

Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks

Mohamed Hefeeda, *Senior Member, IEEE*, and Hossein Ahmadi

Abstract—Various sensor types, e.g., temperature, humidity, and acoustic, sense physical phenomena in different ways, and thus, are expected to have different sensing models. Even for the same sensor type, the sensing model may need to be changed in different environments. Designing and testing a different coverage protocol for each sensing model is indeed a costly task. To address this challenging task, we propose a new probabilistic coverage protocol (denoted by PCP) that could employ different sensing models. We show that PCP works with the common disk sensing model as well as probabilistic sensing models, with minimal changes. We analyze the complexity of PCP and prove its correctness. In addition, we conduct an extensive simulation study of large-scale sensor networks to rigorously evaluate PCP and compare it against other deterministic and probabilistic protocols in the literature. Our simulation demonstrates that PCP is robust, and it can function correctly in presence of random node failures, inaccuracies in node locations, and imperfect time synchronization of nodes. Our comparisons with other protocols indicate that PCP outperforms them in several aspects, including number of activated sensors, total energy consumed, and network lifetime.

Index Terms—Sensor networks, coverage in sensor networks, probabilistic coverage, coverage protocols.

1 INTRODUCTION

MANY real-life applications have been proposed for wireless sensor networks, including forest fire detection, area surveillance, and natural habitat monitoring [1]. Two of the important research problems in such sensor network applications are ensuring area coverage and maintaining the connectivity of the network. Area coverage means that nodes use their sensing modules to detect events occurring in the monitored area. Each sensor is assumed to have a *sensing range*, which depends on the phenomenon being sensed and the environment conditions. Maintaining the sensor network connected is also important because information about the detected events may need to be communicated to processing centers for possible actions. Connectivity is achieved by the radio communication modules installed in the sensors. Each sensor is assumed to have a communication range, which is totally different from the sensing range in general.

This paper presents a new efficient and general coverage protocol, which also considers the network connectivity. The proposed protocol is efficient because it reduces the energy consumed by sensor nodes and prolongs the network lifetime. Energy efficiency is critical for successful deployment and operation of large-scale sensor networks that are typically battery-powered. The protocol is general because it can employ different deterministic and probabilistic models

for the sensing ranges of sensors, with minimal changes in its operation. The generality is important in real-life applications, since different sensor types 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. Thus, the generality of the proposed protocol enables the designers of sensor networks to avoid the costly and complex tasks of designing, implementing, and testing a different coverage protocol for each sensing model.

The proposed protocol, called Probabilistic Coverage Protocol (PCP), works for the disk sensing model used in many of the previous works in the literature, e.g., [2], [3], [4], [5], [6], [7], [8]. This model, depicted in Fig. 1a, assumes that the sensing range is a uniform disk of radius r_s . The simple disk sensing model is appealing because it makes coverage maintenance protocols, e.g., [2], [3], [4], less complicated to design and analyze. It also makes analytical and asymptotic analysis, e.g., [5], [6], tractable. However, it may not be realistic in some environments or it could become too conservative in modeling the sensing range of some sensors. Therefore, better models for sensing ranges may be needed in some sensor network applications.

Several studies [9], [10], [11], [12], [13] 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 [13] 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 [9], [10] use an exponential sensing model, where the sensing capacity degrades according to an exponential distribution after a certain threshold, as shown in Fig. 1b. Whereas the authors of [12] propose a polynomial function to model the probabilistic nature of the sensing range, as shown in Fig. 1d. Furthermore, the authors of [11] assume that the sensing

• M. Hefeeda is with the School of Computing Science, Simon Fraser University, 250-13450 102nd Ave., Surrey, BC V3T 0A3, Canada. E-mail: mhefeeda@cs.sfu.ca.

• H. Ahmadi is with the Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801. E-mail: hahmadi2@uiuc.edu.

Manuscript received 16 Jan. 2009; revised 21 June 2009; accepted 26 June 2009; published online 1 July 2009.

