

A Probabilistic Coverage Protocol for Wireless Sensor Networks

Mohamed Hefeeda
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Hossein Ahmadi
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Abstract—We propose a new probabilistic coverage protocol (denoted by PCP) that considers probabilistic sensing models. PCP is fairly general and can be used with different sensing models. In particular, PCP requires the computation of a single parameter from the adopted sensing model, while everything else remains the same. We show how this parameter can be derived in general, and we actually do the calculations for two example sensing models: (i) the probabilistic exponential sensing model, and (ii) the commonly-used deterministic disk sensing model. The first model is chosen because it is conservative in terms of estimating sensing capacity, and it has been used before in another probabilistic coverage protocol, which enables us to conduct a fair comparison. Because it is conservative, the exponential sensing model can be used as a first approximation for many other sensing models. The second model is chosen to show that our protocol can easily function as a deterministic coverage protocol. In this case, we compare our protocol against two recent deterministic protocols that were shown to outperform others in the literature. Our comparisons indicate that our protocol outperforms all other protocols in several aspects, including number of activated sensors and total energy consumed. We also demonstrate the robustness of our protocol against random node failures, node location inaccuracy, and imperfect time synchronization.

I. INTRODUCTION

Sensor networks have been proposed for many applications such as forest fire detection, area surveillance, and natural habitat monitoring [1]. A common ground for all such applications is that every sensor can detect an event occurring within its *sensing range*, and sensors collaborate in some way to deliver events, or information related to these events, to processing centers for possible actions.

In many of the previous works, the sensing range is assumed to be a uniform disk of radius r_s . The disk model assumes that if an event happens at a distance less than or equal to r_s from the sensor location, the sensor will deterministically detect this event. On the other hand, an event occurring at a distance $r_s + \epsilon$ ($\epsilon > 0$) can not be detected at all, even for very small ϵ values (see Fig. 1(a)). The disk sensing model is appealing, because it makes coverage maintenance protocols, e.g., [2]–[4], less complicated to design and analyze. It also makes analytical and asymptotic analysis, e.g., [5], [6], tractable. However, it is unlikely that physical signals drop abruptly from high, full-strength values to zero, as the disk model assumes. This implies that there might be a chance to detect an event occurring at distances greater than r_s . By

ignoring this extra sensing capacity, the disk model may not fully utilize the sensing capacity of sensors, which may lead to: (i) deploying more sensors than needed and thus incurring higher cost, (ii) activating redundant sensors which increases interference and wastes energy, and ultimately (iii) decreasing the lifetime of the sensor network.

Several studies [7]–[11] have argued that probabilistic sensing models capture the behavior of sensors more realistically than the deterministic disk model. For example, through experimental study of passive infrared (PIR) sensors, the authors of [11] show that the sensing range is better modeled by a continuous probability distribution, which is a normal distribution in the case of PIR sensors. The authors of [7], [8] use an exponential sensing model, where the sensing capacity degrades according to an exponential distribution after a certain threshold, as shown in Fig. 1(b). Whereas the authors of [10] propose a polynomial function to model the probabilistic nature of the sensing range, as shown in Fig. 1(d). Furthermore, the authors of [9] assume that the sensing range can be modeled as layers of concentric disks with increasing diameters, and each layer has a fixed probability of sensing, as shown in Fig. 1(c). A probabilistic sensing model is more realistic because the phenomenon being sensed, sensor design, and environmental conditions are all stochastic in nature. For instance, noise and interference in the environment can be modeled by stochastic processes. Sensors manufactured by the same factory are not deterministically identical in their behavior, rather, sensor characteristics are usually modeled using statistical distributions.

While more realistic, probabilistic sensing models introduce new challenges for coverage protocols in sensor networks. First, the sensing range of a sensor is no longer a nice regular disk, and therefore, it becomes harder to define the notion of overlapping between sensing ranges of different sensors. This notion is critical in coverage protocols, e.g., OGDC [4], that minimize overlapping between sensing ranges to activate the minimum number of sensors while ensuring full coverage. This implies that directly using probabilistic sensing models in coverage protocols that assume disk sensing model may yield incorrect functioning of these protocols, such as terminating while some subareas are uncovered, or activating more sensors than actually needed. Most of the current coverage protocols, including CCP [2], PEAS [12], Ottawa [13], and OGDC [4],

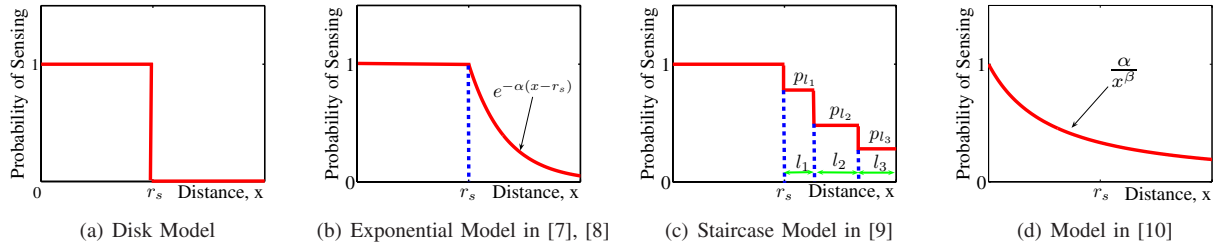


Fig. 1. Some of the sensing models used in the literature.

assume disk sensing model. Second, the traditional definition of the coverage itself—which states that every point in the area must be within the sensing range of at least one sensor—is no longer valid because of the probabilistic nature of the sensing range. Therefore, a new definition for coverage is needed when probabilistic sensing models are considered.

In this paper, we propose a new probabilistic coverage protocol (denoted by PCP) that considers probabilistic sensing models. We design PCP keeping in mind that no single sensing model (probabilistic or not) will accurately model all types of sensors in all environments. It is expected that different sensor types will require different sensing models. Even for the same sensor type, the sensing model may need to be changed in different environments or when the technology changes. Designing, implementing, and testing a different coverage protocol for each sensing model is indeed an extremely costly process, if at all possible. To address this challenging task, we design our protocol with limited dependence on the sensing model. In particular, our protocol requires the computation of a single parameter from the adopted sensing model, while everything else remains the same. We show in this paper how this parameter can be derived in general, and we actually do the calculations for two sensing models: (i) the probabilistic exponential sensing model [7], [8], and (ii) the commonly-used deterministic disk sensing model. The first model is chosen because it is conservative in terms of estimating sensing capacity, and it has been used before in another probabilistic coverage protocol (CCANS [8]). This enables us to compare our protocol against CCANS, which is the only fully-specified probabilistic coverage protocol that we are aware of. Also because it is conservative, the exponential sensing model can be used as a first approximation for many other sensing models. The second model is chosen to show that our protocol can easily function as a deterministic coverage protocol. In this case, we compare our protocol against two recent deterministic protocols that were shown to outperform others in the literature. Our comparisons indicate that our protocol outperforms the other two in several aspects, including number of activated sensors and total energy consumed. We also demonstrate its robustness against random node failures, node location inaccuracy, and imperfect time synchronization.

