other to increase their coverage; and (2) $\mathbf{F}_{\text{degree}}$, which constrains the degree of nodes by making them attract toward each other when they are on the verge of being disconnected. The forces are modeled as being inversely proportional to the square of the distance between a pair of nodes and obey the following two boundary conditions:

1. $\|\mathbf{F}_{\text{cover}}\|$ tends to infinity when the distance between two nodes approaches zero to avoid collision.
2. $\|\mathbf{F}_{\text{degree}}\|$ tends to infinity when the distance between critical neighbors approaches R_c, the communication radius.

In mathematical terms, if $\|\text{X}_i - \text{X}_j\| = \Delta x_{ij}$ is the Euclidean distance between two nodes, i and j, then $\mathbf{F}_{\text{cover}}(i,j)$ and $\mathbf{F}_{\text{degree}}(i,j)$ can be expressed as

$$\mathbf{F}_{\text{cover}}(i,j) = \frac{-K_{\text{cover}}}{\Delta x_{ij}^2}\left(\frac{\text{x}_\text{i} - \text{x}_\text{j}}{\Delta x_{ij}}\right) \tag{9.10}$$

$$\mathbf{F}_{\text{cover}}(i,j) = \begin{cases} \dfrac{-K_{\text{degree}}}{\left(\Delta x_{ij} - R_c\right)^2}\left(\dfrac{\text{x}_\text{i} - \text{x}_\text{j}}{\Delta x_{ij}}\right), & \text{for critical connection} \\ 0, & \text{otherwise} \end{cases} \tag{9.11}$$

In the initial configuration all the nodes are accumulated in one place, and thus each node has more than K neighbors, assuming that the total number of nodes is $\geq K$. Then, they start repelling each other using F_{cover} until there are only K neighbors left, at which point the connections reach a critical level, and none of these connections should be broken at a later point of time to ensure K connectivity. Each node continues to repel all its neighbors using $\mathbf{F}_{\text{cover}}$, but as the distance between the node and its critical neighbors increases, $\|\mathbf{F}_{\text{cover}}\|$ decreases and $\|\mathbf{F}_{\text{degree}}\|$ also increases. Finally, at some distance ηR_c, where $0 < \eta < 1$, the net force $\|\mathbf{F}_{\text{cover}} + \mathbf{F}_{\text{degree}}\|$ becomes 0, at which point each node and its neighbors reach an equilibrium and the sensing field becomes uniformly covered with nodes. At a latter point, if a new node joins the network or an existing node ceases to function, the nodes will need to reconfigure to satisfy the equilibrium criteria.

9.5.3 Virtual Force Algorithm (VFA)

Similar to the potential field approach as described in by Poduri and Sukhatme [32], a sensor deployment algorithm based on virtual forces has been proposed [50,52] to increase the coverage after an initial random deployment. Since a random placement does not guarantee effective coverage, an approach that modifies the sensor

[2] The bold symbol $\mathbf{X}$ represents a vector, and $\|\mathbf{X}\|$ represents the magnitude of the vector.

locations after a random placement is useful. In this section, we describe the virtual force algorithm (VFA) briefly.

A sensor is subjected to three kinds of force, which are either attractive or repulsive in nature. In the VFA model, obstacles exert repulsive forces $(\mathbf{F}_{iR})$, areas of preferential coverage (sensitive areas where a high degree of coverage is required) exert attractive forces $(\mathbf{F}_{iA})$, and other sensors exert attractive or repulsive forces $(\mathbf{F}_{ij})$, depending on the distance and orientation. A threshold distance d_{th} is defined between two sensors to control how close they can approach each other. Likewise, a threshold coverage c_{th} is defined for all gridpoints such that the probability that a target at any given gridpoint is reported as being detected is greater than this threshold value. The coverage model as described in this algorithm is given by Equations (9.2) and (9.3). The net force on a sensor s_i is the vector sum of all three forces:

$$\mathbf{F}_i = \sum_{j=1, j\neq i}^{k} \left(\mathbf{F}_{ij}\right) + \mathbf{F}_{iR} + \mathbf{F}_{iA} \tag{9.12}$$

The term $\mathbf{F}_{ij}$ can be expressed in polar coordinates with magnitude and orientation as

$$\mathbf{F}_{ij} = \begin{cases} \left(w_A\left(d_{ij} - d_{th}\right), \alpha_{ij}\right), & \text{if } d_{ij} > d_{\text{th}} \\ 0, & \text{if } d_{ij} = d_{\text{th}} \\ \left(w_R/d_{ij}, \alpha_{ij} + \pi\right), & \text{otherwise} \end{cases} \tag{9.13}$$

where d_{ij} is the distance between sensors s_i and s_j, α_{ij} is the orientation of the line segment from s_i to s_j, w_A and w_R are measures of attractive and repulsive forces, respectively. The VFA algorithm is a centralized one, and it executes in a cluster head. After the nodes are randomly placed in the sensing field, for all gridpoints, the algorithm calculates the total coverage as defined by Equation (9.3). Then it calculates the virtual forces exerted on a sensor s_i by all other sensors, obstacles, and preferential coverage area, for all i. Next, depending on the net forces, new locations are calculated by the cluster head and sent to the sensor nodes, which perform a one-time movement to the designated positions.

For an $n \times m$ grid with a total number of k sensors deployed, the computational complexity of the VFA algorithm is $O(nmk)$. The efficiency of the algorithm depends on the values of the quantities w_A and w_R. Negligible computation time and a one-time repositioning of sensors are two of its primary advantages. However, the algorithm does not provide any route plan for repositioning the sensors to avoid collision.