Recommended for acceptance by Y.-C. Tseng.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-01-0027. Digital Object Identifier no. 10.1109/TPDS.2009.112.

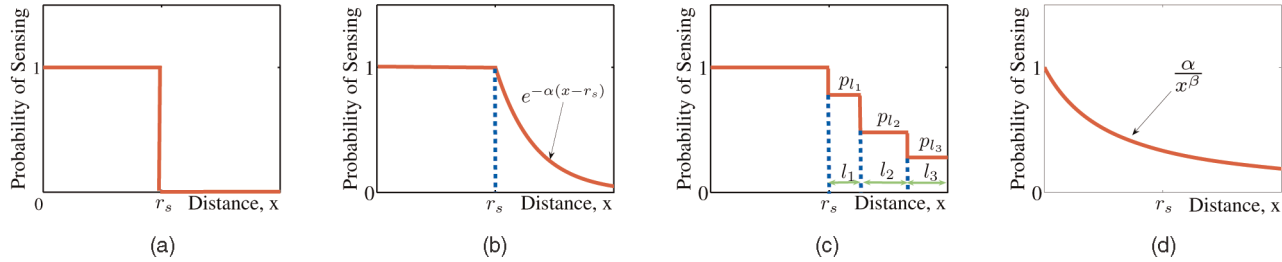


Fig. 1. Some of the sensing models used in the literature. (a) Disk model. (b) Exponential model in [9], [10]. (c) Staircase model in [11]. (d) Model in [12].

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. 1c.

To support probabilistic sensing models, we introduce the notion of probabilistic coverage of a target area with a given threshold θ , which means that an area is considered covered if the probability of sensing an event occurring at any point in the area is at least θ . This notion not only enables probabilistic sensing models but also provides a *coverage-network lifetime* control knob, which is θ . This knob allows the sensor network operators to trade off the coverage level for longer network lifetimes, if the applications using the sensor network could tolerate this trade-off. For example, reducing θ makes the coverage protocol activate fewer sensors to monitor the area, and hence, saves energy and extends the network lifetime. We show that the proposed protocol can work with various probabilistic sensing models. In particular, our protocol 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 calculation for the exponential sensing model [9], [10]. This model is chosen because it is conservative in terms of estimating the sensing capacity, and it has been used before in another probabilistic coverage protocol (CCANS [10]). 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.

We analyze the complexity of the proposed protocol and prove its correctness. We also derive the condition under which the protocol maintains the connectivity of the network in addition to covering the monitored area. Furthermore, we conduct an extensive simulation study of large-scale sensor networks to rigorously evaluate our protocol and compare it against other deterministic and probabilistic protocols in the literature. Our simulation demonstrates that the proposed protocol is robust, and it can function correctly in presence of random node failures, inaccuracies in node locations, and imperfect time synchronization of nodes. Our comparisons with other protocols indicate that our protocol outperforms them in several aspects, including number of activated sensors, total energy consumed, and network lifetime.

The rest of the paper is organized as follows: We summarize the related work in Section 2. In Section 3, we present the proposed protocol in the context of the disk sensing model because it is easier. In Section 4, we formally

define the probabilistic coverage problem and show how our protocol can solve it. In Section 5, we prove the correctness of the protocol and provide bounds on its convergence time and message complexity. We also prove the condition on the communication range needed for our coverage protocol to provide network connectivity as well. In Section 6, we evaluate our protocol and compare it against others using simulation. We conclude the paper in Section 7.

2 RELATED WORK

Coverage in sensor networks has received significant research attention, see [14] for a survey. We summarize the most relevant works to ours in the following sections.

2.1 1-Coverage Using the Disk Sensing Model

The studies in [5], [6] conduct asymptotic and analytical analyses to provide necessary and sufficient conditions for coverage in various environments. While these studies provide useful insights and guidelines, which we indeed benefited from, they do not propose specific coverage protocols. In [15], optimal deployment patterns for different ratios of the communication and sensing ranges are proposed. Exact sensor placement is difficult, if at all possible, in many realistic environments such as battlefields and forests. In addition, Bai et al. [15] do not present dynamic protocols to maintain coverage as the time passes and some sensors fail.