The rest of the paper is organized as follows. We summarize the related work in Section II. In Section III, we formally define the probabilistic coverage problem, and present the

key ideas behind our new probabilistic coverage protocol. In Section IV, we present the details of our new protocol, and in Section V, we prove its correctness and provide bounds on its convergence time and message complexity. We also prove the condition on the communication range needed for our protocol to provide connectivity in addition to coverage. In Section VI, we evaluate our protocol and compare it against other deterministic and probabilistic coverage protocols in the literature. We conclude the paper in Section VII.

II. RELATED WORK

Coverage in sensor networks has received significant research attention. The studies in [5], [6] conduct asymptotic and analytical analysis to provide necessary and sufficient conditions for coverage in various environments. In [14], optimal deployment patterns for different ratios of the communication and sensing ranges are proposed. While these studies provide useful insights and guidelines, which we indeed benefited from, they do not propose specific coverage protocols.

Several distributed coverage protocols have been proposed for the disk model. For example, OGDC [4] tries to minimize the overlap between the sensing circles of activated sensors, while CCP [2] deactivates redundant sensors by checking that all intersection points of sensing circles are covered. Other earlier protocols include PEAS [12] and Ottawa [13]. We compare our protocol against the more recent OGDC and CCP protocols, because, according to the performance evaluations in [2], [4], they outperform the earlier ones.

Probabilistic coverage with various sensing models has also been studied in [8]–[10]. The work in [10] analytically studies the implications of adopting probabilistic and disk sensing models on coverage, but no specific coverage protocol is presented. In [9], the sensing range is modeled as layers of concentric disks with increasing diameters, where the probability of sensing is fixed in each layer. A coverage evaluation protocol is also proposed. Although the authors mention that their coverage evaluation protocol can be extended to a dynamic coverage protocol, they do not specify the details of that protocol. Therefore, we could not compare our protocol with theirs. The closest work to ours is [8], where the authors assume that the sensing capacity decreases exponentially fast after certain threshold. The authors also design a probabilistic coverage protocol (CCANS) based on that model. We use the same sensing model in our coverage protocol and compare

it against CCANS. Unlike CCANS, our protocol can utilize different probabilistic and deterministic sensing models.

Finally, a closely-related problem to coverage is connectivity. k -connectivity ($k \geq 1$) means that there are at least k disjoint paths between any pair of nodes in the network. For the disk sensing and communication models, it has been proven that if the communication range of sensors is at least twice the sensing range and the monitored area is convex, then k -coverage implies k -connectivity [2], [4], [8]. In this paper we prove the conditions under which probabilistic coverage ensures 1-connectivity.

III. PROBABILISTIC COVERAGE

In this section, we define the notion of probabilistic coverage, and we discuss the key ideas behind our probabilistic coverage protocol. We start by presenting some useful facts on coverage using the disk sensing model. Then, we discuss coverage using probabilistic sensing models.

A. Coverage using Disk Sensing Model

The disk sensing model simplifies the coverage problem. In fact, optimal solutions for it can be obtained efficiently. As mentioned in [14], covering an area with disks of same radius (r_s) can optimally be done by placing disks on vertices of a triangular lattice, where the side of the triangle is $\sqrt{3}r_s$. We can use this triangular lattice idea in designing a coverage protocol that activates a minimal subset of deployed sensors to ensure coverage as follows. The protocol works by first activating any sensor in the area. This sensor activates six other sensors located at vertices of the hexagon centered at that sensor. Each activated sensor in turn activates other sensors at vertices of its own hexagon. This process continues till the activated sensors form a virtual triangular lattice over the whole area. Activating sensors in this way minimizes the *overlap* between the sensing ranges of sensors. The above protocol is idealistic and many practical issues need to be addressed, as will be discussed later.

B. Coverage using Probabilistic Sensing Models

Under probabilistic sensing models, the sensing range is no longer a disk. Furthermore, the overlap among sensing ranges of different sensors is not clearly defined. Therefore, the overlap minimization idea may not work with probabilistic coverage protocols that seek to optimize the number of activated sensors. For such protocols, we propose a new method for activating the minimum number of sensors while ensuring the monitored area is probabilistically covered. We first state two definitions that we use in the discussion.

Definition 1 (Probabilistic Coverage): An area A is probabilistically covered by n sensors with threshold parameter θ ($0 < \theta \leq 1$) if $P(x) = 1 - \prod_{i=1}^n (1 - p_i(x)) \geq \theta$ for every point x in A , where $p_i(x)$ is the probability that sensor i detects an event occurring at x .

Note that $P(x)$ in the above definition measures the collective probability from all n sensors to cover point x , $p_i(x)$ is specified by the adopted sensing model, and the coverage threshold parameter θ depends on the requirements of the

target application. If we set $\theta = 1$ and $p_i(x)$ as a binary function that takes on either 0 or 1 in the above definition, we get the commonly-used deterministic coverage definition with the disk sensing model.

Definition 2 (Least-covered Point): A point x within an area A is called the least-covered point of A if $P(x) \leq P(y)$ for all $y \neq x$ in A .

The main idea of our probabilistic coverage protocol is to ensure that the least-covered point in the monitored area has a probability of being sensed that is at least θ . To implement this idea in a distributed protocol with no global knowledge, we divide the area into smaller subareas. For each subarea, we determine the least-covered point in that subarea, and we activate the minimum number of sensors required to cover the least-covered point with a probability more than or equal to θ . To enable our protocol to work optimally under the disk sensing model as well as probabilistic sensing models, we divide the monitored area into equi-lateral triangles forming a triangular lattice. Now we need to compute the location of the least-covered point in each triangle. Then, we need to compute the maximum length of the triangle side at which the probability of sensing at the least-covered point is at least θ . Knowing this maximum length, the coverage protocol functions in the same manner as described in Section III-A: It tries to activate nodes at vertices of the lattice triangles. Notice that this is an *idealistic* version of our protocol to describe the core idea. Practical considerations, such as inaccuracies in node locations, are handled later in the paper. Notice also that the main difference between the deterministic and probabilistic coverage protocols is that the former tries to minimize the overlap between sensing ranges, while the latter stretches the separation between active sensors to its maximum while ensuring that the coverage at the least-covered point exceeds a given threshold θ .