9.5.4 Distributed Self-Spreading Algorithm (DSSA)

Along the lines of potential field and virtual force based approaches, a distributed self-deployment algorithm (DSSA) has been proposed [16] for mobile sensor networks that maximizes coverage and maintains uniformity of node distribution. They define *coverage* as the ratio of the union of covered areas of each node to the complete area of the sensing field and *uniformity* as the average of local standard deviations of internodal distances. In uniformly distributed networks, internodal distances are almost the same and hence the energy consumption is uniform. DSSA assumes that the initial deployment is random and that each node knows its location. Similar to VFA, it uses the concept of electric force that depends on the internode separation distance and local current density (μ_{curr}). In the beginning of the algorithm, the initial density for each node is equal to the number of its neighbors. The algorithm defines the notion of expected density as the average number of nodes required to cover the entire area when the nodes are deployed uniformly. It is given by $\mu(R_c) = (N\pi R_c^2)/A$, where N is the number of sensors and R_c is the communication range. DSSA executes in steps and models the force on the ith node by the jth node at timestep n as

$$f_n^{i,j} = \frac{\mu_{\text{curr}}}{\mu^2(R_c)} \left(R_c - \left|p_n^i - p_n^j\right|\right)\left(p_n^i - p_n^j\right)/\left(\left|p_n^i - p_n^j\right|\right) \tag{9.14}$$

where p_n^i denotes the location of ith node at timestep n. Depending on the net forces from the neighborhood, a node can decide on its next movement location. The algorithm settles down when a node moves an infinitely small distance over a period of time or when it moves back and forth between two same locations.

9.5.5 VEC, VOR and MiniMax Algorithms

Wang et al. [44], describe three distributed self-deployment algorithms (VEC, VOR, and min–max) for mobile sensors using Voronoi diagrams. After the sensors are deployed in the field, the algorithm locates coverage holes (area not covered by any sensor) and calculates new positions that would increase coverage by moving sensors from densely populated regions to sparsely ones. The Voronoi diagram, as explained in Section 9.4.3, consists of Voronoi polygons such that all the points inside a polygon are closest to the sensor that lies within the polygon, as illustrated in Figure 9.7a. Once the Voronoi polygons are constructed, each sensor within the

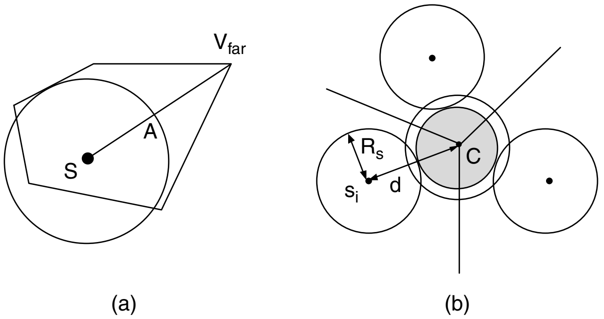

Figure 9.7 (a) The VOR algorithm moves a sensor toward the farthest Voronoi vertex, V_{far}; (b) bid estimated by sensor S_i is the area of the shaded circle with center at C.

polygon examines the existence of possible coverage holes. If such a hole is discovered, the sensors move to new positions according to certain heuristics to reduce or eliminate the coverage hole. In the following we explain the heuristics.

The vector-based algorithm (VEC) pushes sensors from densely covered areas to sparsely covered areas. Two sensors exert a repulsive force when they are too close to each other. If d_{av} is the average distance between any two sensors when they are evenly distributed in the sensing field, the virtual force between the sensors s_i and s_j will move each of them $(d_{\text{av}} - d(s_i, s_j))/2$ distance away from each other. In case, one of the sensor's sensing range completely covers its Voronoi polygon, only the other sensor should move away $(d_{\text{av}} - d(s_i, s_j))$ distance. In addition to the mutual repulsive forces between sensors, the boundaries also exert forces to push sensors too close to the boundary inside. If $d_b(s_i)$ is the distance of a sensor s_i from its closest boundary, then the repulsive force would move it a distance $d_{\text{av}}/2 - d_b s_i$ toward the inside of the region. Before actually moving to the new position, each sensor calculates whether its movement would increase the local coverage within its Voronoi polygon. If not, the sensor wouldn't move to the target location; instead it applies a *movement adjustment scheme* and will move to the midpoint position between its target location and new location.

The Voronoi-based algorithm (VOR) is a greedy algorithm that pulls sensors toward their local maximum coverage holes. If a sensor detects a coverage hole within its Voronoi polygon, it will move toward its farthest Voronoi vertex (V_{far}), such that the distance from its new location (A) to (V_{far}) is equal to the sensing range (see Figure 9.7a). However, the maximum moving distance for a sensor is limited to at most half the communication range, because the local view of the Voronoi polygon might be incorrect because of limitations in communication range. VOR also applies the *movement adjustment scheme* as in VEC and additionally applies an *oscillation control scheme* that limits a sensor's movement to opposite directions in consecutive rounds.

The min–max algorithm is very similar to VOR, but it moves a sensor inside its Voronoi polygon to a point such that the distance from its farthest Voronoi vertex is minimized. Since moving a sensor to its farthest Voronoi vertex might lead to a situation such that the vertex that was originally close now becomes a new farthest vertex, the algorithm positions each sensor such that no vertex is too far away from the sensor. The authors define the concept of min–max circle, the center of which is the new targeted position. To find the min–max circle, all circumcircles of any two and any three Voronoi vertices are found and the one with minimum radius covering all the vertices is the min–max circle. The time complexity of this algorithm is in the cubic order of the number of Voronoi vertices.