Several distributed coverage protocols have been proposed for the disk model, including [2], [4], [7], [8], [16], [17], [18]. 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. CCP can provide coverage with degrees higher than 1 as well. Sensors in PEAS [7] probe their neighbors to decide whether to be in active or sleep mode. The coverage algorithms in [16] solve a variation of the set k -cover problem, where sensors are partitioned into k covers and individual covers are iteratively activated to achieve 1-coverage of the monitored area. The authors of [17] propose three node scheduling schemes that estimate the distance to the nearest neighbor, number of neighbors, or the probability of a node being off duty and use one of these metrics to put some sensors in sleep mode. The coverage algorithm in [18] tries to find uncovered spots and activate sensors in these areas using information from nearby active sensors.

All of the above protocols assume the disk sensing model. Our protocol, in contrast, is general and can adopt

the disk model as well as probabilistic models. To show this generality, 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.

The early work [19] considers a slightly different definition of coverage: finding maximal support and maximal breach paths for which the observability is maximum and minimum, respectively. The authors of [21] improve the work in [19] and present a more efficient algorithm. Furthermore, Li et al. [21] present a distributed algorithm for the maximal support path. We consider a different problem: area coverage. Thus, these algorithms are not comparable to ours.

2.2 Coverage Using Probabilistic Sensing Models

Probabilistic coverage with various sensing models has also been studied in [10], [11], [12], [22]. The work in [12] analytically studies the implications of adopting probabilistic and disk sensing models on coverage. The study in [22] presents closed-form equations for computing the probability of any point in the area to be covered given N deployed sensors, which could be heterogeneous and may not necessarily follow the disk sensing model. These studies do not propose specific coverage protocols. In [11], 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 against theirs. The closest work to ours is [10], where the authors design a probabilistic coverage protocol called CCANS. A brief description of CCANS is presented in Section 6.4. We compare our protocol against CCANS.

2.3 k -Coverage and Network Connectivity

Coverage with various degrees (k -coverage), where every point is sensed by at least k sensors, has also been studied, see the survey in [14]. The problem of *verifying* k -coverage is studied in [23]. Each sensor is modeled as a disk and it is proved that the area is k -covered if the perimeter of all disks is k -covered. An improved modeling is presented in [24], where the authors use the concept of order- k Voronoi diagrams [25] to build a verifier algorithm. In [26], the authors first propose a k -coverage determination algorithm and then present a distributed sleep control protocol to achieve k -coverage by exchanging several types of messages. In [27], the authors formulate the k -coverage problem of a set of n grid points as an integer linear programming to determine the minimum cost of sensors to cover all grid points.

In [28], the authors address the problem of selecting the minimum number of sensors to activate to achieve k -coverage, which is shown to be NP-hard. The authors present a distributed algorithm, which uses a pruning method similar to the algorithms used for constructing connected dominating sets, e.g., [29]: nodes are assigned unique priorities and they broadcast their neighbor set information. Then each node can go to a sleep mode by checking whether the coverage and connectivity can be maintained by other higher priority nodes in its neighbor-

hood. The work in [3] presents two distributed k -coverage algorithms. The first one is a distributed greedy algorithm, which requires carrying around a central state. The second algorithm, called distributed priority algorithm (DPA), is localized and more robust. DPA, which is also used in [30] with some modifications to activate a minimal subset of sensors to answer a query, employs multihop neighborhood information to turn off nodes that are not needed to k -cover the area. A more recent k -coverage algorithm was presented in our previous work [31].

Because of the hardness of the problem, most of these works assume the disk sensing model; the concept of k -coverage under probabilistic sensing models is not yet well defined. In this paper, we focus on 1-coverage with probabilistic sensing models and leave the extension to the k -coverage case for future work.