We refer to the maximum length of the triangle side as the maximum separation between any two active sensors, and we denote it by s . Computing s depends only on the sensing model used. In the next subsection, we derive s for two sensing models: the exponential sensing model [7], [8], and the disk sensing model. Computing s for other sensing models can be done in a similar way. We should emphasize that the operation of our probabilistic coverage protocol (PCP), described in detail in Sections IV and V, does *not* change by changing the sensing model. The only parameter that needs to be determined and given to PCP is the maximum separation between any two active sensors s , which is computed from the sensing model.

C. Computing Maximum Separation for Exponential and Disk Sensing Models

This section presents the details of deriving the maximum separation s between any two active nodes for two example sensing models. s is the only required parameter that needs to be computed from the sensing model for our coverage protocol.

The first model that we derive s for is the exponential

sensing model, which is defined as:

$$p(d) = \begin{cases} 1, & \text{for } d \leq r_s \\ e^{-\alpha(d-r_s)}, & \text{for } d > r_s \end{cases} \quad (1)$$

where $p(d)$ is the probability of detecting an event happening at a distance d from the sensor, r_s is a threshold below which the sensing capacity is strong enough such that any event will be detected with probability 1, and α is a factor that describes how fast the sensing capacity decays with distance. We call α the sensing capacity decay factor. The exponential model is shown Fig. 1(b). We consider this sensing model for two reasons. First, it has been adopted before in [7], [8], which allows us to conduct a fair comparison between our protocol and the protocol in [8]. Second, it is conservative as it assumes that the sensing capacity decreases exponentially fast beyond r_s , which means that the achieved actual coverage will be higher than the estimated by the theoretical analysis. In addition, since the exponential sensing model is conservative, it can be used as a first approximation for other sensing models such as those in [9]–[11]. Therefore, sensor network designers may not need to compute the exact value of the maximum separation parameter for mathematically complex sensing models, and instead use the exponential sensing model.

The following theorem provides the maximum separation between any two active nodes s for the exponential sensing model. The sketch of the proof is given below.

Theorem 1 (Maximum Separation): Under the exponential sensing model defined in (1), the maximum separation between any two active sensors on the triangular lattice to ensure that the probability of sensing at the least-covered point is at least θ is $\sqrt{3}(r_s - \frac{\ln(1-\frac{\sqrt[3]{1-\theta}}{\alpha})}{\alpha})$.

Proof: To prove this theorem, we need to find the location of the least-covered point. Using some geometrical properties of triangles, it is shown in [15] that this location is at the center of the triangle, which is at a distance of $s/\sqrt{3}$ from each vertex. The probability of sensing at the least-covered point is then $1 - (1 - e^{-\alpha(\frac{s}{\sqrt{3}} - r_s)})^3$ which should be greater than or equal to θ . Manipulating this inequality, we get the maximum separation $s = \sqrt{3}(r_s - \frac{\ln(1-\frac{\sqrt[3]{1-\theta}}{\alpha})}{\alpha})$. ■

To derive the maximum separation under the disk sensing model, we notice that the exponential sensing model reduces to the disk model when we set $\alpha = \infty$. From Theorem 1, it is easy to see that $s = \sqrt{3}r_s$ under the disk sensing model, which is the same optimality condition proved in [4], [14].

IV. PCP: A PROBABILISTIC COVERAGE PROTOCOL

In this section, we present our new probabilistic coverage protocol (PCP). We start with an overview of PCP where some simplifying assumptions are made to clarify the presentation. Then, we present more details on various aspects of PCP. In the following section, we prove the correctness of the protocol and analyze its complexity.

A. Overview of PCP

PCP is designed to achieve full coverage of a monitored area. This is needed in many sensor network applications,

such as forest fire detection and habitat monitoring. PCP will ensure (with probability at least θ) that each point in the area is monitored by at least one sensor. Therefore, an event (e.g., increase in air temperature) happening at any point in the area is captured by an active sensor. PCP, however, may not be suitable for applications that require a coverage degree more than one or depend on dynamic characteristics of the event. For example, in an intruder detection and classification system, multiple sensors need to detect the event in order to differentiate between different objects (e.g., person or vehicle) and to estimate the speed and direction of the object. Part of our future work is to extend PCP to support such applications.

As mentioned in the introduction, environmental conditions and other factors make the sensing ranges of sensors deviate from the perfect disk model. PCP does not assume that all sensors are deterministically identical. Rather, it uses a probabilistic distribution to model the sensing range. This probabilistic distribution accounts for variations in the sensing ranges of different sensors deployed in the monitored area.

The idea of PCP is to activate a subset of deployed sensors to construct an *approximate* triangular lattice on top of the area to be covered. The lattice is approximate because it is constructed in a distributed manner and is controlled by sensor deployment. The initial sensor deployment is not assumed to be on a lattice; it could be any distribution. In our simulations we deploy sensors uniformly at random. The maximum separations s between any pair of activated sensors is computed from the adopted probabilistic sensing model and the coverage threshold θ , as discussed in the previous section. The choice of the sensing model only impacts s . After fixing s at the appropriate value, the protocol should work the same regardless of the adopted sensing model.

To simplify the presentation, we first describe our protocol under the following assumptions. We address these assumptions in later sections.

- Single starting node. In the beginning of the protocol, only one node starts as an activator. In Section IV-C, we extend our protocol to handle multiple starting nodes.
- Nodes are time-synchronized at a coarse-grain level. In the evaluation section, we verify that only coarse-grained synchronization is needed and we study the robustness of our protocol to clock drifts. In Section IV-D, we discuss simple schemes to achieve this synchronization.
- Nodes know their locations. This is not hard to achieve in practice given efficient localization schemes such as those in [16], [17], any of them can be used with our protocol. The protocols that we compare ours against [2], [4], [8] also assume nodes know their locations. Note that our protocol does not require accurate knowledge of *global* positions, because the position information is used only in local decisions to activate nodes, as will become clear later. In the evaluation section, we analyze the robustness of our protocol to inaccuracies in node locations.
- Sensing ranges of all sensors follow the same probability distribution.