9.5.6 Bidding Protocol (BIDP)

The algorithms described in the previous sections (PFA, VFA, DSSA, VEC, VOR, min–max) deal with sensor networks where all the nodes are mobile. However, there is a high cost associated with rendering each node mobile. Instead, a balance can be achieved by using both static and mobile sensors (mixed-sensor network), while still ensuring sufficient coverage. Wang et al. [43] describe such a protocol, called the *bidding protocol*, for mixed sensor networks. They reduce the problem to the NP-hard set covering problem and provide heuristics to solve it near-optimally.

Initially, a mixture of static and mobile nodes are randomly deployed in the sensing field. Next, the static sensors calculate their Voronoi polygons and find coverage holes with their polygons and also bid to the mobile sensors to move to holes' locations. If a hole is found, a static sensor chooses the location of the farthest Voronoi vertex as the target location of the mobile sensor and calculates the bid as $\pi(d - R_s)^2$, where d is the distance between the sensor and the farthest Voronoi vertex and R_s is the sensing range (see Fig. 9.7b). A static sensor then finds a closest mobile sensor whose base price (each mobile sensor has an associated base price that is initialized to zero) is lower than its bid and sends a bidding message to this mobile sensor. The mobile sensor receives all such bids from its neighboring static sensors and chooses the highest bid and moves to heal that coverage hole. The accepted bid becomes the mobile sensor's new base price. This approach ensures that a mobile sensor does not move to heal a coverage hole when its departure generates a larger hole in its original place. The authors also incorporate a self-detection algorithm to ensure that no two mobile sensors move to heal the same coverage hole. They also apply the *movement adjustment scheme* as described in VEC, to push sensors away from each other if their movement can guarantee more coverage.

9.5.7 Incremental Self-Deployment Algorithm (ISDA)

Howard and colleagues [17,19] presented an incremental and greedy self-deployment algorithm for mobile sensor networks, in which nodes are deployed one at a time into an unknown environment. Each node makes use of the information gathered by previously deployed nodes to determine its optimal deployment location. The algorithm ensures maximum coverage but at the same time guarantees that each node remains in line of sight with at least another node. Conceptually it is similar to the frontier-based approach [46], but here, occupancy maps are built from live sensory data and are analyzed to find frontiers between free space and unknown space. In the following, we highlight the four phases of the algorithm.

1. *Initialization Phase*. In this phase, nodes are assigned one of the three states: waiting, active, or deployed with the exception of a single node that acts as an anchor and is already deployed.
2. *Goal Selection Phase*. In this next phase an optimal location is chosen for the next node to be deployed on the basis of previously deployed sensors. The concept of *occupancy grid* [10] (see Fig. 9.8b) is used as the first step to global map building. Each cell is assigned a state of either free (known to contain no obstacles), occupied (known to contain one or more obstacle) or unknown. However, not all free space represents valid deployment locations because nodes have finite size and a free cell that is close to an occupied cell may not be reachable. Hence, the occupancy grid is further processed to build a *configuration grid* (see Fig. 9.8c). In a configuration grid, a cell is free iff all

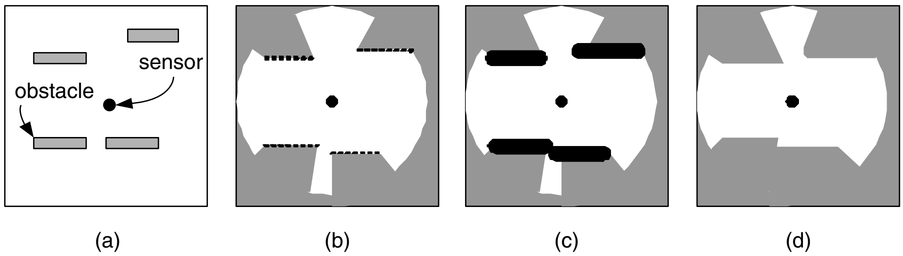

Figure 9.8 (a) Environment with obstacles and a single sensor; (b) occupancy grid — black cells are occupied, gray ones are unknown, and white ones are free; (c) configuration grid — black cells are occupied, gray ones are unknown, and white ones are free; (d) reachability grid–white cells are reachable, gray ones are unknown.

the occupancy grid cells lying within a certain distance are also free. A cell is occupied if there are one or more occupancy grid cells lying within a certain distance are similarly occupied. All other cells are marked as unknown. Once this global map is built, the goal selection phase chooses a location based on certain policies.

3. *Goal Resolution Phase*. Next, this new location is assigned to a waiting node in the goal resolution phase, and a plan for reaching the goal is generated applying a distance transform (also called *flood-fill algorithm*) on the configuration gird, giving rise to a *reachability grid* (see Fig. 9.8d). Thus, the set of reachable cells is a subset of the set of free configuration cells, which in turn is a subset of the set of free occupancy cells. A distance of 0 is assigned to the goal cell (the cell which is chosen to be the optimal location for the next node to be deployed), a distance of 1 to cells adjacent to the goal cell, a distance of 2 to their adjacent cells, and so on. However, distances are not propagated through occupied or unknown cells. Thus, for each node the distance to the goal and whether the goal can be reached is determined.
4. *Execution Phase*. In this phase, the active nodes are deployed sequentially to their respective goal locations. The nodes end up moving in a ‘‘conga line’’; specifically, as the lead node moves forward, the node immediately behind it steps forward to take its place; this node in turn is replaced by the one behind it, and so on.