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 proved 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], [10]. In this paper, we prove the conditions under which probabilistic coverage ensures 1-connectivity.

3 PCP WITH DISK SENSING MODEL

In this section, we present our new PCP, in the context of the disk sensing model because it is simpler.

3.1 Overview of PCP

It has been shown before, e.g., in [15], that 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 $\sqrt{3}r_s$. Optimality here means the minimum number of disks required. 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. PCP starts by activating any sensor in the area, which we refer to as an activator. 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. As illustrated in Fig. 2, this process continues till the activated sensors form a virtual triangular lattice over the whole area.

We refer to the distance between the vertices of the triangular lattice as the maximum separation between active nodes, and it is denoted by s . The value of s is determined from the sensing range r_s of sensors. Under the disk sensing model, the maximum separation is set to $s = \sqrt{3}r_s$. 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 above description is idealistic and makes several assumptions. We list these assumptions below and describe how we address them. We also note that a similar idea of activating sensors on a triangular lattice was used by the OGDC protocol [4]. However, the optimization proposed

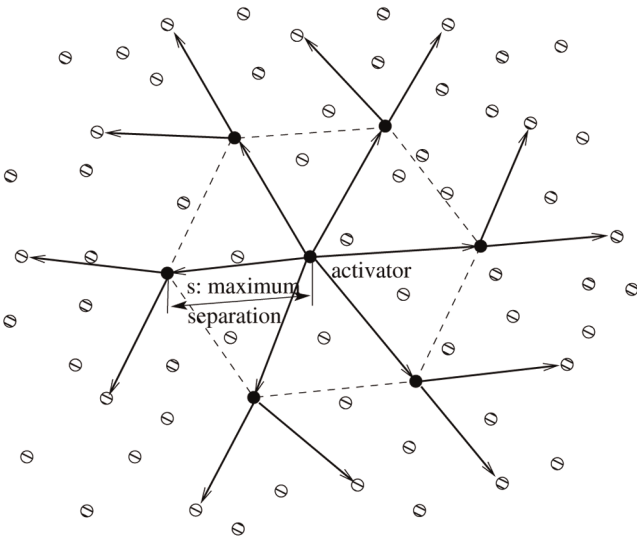


Fig. 2. A simplification of the node activation process in PCP. Activated nodes try to form a triangular lattice over the area.

later in Section 3.2 is new and yields significant performance improvement as shown in Section 6. In addition, the authors of [4] do not address probabilistic sensing models, as we do in Section 4.

- Single starting node. In the beginning of the protocol, only one node starts as an activator. In Section 3.3, we extend our protocol to handle multiple starting nodes, which is important for large-scale sensor networks.
- Nodes are time synchronized at a coarse-grained level. In Section 6, we verify that only coarse-grained synchronization is needed and we study the robustness of our protocol to clock drifts.
- Nodes know their locations. This is not hard to achieve in practice given efficient localization schemes such as those in [32], [33], any of them can be used with our protocol. The protocols that we compare ours against [2], [4], [10] also assume that 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 Section 6, we analyze the robustness of our protocol to inaccuracies in node location information.

We now present some details of the proposed protocol. 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 the 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 start-up timer T_s inversely proportional to its remaining energy level. The node with the smallest T_s will become active and broadcast 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, and it 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. The angle is measured starting from the positive x-axis and going anticlockwise. If the angle is multiple of $\pi/3$ and the distance is s , then the 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:

$$T_a = \tau_a (d_v^2 + d_a^2 \gamma^2), \quad (1)$$

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 Note 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.

3.2 Optimization Using δ -Circles

We propose a new optimization of the PCP protocol. This optimization puts more sensors in sleep mode faster, shortens the protocol convergence time, and thus, saves more energy. We first introduce the concept of δ -circle in the following definition:

Definition 1 (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.*

1. The 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 distance 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 the summation of d_v^2 and the scaled angle $(d_a \gamma)^2$.