PCP works in rounds of R seconds each. R is chosen

to be much smaller than the average lifetime of sensors. In the beginning of each round, all nodes start running PCP independent of each other. A number of messages will be exchanged between nodes to determine which of them should be on duty (i.e., active) during the current round, and which should sleep till the beginning of the next round. The time it takes the protocol to determine active/sleep nodes is called the *convergence time*, and it is desired to be as small as possible. After convergence, no node changes its state and no protocol messages are exchanged till the beginning of the next round.

In PCP, a node can be in one of four states: ACTIVE, SLEEP, WAIT, or START. In the beginning of a round, each node sets its state to be START, and selects a random startup timer T_s proportional to its remaining energy level. The node with the smallest T_s will become active, and broadcasts an activation message to all nodes in its communication range. The sender of activation message is called the activator. The activation message contains the coordinates of the activator. The activation message tries to activate nodes at vertices of the hexagon centered at the activator, while putting all other nodes within that hexagon to sleep. A node receiving the activation message can determine whether it is a vertex of the hexagon by measuring the distance and angle between itself and the activator. If the angle is multiple of $\pi/3$ and the distance is s , then node sets its state to ACTIVE and it becomes a new activator. Otherwise it goes to SLEEP state.

In real deployment, nodes may not always be found at vertices of the triangular lattice because of randomness in node deployment or because of node failure. PCP tries to activate the closest nodes to hexagon vertices in a distributed manner as follows. Every node receiving an activation message calculates an activation timer T_a as a function of its closeness to the nearest vertex of the hexagon using the following equation (refer to Fig. 2): $T_a = \tau_a(d_v^2 + d_a^2\gamma^2)$, where d_v , and d_a are the Euclidean distances between the node and the vertex, and the node and the activator, respectively; γ is the angle between the line connecting the node with the activator and the line connecting the vertex with the activator; and τ_a is a constant.¹ Notice that the closer the node gets to the vertex the smaller the T_a will be. After computing T_a , a node moves to WAIT state and stays in this state till its T_a timer either expires or is canceled. When the smallest T_a timer expires, its corresponding node changes its state to ACTIVE. This node then becomes a new activator and broadcasts an activation message to its neighbors. When receiving the new activation message, nodes in WAIT state cancel their T_a timers and move to SLEEP state.

¹The intuition behind this formula is as follows. We need the activation timer T_a to rank points in terms of their deviation from the lattice vertex. For each point, the timer has to be related to the number of points with better positions. Since the number of points around the lattice vertex having the distance of less than d_v is proportional to d_v^2 , the waiting should be proportional to d_v^2 . In addition, the angle γ is between 0 and 2π while the scale of d_v can change in different applications. Therefore, γ is multiplied by the distance between sensor and the activator d_a to make it on the same scale as d_v . The number of points with better γ inside a δ -circle is proportional to γ^2 . Thus, the activation timer is formed by summation of d_v^2 and scaled angle $(d_a\gamma)^2$.

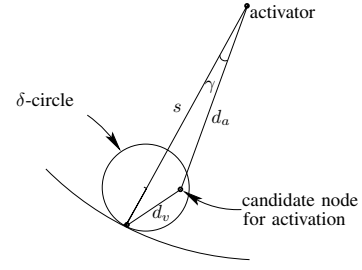


Fig. 2. Choosing the closest node to a triangle vertex.

Further optimization is possible on top of the above distributed node activation method. For this optimization, we first introduce the concept of δ -circle in the following definition.

Definition 3 (δ -circle): The smallest circle drawn anywhere in the monitored area such that there is at least one node inside it is called the δ -circle, where δ is the diameter of the circle.

The diameter δ is computed from the deployment distribution of nodes. In Section IV-B, we show how δ is computed for different deployment distributions.

Now the optimization is to minimize the number of nodes in WAIT state, that is, nodes decide quickly to be either in ACTIVE or SLEEP state. This saves energy because nodes in WAIT state must have their wireless receiving modules turned on, while all modules are turned off in SLEEP state. The savings in energy are significant as shown in the evaluation section. PCP achieves this optimization by making only nodes inside δ -circles near to the six vertices of the hexagon stay in WAIT state, all others move to SLEEP state once they determine they are outside of all δ -circles. Nodes inside δ -circles compute activation timers, as described above, to choose the closest node the vertex to be active. As shown in Fig. 2, centers of δ -circles are located at a distance of $s - \delta/2$ from the activator and at an angle that is multiple of $\pi/3$.

As a final remark, during transition between rounds, active nodes in the finished round stay active for a short period in the new round while they are participating in the protocol. This period is approximately equal to the expected convergence time. After this short period, these nodes will move to states determined by the protocol in the new round. This is done to prevent any outages in coverage during transition.

B. Computing δ -circles for Different Deployment Distributions

As mentioned in the previous section, node deployment distribution determines the value of δ , which is the diameter of the smallest circle with at least one node inside it. In this section, we compute δ for two common deployment schemes: grid and uniform distribution. δ for other schemes can be derived in a similar way. We assume that there are n nodes to be deployed on the monitored area, which is an $l \times l$ square.

For the grid distribution, nodes are deployed on a $\sqrt{n} \times \sqrt{n}$ virtual grid. The spacing between any two adjacent grid points is l/\sqrt{n} . To compute δ , consider any grid cell that is composed of four nodes forming a small square of size $l/\sqrt{n} \times l/\sqrt{n}$. Clearly, setting δ larger than the diagonal of this small square

ensures that a δ -circle drawn anywhere on the grid will contain at least one node. Therefore, $\delta = l\sqrt{2/n}$ for grid deployment.

Next we consider the case when nodes are deployed according to a uniform distribution in the range $[0, 2\lambda]$, i.e., the mean distance between adjacent nodes is λ , whereas the maximum distance does not exceed 2λ . Using a similar argument as in the grid distribution, δ should be $2\sqrt{2}\lambda$. To uniformly distribute n nodes over an $l \times l$ square, λ should be l/\sqrt{n} , which results in $\delta = 2l\sqrt{2/n}$. Note that randomness in the deployment distribution results in larger δ values.