9.5.8 Integer Linear Programming Algorithm (ILPA)

Chakrabarty et al. [4] model the optimization problem of coverage with integer linear programming (ILP) and represent the sensor field as a two- or three-dimensional grid. Given a variety of sensors with different ranges and costs, they provide strategies for minimizing the cost, provide coding-theoretic bounds on the number of sensors, and present methods for their placement with desired coverage. Their approach of maximizing coverage in the sensing field is different in the sense that it determines a deployment strategy, such that every gridpoint is covered by a unique subset of sensors. In this way, the set of sensors reporting a target at a particular time uniquely identifies the grid location for the target at that time.

9.5.9 Uncertainty-Aware Sensor Deployment Algorithm (UADA)

In most of the sensor deployment algorithms discussed so far, the optimal positions of the sensors are determined for maximizing coverage. However, there is an inherent uncertainty in sensor locations when sensors are dispersed, scattered, or airdropped. Hence, for every point in the sensing field there is only a certain probability of a sensor being located at that point. Zou and Chakrabarty [51] present two algorithms for efficient placement of sensors when exact locations are not known.

The sensor locations are modeled as random variables following Gaussian distribution. Let the intended sensor locations (x, y) be taken as mean values and σ_x, σ_y as standard deviations in the x and y dimensions, respectively. Assuming that these deviations are independent, the join distribution $p_{xy}(x', y')$ of a sensor's actual location is calculated. Then, the uncertainty in sensor location is modeled by a conditional probability $c^*_{ij}(x, y)$, for a gridpoint (i, j) to be detected by a sensor that is supposed to be deployed at (x, y). Hence, the miss probability (probability of missing) of a gridpoint (i, j) due to a sensor at (x, y) is calculated as $m_{ij}(x, y) = 1 - c^*_{ij}(x, y)$. From this, the collective miss probability of the gridpoint (i, j) due to a set L_s of already deployed sensors is given by $m_{ij} = \prod_{(x,y)\in L_s} (1 - c^*_{ij}(x, y))$. The algorithm then determines the location of the sensors one at a time. It finds out all possible locations that are available on the grid for the next sensor to be deployed and calculates the overall miss probability $m(x, y)$, due to the already deployed sensors and this sensor, assuming that it will be placed at (x, y): $m(x, y) = \sum_{(i,j)\in \text{Grid}} m_{ij}(x, y) m_{ij}$. Based on the $m(x, y)$ values, the current sensor can be placed at gridpoint (i, j) with maximum overall miss probability (worst-case coverage) or minimum overall miss probability (best-case coverage). Once the best location is found, the miss probabilities are updated and the process continues until each gridpoint is covered with a minimum confidence level. The complexity of the first phase of the algorithm where it calculates the conditional and miss probabilities is $O((mn)^2)$, for a $m \times n$ grid. The computational complexity of the second phase where the algorithm deploys the sensors is $O(mn)$.

9.5.10 Comparison of the Deployment Algorithms

The various sensor deployment strategies discussed in the previous sections, have different assumptions and goals depending on the underlying application requirements and the nature of the sensor network. Some of those strategies are applicable to mobile sensor networks, whereas some other ones are applicable only to static sensor networks. Then, a couple of the algorithms work under the scenarios having a mixture of static and mobile nodes. Therefore, the algorithms vary in terms of their applicability, complexity, and several other factors. In this section, we compare these sensor deployment strategies on the basis of their goals, advantages, disadvantages, performance, computation complexity, and applicability. These comparisons are summarized in Table 9.1.

TABLE 9.1 Comparison of Sensor Deployment Strategies

Algorithm	Network and Goals	Advantages	Disadvantages	Performance
IDA [9]	Static; minimize the number of sensors and communication traffic	Minimum confidence level for each grid; allows modeling of obstacles and preferential areas	Complete terrain knowledge assumed and sensor detections assumed to be independent	Outperforms random deployment in case of obstacles and preferential coverage; $O(n^2)$ time complexity, where n is the number of gridpoints
PFA [22,32]	Mobile; redeploy mobile nodes from an initial configuration to maximize coverage while maintaining at least k-connectivity	Good coverage without global maps; does not require centralized control, localization; hence scalable	Computationally expensive and assumes that each node can sense the exact relative range and bearing of its neighbors	Outperforms random deployment but poorer in performance than tiled networks (networks in which nodes are deployed in tiled patterns, e.g., haxagonal, triangular)
VFA [50,52]	Mobile; redeploy mobile nodes from an initial random placement to enhance coverage	One time computation and sensor location determination; allows modeling of obstacles and preferential areas	Centralized and extra computational capability of cluster head; no route plan for repositioning of nodes; efficiency depends on the force parameters; discrete coordinate system	Outperforms random placement. $O(nmk)$ time complexity for a $n \times m$ grid with k sensors deployed
DSSA [16]	Mobile; spread nodes from an initial random deployment to maximize coverage and maintain uniformity	Distributed self-deployment algorithm	Every node should know its own location; obstacles and preferential coverage areas are not modeled	Outperforms simulated annealing [41] (forms the basis of an optimization technique for combinatorial problems) in trms of uniformity, deployment time, and the mean distance traveled by the nodes to reach their final locations.