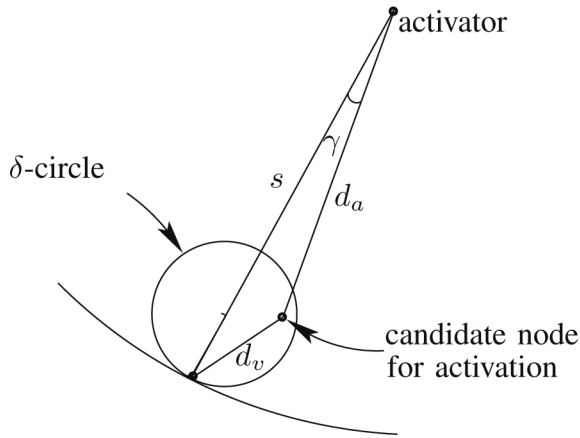


Fig. 3. Choosing the closest node to a triangle vertex.

The diameter δ is computed from the deployment distribution of nodes. 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 $l/(\sqrt{n} - 1)$. To compute δ , consider any grid cell that is composed of four nodes forming a small square of size $l/(\sqrt{n} - 1) \times l/(\sqrt{n} - 1)$. 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, $\delta = \sqrt{2}l/(\sqrt{n} - 1)$ for grid deployment. Next, we consider the case when nodes are deployed uniformly at random 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 $l/(\sqrt{n} - 1)$, which results in $\delta = 2\sqrt{2}l/(\sqrt{n} - 1)$. Note that randomness in the deployment distribution results in larger δ values.

We employ the δ -circle concept 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 Section 6. 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 that 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. Fig. 3 shows one of the six δ -circles of a given activator. Note that the centers of the δ -circles are located at a distance of $s - \delta/2$ from the activator and at angles that are multiple of $\pi/3$. The state diagram of the PCP protocol is illustrated in Fig. 4. The figure shows the status of the sensing, sending, and receiving modules in each state of the node.

Note that the 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

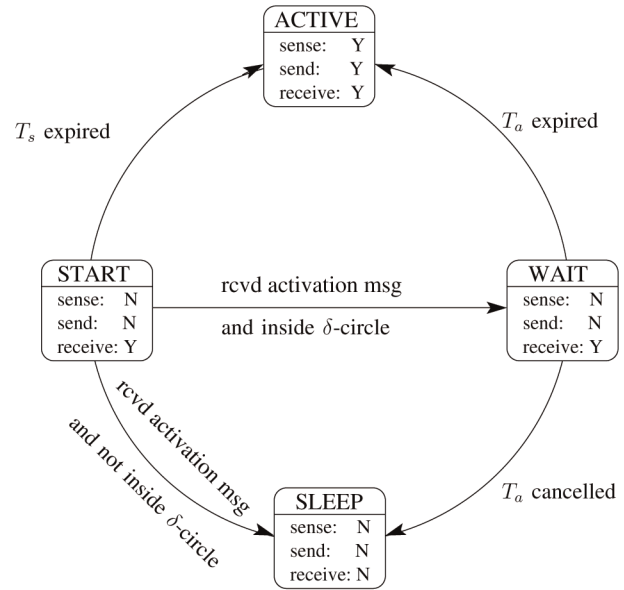


Fig. 4. The state diagram of the PCP protocol. In each state, we mark which of the sensing, sending, and receiving modules is on.

decreasing node density with the time. For example, δ can be doubled after certain number of rounds of the protocol. This only requires each node to keep a counter on the number of elapsed rounds. Also note that 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.

3.3 Multiple Starting Nodes

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. 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.