Our PCP protocol does not require that δ to be static throughout the lifetime of the sensor network. Rather, δ can be changed to account for node failures or decreasing node density with time. For example, δ can be doubled after certain number of rounds of the protocol. This only requires that each node to keep a counter on number of elapsed rounds.

C. Multiple Starting Nodes

In Section IV-A, we assumed that PCP starts with only one node as an activator. For large-scale sensor networks, it may be desired to have multiple starting nodes such that the coverage protocol converges faster in each round. Faster convergence means that nodes move quicker from START or WAIT state to either SLEEP or ACTIVE state, which reduces the total energy consumed in the network. This is because START and WAIT are temporary states and they consume more energy than the SLEEP state. Multiple starting nodes, however, could increase the number of activated sensors because of the potential overlap between subareas that are covered due to different starting nodes. In this section, we show how PCP can be configured to enable multiple starting nodes. In the evaluation section, we study the impact of multiple starting nodes on number of activated nodes, convergence time, and total energy consumed in the network.

The number of starting nodes in a round can be controlled by setting the range of the start up timer T_s . T_s is chosen randomly between 0 and a constant τ_s . Suppose that we want to compute the value of τ_s such that each round of PCP start with k nodes on average. Let us assume that the average convergence time of PCP is T_c . Note that if the startup timer T_s of a node is less than T_c , this node will become a starting node before the protocol converges. The expected number of nodes with T_s smaller than T_c is $k = (T_c/\tau_s)n$, which yields $\tau_s = nT_c/k$. Finally, τ_s is scaled by the inverse of the normalized remaining energy level E_r ($0 < E_r \leq 1$) of each node such that nodes with higher energy levels will have higher chances for becoming starting nodes. Therefore, τ_s is set to nT_c/kE_r to allow, on average, k nodes with the highest remaining energy levels to become starting nodes. In the evaluation section, we show that our protocol consumes node energy in a uniform manner, therefore, keeps more nodes alive for longer periods and prolongs the network lifetime.

D. Time Synchronization

Our protocol requires nodes to start each round at roughly the same time. As shown in the evaluation section, the protocol

only needs coarse-grained time synchronization. Any time synchronization scheme can be used with our coverage protocol. However, the following simple scheme suffices. The first activator puts the remaining time in the current round in the activation message. When other nodes receive this activation message, they can adjust their end-of-round timers accordingly after subtracting propagation and processing delays. This process is repeated for successive activators.

V. ANALYSIS OF THE PCP PROTOCOL

In this section, we proof the correctness of our PCP protocol, and provide bounds on its convergence time, message complexity, and number of nodes activated in each round. We also prove the condition under which the activated nodes form a connected network.

Due to space limitations, the proofs of the theorems are given in the technical report [15], which is available online.

A. Correctness and Complexity Analysis

We carry out our analysis in terms of the input parameters δ, θ, s , and l , and the protocol parameter τ_a , which is the maximum value of the activation timer. δ is determined from the deployment distribution of sensors as explained in Section IV-B. The maximum separation between any two active nodes s is computed from the adopted probabilistic sensing model as explained in Section III-C. θ is the probabilistic coverage threshold, which is application dependent. l is the length of the area to be covered, which is assumed to be a square for simplicity of the analysis. We assume that the area is large compared to the sensing radius, and therefore, we ignore the boundary effects. We further assume that a message transferred between two neighboring nodes takes at most τ_m time units, which includes propagation and transmission delays.

The following theorem proves the correctness of PCP and provides an upper bound on its convergence time. PCP is considered correct if terminates with every point in the area has a probability of being sensed at least θ . Convergence time is defined as the time it takes PCP to decide for each node whether it is in ACTIVE or SLEEP state. After convergence, nodes do not change their states and no protocol messages are exchanged till the beginning of the next round.

Theorem 2 (Correctness and Convergence Time): The PCP protocol converges in at most $l(\tau_a\delta^2 + \tau_m)/(s - \delta)$ time units with every point in the area has a probability of being sensed at least θ , unless node density is not enough to achieve coverage of the whole area.

The next theorem provides upper bounds on the number of activated sensors, and number of messages exchanged by PCP in a round.

Theorem 3 (Activated Nodes and Message Complexity): The number of nodes activated by the PCP protocol is at most $l^2/\sqrt{3}(s - \delta)^2$, which is the same as the number of exchanged messages in a round.

B. Network Connectivity Analysis

So far in this paper, we have focused on the coverage problem, which ensures that an event happening at any point in the monitored area is detected. In order to transfer information gathered by a node to any other node in the network or to a processing center, there should be a communication path between any pair of nodes in the network. That is, the network should be connected.

Under the disk sensing model, previous studies [2], [4], [8] have shown that if the communication range of sensors is at least twice the sensing range and the surveillance area is convex, then coverage implies that the network is connected. These results do not hold in case of PCP, because it uses probabilistic sensing models. The following theorem provides the condition on the communication range to ensure that PCP results in a connected network of activated sensors. The theorem assumes that the communication range of nodes is a circle with radius r_c .

Theorem 4 (Network Connectivity): The subset of nodes activated by PCP will result in a connected network if the communication range of nodes r_c is greater than or equal to the maximum separation between any two active nodes s .

VI. EVALUATION

In this section, we rigorously evaluate our protocol and compare it against others in the literature. We first describe our experimental setup. Then, we verify the correctness of our protocol and validate the theoretical bounds derived in Section V. Next, we study the robustness of our protocol against node failures, inaccuracy in node locations, and clock drifts. Then, we compare our protocol against a probabilistic coverage protocol called CCANS [8]. Finally, we compare our protocol versus two recent deterministic coverage protocols: OGDC [4] and CCP [2].

Due to space limitations, only a sample of the results are presented. All figures and results are available in the technical report [15], which is available online.

A. Experimental Setup

We have implemented our PCP protocol in NS-2 [18] and in our own packet level simulator in C++. The source code for both implementations are available at [19]. Some results from the NS-2 implementation (Figs. 3(a) and 3(b)) with reasonable network sizes (up to 1000 nodes) are presented. Most results, however, are based on our own simulator because it supports much larger networks, which we need to rigorously evaluate our protocol.