(continued)

TABLE 9.1 (*Continued*)

Algorithm	Network and Goals	Advantages	Disadvantages	Performance
VEC, VOR, min–max, [44]	Mobile; reduce or eliminate coverage holes by relocating mobile sensors	Distributed algorithms; extensible to large deployment because communication and movements are local	Poor performance on initial clustered deployment and lower communication range	Perform better if initial deployment is random rather than clustered; does not perform well in case of insufficient communication range
BIDP [43]	Mixed; reduce or eliminate coverage holes and minimize cost by relocating mobile nodes	Distributed protocol; provides cost balance by using a combination of static and mobile nodes	No obstacle modeling; performance depends on ratio of mobile to static sensors	Coverage increases as the percentage of mobile sensors increases; however, duplicate healing occurs and average movement distance of sensors increases along with it
ISDA [17,19]	Mobile; deploy mobile nodes in an unknown environment to maximize coverage while retaining line of sight	Incremental and greedy algorithm; not dependent on prior environment models; global maps are built from live sensory data	It takes a long time; difficult to make it scalable for large networks	Coverage increases linearly with the number of deployed sensors; $O(n^2)$ worst-case time complexity, where n is the number of deployed sensors
ILPA [4]	Static; provide maximum grid coverage for surveillance and target detection, while minimizing the cost of sensors	Targets can be uniquely identified from the subset of sensors that detect the targets	High computational complexity makes it infeasible for large-scale deployment; relies on perfect binary sensing model	Computationally very expensive
UADA [51]	Static; determine minimum number of sensors and their locations under constraint of imprecise detection and terrain properties	Each gridpoint is covered with a minimum confidence level; models sensor locations as random variables	Sensor detections are assumed to be independent; computationally expensive	Time complexity of first phase of algorithm is $(O(mn)^2)$ and that of second phase is $O(mn)$, for a $m \times n$ gird

9.6 MISCELLANEOUS STRATEGIES

Our discussion so far has concerned mainly algorithms that guarantee optimal coverage of the sensing field. However, as mentioned earlier, a sensor network needs to be connected as well, so that the data sensed by the nodes can be transmitted by multihop communication paths to other nodes and possibly to a basestation where intelligent decisions can be made. Therefore, it is equally important for a coverage algorithm to ensure a connected network. In this section, we will discuss a few techniques that ensure coverage as well as connectivity in a sensing field while at the same time reduce redundancy and increases overall network lifetime.

It is envisioned that a typical wireless sensor network would consist of large numbers of energy-constrained nodes deployed with high density. In such a network, it is sometimes undesirable to have all the nodes in the active state simultaneously, because there would be redundancy in sensing and excessive packet collisions. Also, keeping all the nodes active simultaneously would dissipate energy at a much faster rate and would reduce overall system lifetime. Hence, it is important to turn off the redundant nodes and maximize the time interval of a continuously monitoring, transmitting, or receiving function. Scheduling of nodes that would control the density of active nodes in a sensor network has been the focus of many research works. An optimal scheduling scheme ensures that only a subset of nodes are active at any given point of time, while satisfying the following two requirements relating coverage and connectivity:

1. The area that can be monitored by the working set of nodes is not smaller than the area that can be monitored by the set of all nodes.
2. Network connectivity is maintained even after turning off the redundant nodes.

Zhang and Hou [48] proposed a decentralized and localized density control algorithm [Optimal geographic density control (OGDC)] based on certain optimality conditions of coverage and connectivity for large-scale sensor networks. They investigated the relation between coverage and connectivity and proved that if the communication range is at least twice the sensing range $(R_c \geq 2R_s)$, then complete coverage of an area guarantees a connected network. The OGDC algorithm tries to minimize the overlap of sensing areas of all the nodes and finds a node scheduling scheme. It defines the notion of a *crossing point* as an intersection point of the sensing circles of two nodes (see Fig. 9.9a) and proves that to cover one crossing point of two nodes with minimum overlap, only one other node should be used and the centers of the three nodes should form an equilateral triangle with side-length $\sqrt{3}\,R_s$. As illustrated in Figure 9.9a, nodes A and B have two crossing points. To cover that crossing point optimally, another node C should be placed such that the centers of the three nodes form an equilateral triangle ΔABC. Furthermore, to cover one crossing point of two nodes whose positions are fixed (i.e., with x_1 fixed), only one disk[3] should be used and $x_2 = x_3 = (\pi - x_1)/2$.

[3]Refer to Section 9.3.1 for definition of a *disk*.

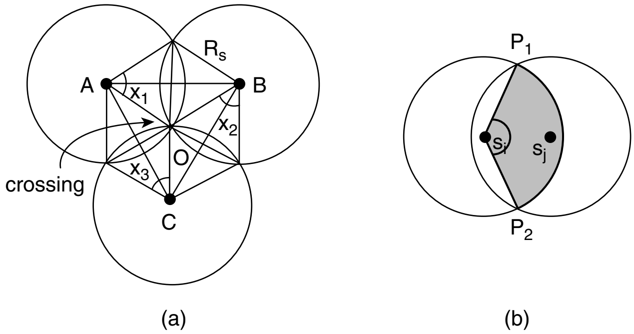

Figure 9.9 (a) Optimal positions of sensors to minimize overlap; (b) sponsored sector.

Tian and Georganas [40] proposed a self-scheduling scheme that can reduce overall energy consumption and increase system lifetime by exploiting the redundancy of nodes. Their approach is based on sponsorship criteria, by which each node decides whether to turn itself off or on using only local neighborhood information. We define the notion of sponsorship and the algorithm in the following.

Definition 9.7 (**Sponsor Nodes**) Let $N(i)$ denote the set of one-hop neighbors of node s_i. Node s_i is said to be sponsored by its one-hop neighbors if the union of its neighbors' sensing areas is a superset of node s_i's sensing area. If we denote the sensing area of a node s_i as $S(i)$, then the sponsorship criterion is $\bigcup_{j\in N(i)} S(j) \supseteq S(i)$.

Definition 9.8 (**Sponsored Sector**) Let the sensing areas of node s_i and one of its one-hop neighbors s_j intersect at points P_1 and P_2, respectively, as shown in Figure 9.9b. The area bounded by radius s_iP_1, radius s_iP_2, and the inner arc $\widehat{P_1P_2}$ is called the *sponsored sector* of node s_i by node s_j. The central angle of the sector is denoted as $Q_{j\to i}$, which lies in the interval $120 \leq \theta_{j\to i} \leq 180$.