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 starts with k nodes, on average. Let us assume that the average convergence time of PCP is T_c . Note that if the start-up 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 $\tau_s = nT_c/k$. In Section 6, we verify that our protocol consumes the energy of nodes in a uniform manner, therefore, it keeps more nodes alive for longer periods and prolongs the network lifetime. We also study the impact of multiple starting nodes on the number

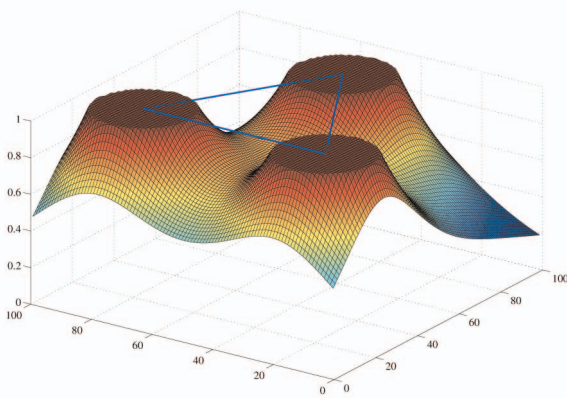


Fig. 5. The sensing capacity of three sensors that use the exponential sensing model and deployed at vertices of an equilateral triangle. The least-covered point by these three sensors is at the center of the triangle.

of activated nodes, convergence time, and total energy consumed in the network.

4 PCP WITH PROBABILISTIC SENSING MODELS

In this section, we define the notion of probabilistic coverage and extend the proposed PCP protocol to support probabilistic sensing models.

4.1 Definitions and Overview

We start by stating the following two definitions:

Definition 2 (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 .

$P(x)$ in the above definition measures the collective probability from all 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 3 (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 .

Fig. 5 demonstrates the concept of the least-covered point by showing the sensing capacity of three nodes deployed on an equilateral triangle. The three sensors use the exponential sensing model in Fig. 1b.

The proposed PCP protocol works again by building an approximate triangular lattice in a distributed manner, as explained in Section 3. However, in the probabilistic case, PCP ensures that the least-covered point in the monitored area has a probability of being sensed that is at least θ . In order for PCP to provide this assurance, we need to compute the maximum separation s between any two active

nodes on the triangular lattice. Once we compute s , PCP functions in the same manner as described in Section 3.

Note that computing s depends only on the sensing model used. In the following section, we derive s for the exponential sensing model [9], [10] as an example. Computing s for other sensing models can be done in a similar way. We should emphasize that the operation of the PCP protocol 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 .

4.2 Computing Maximum Separation

This section presents the details of computing the maximum separation s between any two active nodes for the exponential sensing model, which is defined as

$$p(d) = \begin{cases} 1, & \text{for } d \leq r_s, \\ e^{-\alpha(d-r_s)}, & \text{for } d > r_s, \end{cases} \quad (2)$$

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 in Fig. 1b. We consider this sensing model for two reasons. First, it has been adopted before in [9], [10], which allows us to conduct a fair comparison between our protocol and the protocol in [10]. 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 [11], [12], [13]. 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:

Theorem 1 (maximum separation). Under the exponential sensing model defined in (2), 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 $\sqrt{3}(r_s - \frac{\ln(1-\sqrt[3]{1-\theta})}{\alpha})$.

Proof. To prove this theorem, we need to find the location of the least-covered point. We prove in the Appendix that this location is at a distance of $s/\sqrt{3}$ from each vertex of the equilateral triangle. 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 . By manipulating this inequality, we get the maximum separation $s = \sqrt{3}(r_s - \frac{\ln(1-\sqrt[3]{1-\theta})}{\alpha})$. \square

Note 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 value used by our protocol in Section 3 and the same as the optimality condition proved in [4], [15].

Remark. Note that as the coverage threshold θ decreases, the maximum separation between activated sensors increases. Therefore, a fewer number of active sensors will be needed to maintain the coverage of the monitored area. This implies that the lifetime of the sensor network could be extended by reducing θ if the application can tolerate this reduction. Therefore, the coverage threshold θ can be used as a control knob, which can trade off the coverage quality for longer network lifetimes. In Section 6, we assess the savings in the number of active sensors for different values of θ .

5 ANALYSIS OF THE PCP PROTOCOL

In this section, we first state and prove the condition under which the activated nodes by the PCP protocol form a connected network. Then, we prove the correctness of the PCP protocol and provide bounds on its convergence time, message complexity, and number of nodes activated in each round. All theorems in this section are validated using simulation in Section 6.