We use the following parameters in the experiments, unless otherwise specified. We uniformly at random deploy 20,000 sensors over a $1km \times 1km$ area. Therefore, we have built our own packet-level simulator. We use two sensing models: The disk sensing model with a sensing range of $r_s = 15m$; and the exponential sensing model with sensing capacity decay factor $\alpha = 0.05$ and we set $r_s = 15m$ as the threshold value below which sensing is achieved with probability 1. We employ the energy model in [12] and [4], which is based on the

Mote hardware specifications. In this model, the node power consumption in transmission, reception, idle and sleep modes are 60, 12, 12, and 0.03 mWatt, respectively. The initial energy of a node is assumed to be 60 Jules, which allows a node to operate for about 5,000 seconds in reception/idle modes.

When we compare various coverage protocols, we assume that the wireless communication channel has a bandwidth of 40 kbps. Since the message sizes in all protocols are almost the same, we assume that the average message size is 34 bytes, which is the same size used in [4]. We ignore the propagation delay because it is negligible for the $1km \times 1km$ area considered in the simulation. This results in a message transmission time $\tau_m = 6.8ms$.

We repeat each experiment 10 times with different seeds, and we report the averages in all of our results. We also report the minimum and maximum values if they do not clutter figures.

B. Validation and Savings Achieved by PCP

In this section, we validate that PCP indeed achieves the requested coverage level for all points in a monitored area for deterministic as well as probabilistic sensing models. We also study the potential gain of adopting probabilistic sensing models.

Coverage and Connectivity. In the first experiment, we fix the coverage threshold θ at a specific value, run our protocol till it converges, and measure the resulting coverage in the whole area. To approximate area coverage, we measure the coverage of all points of a very dense grid deployed on top of the area. The dense grid points have spacing of $0.03r_s = 0.5m$. We conduct this experiment for different values of θ , and the results are shown in Fig. 3(a). Notice that $\theta = 1$ denotes a deterministic (disk) sensing model. The y-axis of the figure shows the fraction of the grid points meeting the coverage degree indicated on the x-axis. As the figure shows, in all cases, PCP ensured that 100% of the area is 1-covered.

In addition, we check the connectivity of the nodes activated by PCP when the communication range varies from 15 to 40m. The maximum separation s in this experiment is set to 34m. We measure connectivity as the fraction of active nodes that are connected. We plot the results in Fig. 3(b). We show the minimum, average, and maximum values obtained from the ten iterations. Confirming our analysis in Theorem 4, our protocol achieves full connectivity when $r_c \geq s$.

Savings Achieved by PCP. As mentioned in Section I, the disk sensing model may activate more than necessary nodes to ensure coverage, because it ignores sensing capacity beyond the threshold r_s . We conduct an experiment to assess the potential savings in number of active nodes because of using the (conservative) exponential sensing model instead of the disk sensing model. Fig. 3(c) shows the results for different values of the coverage threshold θ , and for a range of values for the sensing decay factor α . The figure indicates that even for a conservative value of $\alpha = 0.05$ and for $\theta = 0.99$, a saving of up to 30% in number of active nodes can be achieved, which means less energy consumed and ultimately longer lifetimes

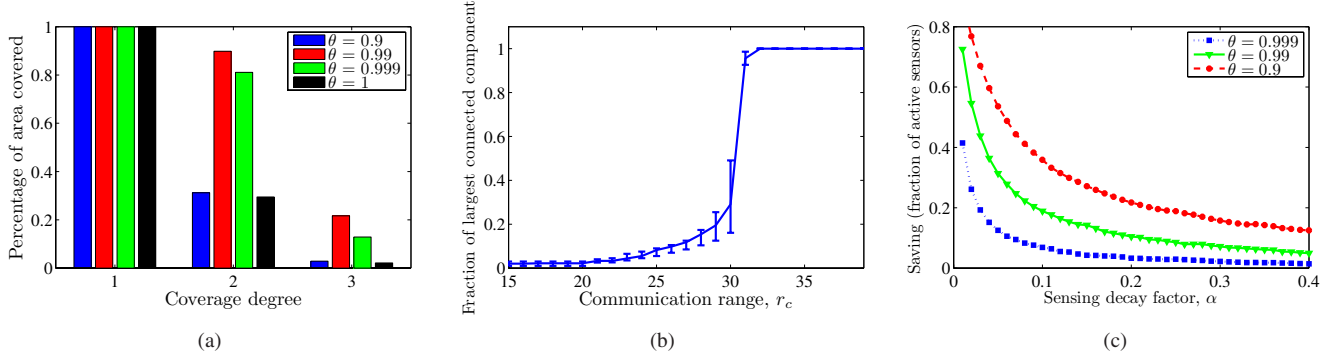


Fig. 3. Validation of the PCP protocol: (a) Achieved coverage, (b) Connectivity of active nodes, and (c) Savings in number of active nodes.

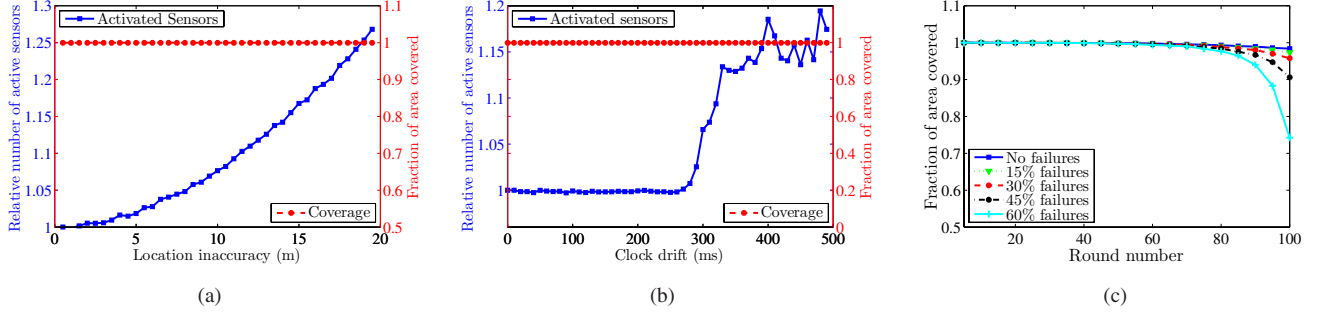


Fig. 4. Robustness of the PCP protocol against: (a) Inaccurate node locations, (b) Imperfect time synchronization, and (c) Node failures.

for the sensor network. It is expected that the savings will be higher for other probabilistic sensing models in which the sensing capacity decays slower than exponential. In addition, the savings can be increased if the coverage threshold θ is reduced, which is feasible in applications that can tolerate a small probability of not detecting an event happening at a point.