Gao, et al. [13] proved that at least three and at most five one-hop neighbors are needed to cover the whole sensing area of node s_i.

The algorithm described by Tian and Georganas [40] consists of two phases: self-scheduling phase and sensing phase. In the self scheduling phase, each sensor broadcasts its position and node id, and listens to the advertisement messages from its neighbors to obtain their location information. Then it calculates the sponsored sectors by its neighbors and checks whether the union of their sponsored sectors can cover its own sensing area. If so, it decides to turn itself off. However, if all the nodes make decisions simultaneously, blindspots might appear. To avoid such a situation, each node waits a random period of time and also broadcasts its status message to other nodes. In this way the nodes self-schedule, thus reducing energy consumption while maintaining the original coverage area.

Ye et al. [47] described a distributed localized algorithm for density control based on probing mechanism. In their algorithm, each node can be in one of the three states: sleeping, wakeup, or working. A working node is responsible for sensing and data communication, while nodes in wakeup state prepare themselves for replacing a dying node due to energy depletion or other kinds of failures. A sleeping node wakes up after sleeping for an exponentially distributed period of time (termed as wakeup rate λ) and broadcasts a probing message within a radius of r. If there are any working nodes in the vicinity, they reply back to the wakeup node. If the wakeup node hears such a reply message, it knows that there is a working node within its probing range r and goes back to sleep again. If the wakeup node does not hear a reply message within a certain time, it assumes that there is no working node within its probing range and it starts working. By tuning the parameters λ and r in simulations, the authors show that an optimal node density can be achieved while ensuring that each area is monitored by at least k working nodes. The algorithm is fully distributed and localized, and no neighborhood topology discovery is necessary. The computation and memory overhead per node is also negligible and is independent of the number of neighbors.

Shakkottai et al. [38] considered an unreliable wireless sensor grid network with n nodes placed over a unit area. Defining $r(n)$ as the transmitting radius of each node, and $p(n)$ as the probability that a node is active at some time t, they found that the necessary and sufficient condition for the grid network to cover the unit square region as well as ensure that the active nodes are connected is of the form $p(n)\dot{r}^2(n) \approx \log(n)/n$. This result indicates that when n is large, each node can be highly unreliable and the transmission power can be small and we can still maintain connectivity with coverage. They have also shown that the diameter of the random grid (i.e., the maximum number of hops required to travel from any active node to another) is of the order of $\sqrt{n/\log(n)}$. A corollary of this is that the shortest-hop path between any pair of nodes is nearly the same as a straight-line path between the nodes. Finally, the authors derived a sufficient condition for connectivity of the active nodes (without necessarily having coverage) and showed that if $p(n)$ is small enough, connectivity does not imply coverage.

9.7 DISCUSSIONS AND CONCLUSIONS

In this chapter, we have discussed the importance of coverage and connectivity, which are two fundamental factors for ensuring efficient resource management in wireless sensor networks, and surveyed various methods and protocols, which optimally cover a sensing field while maintaining global network connectivity at the same time. We have seen that exposure paths can be viewed as a measure of goodness of detectability of a moving target in a sensing field. The notions of min–max exposure paths, breach paths, and support paths provide critical information to the application in terms of identifying sparsely and densely covered areas. We also discussed and compared several node deployment algorithms for static and mobile as well as for mixed-sensor networks, and observed that, depending on the coverage requirements, topological information, presence of obstacles, and other variables, the algorithms vary with respect to their goals, assumptions, and complexities.

The node scheduling schemes that we described under miscellaneous strategies using the notion of sponsored sectors ensure longer network lifetime and guarantee uniform dissipation of battery power throughout the network. This in turn implies better resource management.

However, the works existing in the literature have not addressed some of problems on theoretical bounds related to coverage and connectivity. Although Zhang and Hou [48] provided a theoretical result, proving that if the communication range is at least twice the sensing range, then complete coverage of an area guarantees a connected network, the probabilistic bounds on the number of nodes for a certain percentage of coverage is still unresolved. Future research in this area would provide insights into the probabilistic bounds on the best coverage that one can achieve given a number of nodes. The problem can be formulated as, given a total N number of nodes and a rectangular sensing field $A = a \times b$, with what probability one can guarantee p percentage coverage, while ensuring k degrees of connectivity across the network. This is a combinatorial optimization problem and can be tackled using statistical techniques and integer linear programming.

The deployment of nodes in mixed-sensor networks, which require one to strike a balance between the number of static and mobile sensors, involves the optimization of a cost/performance-based objective function and is therefore challenging. We discussed one approach [43] that initially deploys a fixed number of static and mobile nodes in a sensing field, after which the static nodes are required to find local coverage holes and bid for mobile sensors to relocate to the targeted locations and reduce or eliminate those holes, thus increasing area coverage. However, this approach has a drawback because it deploys a fixed number of mobile nodes.