5.1 Network Connectivity Analysis

Under the disk sensing model, previous studies [2], [4], [10] 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 may not hold in the 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 2 (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 , where s is computed from the sensing model.*

Proof. First, we prove that the subset of nodes activated by PCP is connected when there is a single starting node in each round. We use induction in the proof. Initially, we have one node activated, which is connected. Suppose at step k , we have a connected subset A_k of active nodes formed after k steps of sending activation messages. We show by contradiction that the subset A_{k+1} constructed in step $k+1$ is also connected. Suppose that A_{k+1} is not connected. Since A_k is connected, there are some nodes (denoted by the set V) that are activated in step $k+1$ and not connected to A_k . Consider any $v \in V$. v must have been activated by an activator (say u) in A_k because v is activated in step $k+1$. Since v is at a distance of at most s from u , v is reachable from u because $r_c \geq s$. Since v is chosen arbitrarily from V , all nodes in V are reachable from A_k . That is, A_{k+1} is connected, which contradicts the assumption.

Second, we consider the case for multiple starting nodes. From the previous case, we know that each starting

node creates a connected subset of activated nodes. Thus, we need to prove that the union of subsets activated by different starting nodes is also connected. We prove this by contradiction. Consider any two connected subsets A and A' that are activated by two different activators. Let $u \in A$ and $v \in A'$ be the nearest nodes in the two subsets. Assume that the PCP protocol terminates and the network is not connected, i.e., A is disconnected from A' . Thus, the distance between u and v is more than their communication range: $\text{dist}(u, v) > r_c$. Since the protocol has terminated, there is no node in the WAIT state. Therefore, there are six activated neighbors of u with a distance at most s ; otherwise, some nodes around u are still in WAIT state. Let u' be the neighbor with the least distance to v . We identify two cases as follows:

1. $u' \in A'$. Since $\text{dist}(u, u') \leq s$ and $\text{dist}(u, v) > r_c \geq s$, we have $\text{dist}(u, v) > \text{dist}(u, u')$. Thus, $u' \in A'$ is closer to $u \in A$ than $v \in A'$. This is a contradiction because u and v are assumed to be the closest nodes in A and A' .
2. $u' \in A$. Consider the triangle $uu'v$, and recall that any triangle has the following property:

$$\begin{aligned} \text{dist}(u', v)^2 &= \text{dist}(u, v)^2 + \text{dist}(u, u')^2 \\ &\quad - 2 \cos(u'uv) \text{dist}(u, v) \text{dist}(u, u'). \end{aligned}$$

Since $\text{dist}(u, u') \leq s$, we have

$$\begin{aligned} \text{dist}(u', v)^2 &\leq \text{dist}(u, v)^2 + s^2 \\ &\quad - 2 \cos(u'uv) \text{dist}(u, v) s. \end{aligned}$$

The angle between lines uv and uu' , called $u'uv$, is less than or equal to 60 degrees. Otherwise, there is another neighbor of u nearer than u' to v . Therefore, $\cos(u'uv) \geq 0.5$ and $\text{dist}(u', v)^2 \leq \text{dist}(u, v)^2 + s(s - \text{dist}(u, v))$. Since

$$\text{dist}(u, v) > r_c \geq s,$$

we have $s(s - \text{dist}(u, v)) < 0$. Therefore,

$$\text{dist}(u', v) < \text{dist}(u, v).$$

In other words, u' is closer to v than u , which is a contradiction. \square

5.2 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 3.2. The maximum separation between any two active nodes s is computed from the adopted probabilistic sensing model as explained in Section 4.2. 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 . After convergence, nodes do not change their states and no protocol messages are exchanged till the beginning of the next round:

Theorem 3 (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 .*

Proof. First, we prove the correctness part. PCP incrementally constructs a triangular lattice of active nodes. This triangular lattice will eventually cover the whole area because each node begins a round with setting a start-up timer T_s , and if T_s expires, the node becomes active (i.e., it will be a vertex of a triangle). The T_s timer of a node n_1 can be canceled only if another node n_2 has become active and n_2 is at a vertex of the triangle that contains n_1 . Now we need to show that each triangle of the lattice is covered. Consider any triangle. Since nodes activated by an activator are at a distance of at most s from the activator, the triangle formed by activated nodes will have side lengths of at most s . Recall that s is computed from the sensing model to ensure that the coverage probability at the least-covered point in a triangle is at least θ . Therefore, the coverage probability in the whole triangle is at least θ .