C. Robustness of PCP

In this section, we show that our protocol is robust against many practical aspects, such as inaccuracy in node locations, imperfect time synchronization, and node failures. We also show that the protocol consumes the energy of nodes in a uniform manner, and functions correctly when multiple nodes start as activators, which is important for large-scale sensor networks.

Location Inaccuracy. We use the same setup described in Section VI-A, except that we add random errors to the (x, y) coordinates of each of the 20,000 deployed nodes. The error can be positive or negative, and it is chosen randomly in the interval $[0, er_{max}]$. We vary er_{max} between 0 and 20m, that is, a node could have as much as 20m of error on any (or both) of its coordinates. For every value of er_{max} , we run our protocol till it converges, and compute the fraction of the area covered. We repeat the experiment 10 times and report the average. As shown in Fig. 4(a), PCP achieved full coverage even in presence of large location errors. This shows the robustness of PCP against location inaccuracy. There is a slight cost, though, for location inaccuracy. We compute the average number of sensors activated by the protocol to maintain coverage. We normalize this number by the number of sensors needed when

there are no location errors. The results are also shown in Fig. 4(a) (notice that some figures have two y-axes). As shown in the figure, location inaccuracy could increase the number of active sensors. This increase is not large in most practical cases: There is less than 9% increase in number of active sensors for location error of up to 10m.

Imperfect Time Synchronization. Exact, or fine-grained, time synchronization of nodes in large-scale sensor networks is costly to achieve in practice. In this experiment, we assess the impact of the granularity of time synchronization on our protocol. In our protocol, nodes need to know the start of the round so that they begin executing the protocol. Nodes will start at exactly the same time if their clocks are perfectly synchronized. We let clocks of nodes drift with different random values in the interval $[0, d_{max}]$, where d_{max} is the maximum clock drift. We vary d_{max} between 0 and 500ms. For every value of d_{max} , we run our protocol till it converges, and compute the fraction of the area covered. We repeat the experiment 10 times and report the average. As indicated by Fig. 4(b), PCP is robust against clock drifts: It achieved full coverage in all cases. In addition, for practical clock drifts (up to 300ms), there is virtually no increase in the number of activated sensors. For larger clock drifts, the cost is not significant as shown in 4(b). Notice that PCP converges in about 300ms on average. This explains why the number of active sensors starts to increase for clock drifts beyond 300ms: Some nodes with high clock drifts may start executing the protocol after others have already terminated it (i.e., they are either in SLEEP or ACTIVE states). Therefore, some of the late nodes may become unnecessarily active.

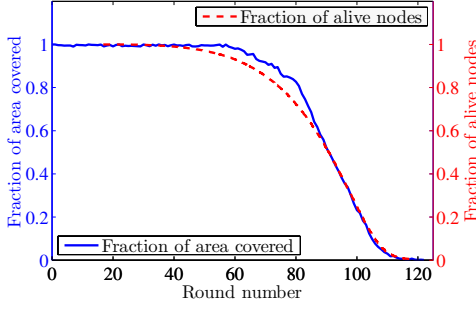


Fig. 5. Energy consumption and network lifetime under PCP.

Random Node Failures. Nodes deployed in real fields might get damaged, burned, or just fail at any time. We simulate failures at arbitrary times during the lifetime of the network. In particular, we randomly choose a fraction f of the nodes to be failed during the first 100 rounds of the protocol. We randomly schedule a failure time for each node. We change f between 0% and 60%. For each value of f , we run our protocol and we periodically check the coverage of the whole area. The results are shown in Fig. 4(c). Even with high failure rates, PCP maintained full coverage in almost all rounds.

Uniform Energy Consumption. In this experiment, we show that our protocol distributes the load uniformly across all deployed nodes. This is critical in order to keep nodes alive for the longest possible period, and thus to prolong the network lifetime and achieve more reliable coverage. We measure the load on a node by the energy consumed by that node. Once a node runs out of energy, it is assumed to be dead. We run our protocol till all nodes are dead. After each round of the protocol, we count the number of alive nodes. Again, we repeat 10 times, and we plot the average of the number of alive nodes versus round number in Fig. 5. As the figure shows, most of the nodes stay alive till round number 60. Then, they gradually die. This means that the protocol did not over utilize some nodes in early rounds, otherwise, they would have died earlier. Notice that the energy of a node is enough for it to be active in about only five rounds. In addition, Fig. 5 shows that the coverage is maintained in most of the area throughout the network lifetime.

Multiple Starting Nodes. Finally, we analyze the impact of multiple starting nodes on the performance of the PCP protocol. Multiple starting nodes are desired for large-scale networks. We change the number of starting nodes k from 1 to 9 and we study the number of sensors activated by PCP to ensure coverage normalized by the number of active sensors when $k = 1$. We also study the normalized convergence time. Our results indicate that increasing the number of starting points increases the number of active sensors but makes the protocol converges faster. From further analysis, we have found that reducing the convergence time is more beneficial for the network lifetime than reducing the number of active sensors. This is because before convergence many nodes are either in WAIT or ACTIVE state before the protocol

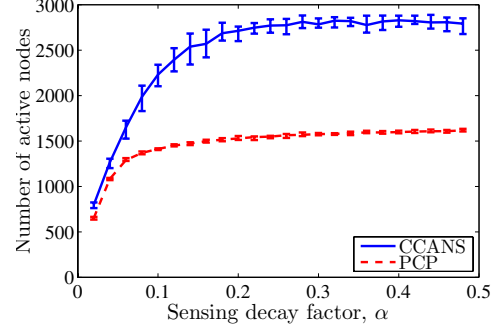


Fig. 6. Comparison between PCP and CCANS. For both simulations, we show the minimum, average, and maximum values.

converges, which consume more energy.

D. Comparing PCP versus another Probabilistic Coverage Protocol (CCANS)

In this section, we compare our PCP protocol against the probabilistic coverage CCANS, proposed in [8], in terms of number of activated sensors, network lifetime, and energy consumption. CCANS employs the exponential sensing model. The idea of CCANS is to start all nodes in active mode then iteratively deactivate nodes that are not needed for coverage. A token is circulated among nodes in the network in a certain manner. The node holding the token calculates the coverage on the grid points around it. If coverage is achieved at these points, it broadcasts a notification to its neighbors, passes the token to another node, and deactivates itself. All redundant nodes are deactivated when the token visits each node in the network. We implemented CCANS in C++, and we validated our implementation of CCANS by obtaining the same results in [8].