To overcome this shortcoming, we [14] considered a mixed-sensor network, where initially a fixed number of static nodes are deployed, which deterministically find the exact amount of coverage holes existing in the entire network using the structure of Voronoi diagrams and then dynamically estimate the additional number of mobile nodes needed to be deployed and relocated to the optimal locations of the holes to maximize overall coverage. This approach of deploying a fixed number of static nodes and a varying estimated number of mobile nodes can provide optimal coverage under controlled cost. A mixed sensor approach is a very attractive one, because it allows one to choose the degree of coverage required by the underlying application as well as gives an opportunity to optimize on the number of additional mobile nodes needed to be deployed. We [15] provided distributed algorithms to find suboptimal minimum connected sensor covers, such that the whole sensing field is covered using a suboptimal number of sensors. In another study [5,7] we proposed a novel energy conserving data gathering strategy based on a tradeoff between coverage and data reporting latency with the ultimate goal of maximizing a network's lifetime. The basic idea is to select in each data reporting round only a minimal number of k sensors as data reporters, based on a desired sensing coverage specified by the user or application. Besides conserving energy, such a selection of minimum data reporters also reduces the amount of traffic flow, thus avoiding traffic congestion and channel interference. Simulation results of our proposed schemes demonstrate that the user-specified percentage of the monitored area can be covered using only k sensors. It also shows that the sensors can conserve a significant amount of energy with a small tradeoff and that the higher the network density, the higher is the energy conservation rate without any additional computation cost. In one of our works [6] for efficient resource management in wireless sensor networks, we presented a two-phase clustering scheme for energy saving and delay-adaptive data gathering in order to extend a network's lifetime.

Further research on optimization algorithms in mixed sensor networks and evaluating tradeoffs between latency and data gathering strategies can provide valuable information to optimize resources in a sensing field and help answer questions related to the theoretical bounds on coverage and connectivity.

ACKNOWLEDGMENT

This work is supported by NSF grant IIS-0326505.

REFERENCES

1. S. Adlakha and M. Srivastava, Critical density thresholds for coverage in wireless sensor networks, *Proc. IEEE Wireless Communications and Networking Conf. (WCNC'03)*, New Orleans, LA, March 2003, pp. 1615–1620, Louisiana, Mar. 2003.
2. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: A survey, *Comput. Networks* **38**(2):393–422 (2002).
3. L. Booth, J. Bruck, and R. Meester, Covering algorithms, continuum percolation and the geometry of wireless networks, *Annals Appl. Probability* **13**(2):722–741 (May 2003).
4. K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, Grid coverage for surveillance and target address in distributed sensor networks, *IEEE Trans. Comput.* **51**(12):1448–1453 (Dec. 2002).
5. W. Choi and S. K. Das, in S. Phoha and T. La Porta, eds., *An Energy-conserving Data Gathering Strategy Based on Trade-off between Coverage and Data Reporting Latency in Wireless Sensor Networks*, Sensor Network Operations, IEEE Press, 2004.
6. W. Choi and S. K. Das, A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks, *Proc. Mobile and Ubiquitous Systems: Networking and Services Conf., Mobiquitous'04*, Boston, Aug. 2004, pp. 203–212.
7. W. Choi and S. K. Das, Trade-off between coverage and data reporting latency for energy-conseving data gathering in wireless sensor networks, *Proc. 1st Int. Conf. Mobile Ad Hoc and Sensor Systems, MASS'04*, Ft. Lauderdale, FL, Oct. 2004.
8. T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, Sensor deployment strategy for target detection, *Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications* (*WSNA'02*), Atlanta, GA, Sept. 2002, pp. 42–48.
9. S. S. Dhilon, K. Chakrabarty, and S. S. Iyengar, Sensor placement for grid coverage under imprecise detections, *Proc. 5th Int. Conf. Information Fusion* (*FUSION'02*), Annapolis, MD, July 2002, pp. 1–10.
10. A. Elfes, Occupancy grids: A stochastic spatial representation for active robot perception, *Proc. 6th Conf. Uncertainty in AI*, Cambridge, MA, July 1990, pp. 60–70.

11. S. V. Fomin and I. M. Gelfand, *Calculus of Variations*, Dover Publications, Oct. 2000.
12. D. W. Gage, Command control for many-robot systems, *Proc. 19th Annual AUVS Technical Symp*. Reprinted in *Unmanned Syst. Mag.* **10**(4):28–34 (Jan. 1992).
13. Y. Gao, K. Wu, and F. Li, Analysis on the redundancy of wireless sensor networks, *Proc. 2nd ACM Int. Conf. Wireless Sensor Networks and Applications* (*WSNA'03*), San Diego, CA, Sept. 2003 pp. 108–114.
14. A. Ghosh, Estimating coverage holes and enhancing coverage in mixed sensor networks, *Proc. 29th Annual IEEE Conf. Local Computer Networks* (*LCN'04*), Tampa, FL, Nov. 2004, pp. 68–76.
15. A. Ghosh and S. K. Das, Distributed connected sensor cover algorithms for lattice and random deployment of nodes in dense sensor networks, *Proc. 1st IEEE/ACM Int. Conf. Distributed Computing in Sensor Systems* (*DCOSS'05*), Marina del Rey, CA, June–July 2005.
16. N. Heo and P. K. Varshney, A distributed self-spreading algorithm for mobile wireless sensor networks, *Proc. IEEE Wireless Communications and Networking Conf.* (*WCNC'03*), New Orleans, LA, March 2003, pp. 1597–1602.
17. A. Howard, M. J. Matari, and G. S. Sukhatme, An incremental self-deployment algorithm for mobile sensor networks, *Autonomous Robots* special issue on intelligent embedded systems **13**(2):113–126 (2002).
18. A. Howard, M. Mataric, and G. Sukhatme, Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem, *Proc. 6th Int. Symp. Distributed Autonomous Robotic Systems* (*DARS'02*), Fukuoka, Japan, June 2002, pp. 299–308.
19. A. Howard and M. J. Mataric, Cover me! A self-deployment algorithm for mobile sensor networks, *Proc. IEEE Int. Conf. Robotics and Automation* (*ICRA'02*), Washington DC, May 2002, pp. 80–91.
20. C.-F. Huang and Y.-C. Tseng, The coverage problem in a wireless sensor network, *Proc. 2nd ACM Int. Conf. Wireless Sensor Networks and Applications* (*WSNA'03*), San Diego, CA, Sept. 2003, pp. 115–121.
21. J. M. Kahn, R. H. Katz, and K. S. J. Pister, Next century challenges: Mobile networking for smart dust, *Proc. 5th Annual ACM/IEEE Int. Conf. Mobile Computing and Networking* (*MOBICOM'99*), Seattle, WA, Aug. 1999.
22. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robotics Res.* **5**(1):90–98 (1986).
23. M. M. Kokar, J. A. Tomasik, and J. Weyman, Data vs. decision fusion in the category theory framework, *Proc. 4th Int. Conf. Information Fusion* (*FUSION'01*), Montreal, Australia, Aug. 2001.
24. X.-Y. Li, P.-J. Wan, and O. Frieder, Coverage in wireless ad-hoc sensor networks, *IEEE Trans. Comput.* **52**:753–763 (2003).
25. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless sensor networks for habitat monitoring, *Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications* (*WSNA'02*), Atlanta, GA, Sept. 2002, pp. 88–97.
26. S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, Exposure in wireless sensor networks: Theory and practical solutions, *Wireless Networks* **8**(5):443–454 (2002).

27. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, Coverage problems in wireless ad-hoc sensor networks, *Proc. IEEE InfoCom* (*InfoCom'01*), Anchorage, AK, April 2001, pp. 115–121.
28. S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, Exposure in wire less ad-hoc sensor networks, *Proc. 7th Annual Int. Conf. Mobile Computing and Networking* (*MobiCom'01*), Rome, Italy, July 2001, pp. 139–150.
29. A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications ofVoronoi Diagrams*, 2nd ed., Wiley July 2000.
30. J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, Oxford, UK, 1987.
31. M. Penrose, The longest edge of the random minimal spanning tree, *Annals Appl. Probability* **7**(2):340–361 (May 1997).
32. S. Poduri and G. S. Sukhatme, Constrained coverage in mobile sensor networks, *Proc. IEEE Int. Conf. Robotics and Automation* (*ICRA'04*), New Orleans, LA, April–May 2004, pp. 40–50.
33. G. J. Pottie, Wireless sensor networks, *Proc. Information Theory Workshop*, June 1998, pp. 139–140.
34. G. J. Pottie and W. Caiser, Wireless sensor networks, *Commun. ACM* **43**(5):51–58 (May 2000).
35. P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, Wiley, May 2005.
36. P. Santi and D. M. Blough, The critical transmitting range for connectivity in sparse wireless ad hoc networks, *IEEE Trans. Mobile Comput.* **2**(1):25–39 March 2003.
37. R. C. Shah and J. M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, *Proc. IEEE Wireless Communications and Networking Conf.* (*WCNC'02*), Orlando, FL, March 2002.
38. S. Shakkottai, R. Srikant, and N. Shroff, Unreliable sensor grids: Coverage, connectivity and diameter, *Proc. IEEE InfoCom* (*InfoCom'03*), pages 1073–1083, San Francisco, CA, March 2003.
39. C. Shurgers and M. B. Srivastava, Energy efficient routing in wireless sensor networks, *Proc. Military Communications Conf.* (*MilCom'01*), Vienna, VA, Oct. 2001.
40. D. Tian and N. D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, *Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications* (*WSNA'02*), Atlanta, GA, Sept. 2002, pp. 32–41.
41. P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Reidel Publishing, Kluwer, 1987.
42. G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, Minimal and maximal exposure path algorithms for wireless embedded sensor networks, *Proc. 1st Int. Conf. Embedded Networked Sensor Systems* (*SenSys'03*), Los Angeles, Nov. 2003, pp. 40–50.
43. G. Wang, G. Cao, and T. LaPorta, A bidding protocol for deploying mobile sensors, *Proc. 11th IEEE Int. Conf. Network Protocols* (*ICNP'03*), Atlanta, GA, Nov. 2003, pp. 80–91.
44. G. Wang, G. Cao, and T. LaPorta, Movement-assisted sensor deployment, *Proc. IEEE InfoCom* (*InfoCom'04*), Hong Kong, March 2004, pp. 80–91.
45. D. B. West, *Introduction to Graph Theory*, 2nd ed., Prentice-Hall, Aug. 2003.

46. B. Yamauchi, A frontier-based approach for autonomous exploration, *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation* (*CIRA'97*), Monterey, CA, June 1997, pp. 146–156.
47. F. Ye, G. Zhong, S. Lu, and L. Zhang, Peas: A robust energy conserving protocol for long-lived sensor networks, *Proc. 10th IEEE Int. Conf. Network Protocols* (*ICNP'02*), Paris, Nov. 2002, pp. 200–201.
48. H. Zhang and J. C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, *Proc. Int. Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks* (*AlgoSensors'04*), Florida, Feb. 2004.
49. F. Zhao, J. Liu, J. Liu, et al. Collaborative signal and information processing: An information-directed approach, *Proc. IEEE* **91**(8):1199–1209, (2003).
50. Y. Zou and K. Chakrabarty, Sensor deployment and target localization based on virtual forces, *Proc. IEEE InfoCom* (*InfoCom'03*), San Francisco, CA, April 2003, pp. 1293–1303.
51. Y. Zou and K. Chakrabarty, Uncertainty-aware sensor deployment algorithms for surveillance applications, *Proc. IEEE Global Communications Conf.* (*GLOBECOM'03*), Dec. 2003.
52. Y. Zou and K. Chakrabarty, Sensor deployment and target localization in distributed sensor networks, *Trans. IEEE Embedded Comput. Syst*. **3**(1):61–91 (2004).