In our comparison, we use the same sensing model for PCP and CCANS. The parameters used for CCANS [8] are: $\xi_{th} = 1$ and $t_{max} = \tau_m$. The parameters used for PCP are: $\tau_a = \tau_m / \delta^2$, and $\tau_s = n / E_r$, where E_r is the fraction of the remaining energy in the node. δ is computed from IV-B. For a uniform distribution of 20,000 nodes in a $1km \times 1km$ area we have $\delta = 2 \times 1000 \sqrt{2/20,000} = 20m$.

We plot in Fig. 6 the average number of nodes activated by PCP and CCANS for different values of the sensing decay factor α and the coverage threshold θ . As the figure shows, PCP activates a much smaller number of nodes than CCANS, while ensuring the same level of probabilistic coverage. This is significant because it indicates that the sensor network could last much longer using our protocol. To validate this claim, we study the fraction of the remaining energy in nodes as the time progresses from 0 to 1000 seconds. Our results show that, because CCANS activates more nodes and exchanges more messages than PCP, the node energy is depleted at a much faster rate. For example, after 1000 seconds, the average energy of a node is 60% of its original energy if the sensor network uses CCANS to maintain coverage, while this average is 90% if our PCP protocol is used.

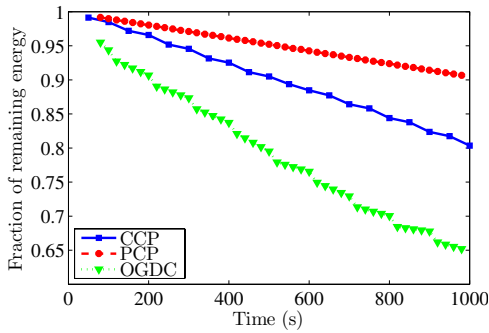


Fig. 7. Comparison between PCP, OGDC, and CCP.

E. Comparing PCP versus other Deterministic Coverage Protocols (OGDC and CCP)

This section shows that, in addition to its use as a probabilistic coverage protocol, PCP can be used as an efficient deterministic coverage protocol that outperforms previous deterministic coverage protocols.

We have implemented two recent coverage protocols: OGDC [4] and CCP [2] that were shown to outperform others in the literature. Both protocols are implemented in C++. We validated our implementation of OGDC and CCP by obtaining the same results in their respective paper. We use the disk sensing model for all protocols. To conduct a fair comparison and remove the overhead imposed by CCP and OGDC to maintain connectivity, we assume that the communication range is twice as the sensing range in all experiments for all protocols. The round length is 100 seconds for both PCP and OGDC. We set the parameters p_0 in OGDC and τ_s in PCP such that both protocols have a single starting node.

We focus our comparison on the energy consumption of deployed nodes under different coverage protocols. In Fig. 7, we plot the fraction of remaining energy in nodes as the time progresses. The figure shows that our PCP protocol is much more energy conserving than CCP and OGDC. This is because our protocol activates fewer number of nodes, converges faster, and exchanges fewer number of messages than CCP and OGDC.

VII. CONCLUSION

In this paper, we considered coverage problems under probabilistic sensing models, which are more realistic than the disk sensing model used in many of the previous works. We showed through simulation that a probabilistic sensing model may result in significant savings in the number of activated sensors, which reduces energy consumption and extends the network lifetime. At a high-level, our results advocate the use of probabilistic sensing models because of the potential savings. We proposed and evaluated a fully distributed, probabilistic coverage protocol. A key feature of our protocol is that it can be used with different sensing models, with minimal changes. We analyzed our protocol and showed that it converges fast and has a small message complexity. We verified our analytical results using simulations. We also implemented our protocol

and three other coverage protocols: one of them is probabilistic (CCANS) and the other two are deterministic (OGDC and CCP). Our extensive experimental study shows that our protocol activates less sensors than the others while maintaining the same level of coverage, and consumes much less energy.

ACKNOWLEDGMENT

This work is partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Discovery Grant #313083 and RTI Grant #344619.

REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.
- [2] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 36–72, August 2005.
- [3] Z. Zhou, S. Das, and H. Gupta, "Connected k-coverage problem in sensor networks," in *Proc. of ICCCN'04*, Chicago, IL, October 2004, pp. 373–378.
- [4] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc and Sensor Wireless Networks: An International Journal*, vol. 1, no. 1-2, pp. 89–123, January 2005.
- [5] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity, and diameter," *Ad Hoc Networks*, vol. 3, no. 6, pp. 702–716, November 2005.
- [6] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proc. of ACM MOBIKOM'04*, Philadelphia, PA, September 2004, pp. 144–158.
- [7] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, February 2004.
- [8] —, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 978–991, August 2005.
- [9] N. Ahmed, S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in *Proc. of IEEE Conference on Local Computer Networks (LCN'05)*, Sydney, Australia, November 2005, pp. 672–681.
- [10] B. Liu and D. Towsley, "A study on the coverage of large-scale sensor networks," in *Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*, Fort Lauderdale, Florida, October 2004, pp. 475–483.
- [11] Q. Cao, T. Yan, T. Abdelzaher, and J. Stankovic, "Analysis of target detection performance for wireless sensor networks," in *Proc. of International Conference on Distributed Computing in Sensor Networks*, Marina Del Rey, CA, June 2005, pp. 276–292.
- [12] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," in *Proc. of International Conference on Distributed Computing Systems (ICDCS'03)*, Providence, RI, May 2003, pp. 28–37.
- [13] D. Tian and N. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. of First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002, pp. 32–41.
- [14] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proc. of ACM MobiHoc'06*, Florence, Italy, May 2006, pp. 131–142.
- [15] M. Hefeeda and H. Ahmadi, "Probabilistic coverage in wireless sensor networks," Simon Fraser University, TR 2006-21, Tech. Rep., Updated March 2007, available at: <http://www.cs.sfu.ca/~mhfeeda/>.
- [16] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of ACM MOBIKOM'01*, Rome, Italy, July 2001, pp. 166–179.
- [17] L. Doherty, L. E. Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," in *Proc. of IEEE INFOCOM'01*, Anchorage, AK, April 2001, pp. 1655–1663.
- [18] NS-2 Web Page, "<http://nsnam.isi.edu/nsnam/>"
- [19] Network Systems Lab Web Page, "<http://nsl.cs.sfu.ca/>"