Second, we bound the convergence time. Within each round, PCP runs in steps. In each step, an activation message is sent, and at least one node is activated in each of the six directions. Consider one direction. In the worst case, the newly activated node is at a distance of $s - \delta$ from the old node. Thus, in the worst case, PCP needs $l/(s - \delta)$ steps if the first activated node is at the border. The maximum time to complete one step occurs when the node chosen to be active happens to have the largest value for the activation timer T_a , which is $\tau_a\delta^2$ (computed from (1)). Adding the message transmission time τ_m to the maximum value of the activation timer yields a worst-case time for any step as $(\tau_a\delta^2 + \tau_m)$. Multiplying this value by the number of steps $l/(s - \delta)$ yields the worst-case convergence time of PCP. \square

The following theorem provides upper bounds on the number of activated sensors and number of messages exchanged by PCP in a round:

Theorem 4 (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.*

Proof. The number of nodes to cover an $l \times l$ area is equal to the number of vertices of a triangular lattice with spacing s . This number is $l^2/\sqrt{3}s^2$ and computed as follows: Since the area of an equilateral triangle with side s is $s^2\sqrt{3}/2$ and the triangles completely tile the area, the total number of triangles required is $2l^2/\sqrt{3}s^2$. Since there are three nodes used in each triangle and each node is also used in six different triangles, the total number of nodes is

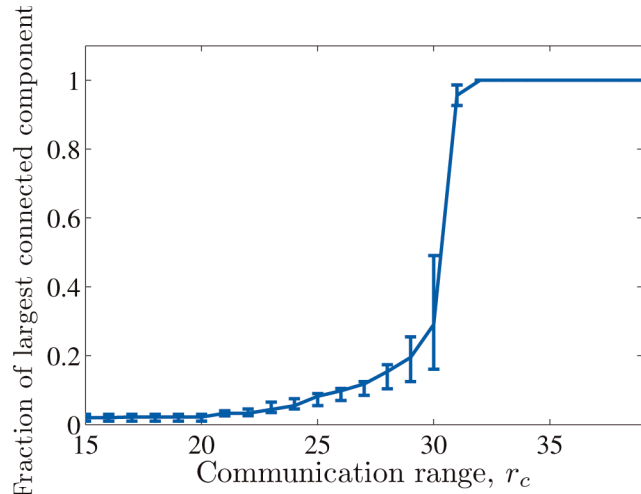


Fig. 6. Connectivity among nodes activated by PCP.

$3/6 \times 2l^2/\sqrt{3}s^2 = l^2/\sqrt{3}s^2$. The number of nodes activated by PCP is computed in a similar way, but with a triangular lattice with spacing at most $s - \delta$. Thus, the number of activated nodes by PCP is at most $l^2/\sqrt{3}(s - \delta)^2$.

For message complexity, we notice that there is only one message sent by each activated node. Thus, the total number of messages sent in a round is equal to the number of activated sensors. \square

6 EVALUATION

In this section, we evaluate our protocol and compare it against others in the literature using extensive simulations. We first describe our experimental setup. Then, we verify the correctness of our protocol and validate the theoretical bounds derived in Section 5. Next, we study the robustness of our protocol against node failures, inaccuracy in nodes location information, and clock drifts. Then, we compare our protocol against a probabilistic coverage protocol called CCANS [10]. Finally, we compare our protocol versus two recent deterministic coverage protocols: OGDC [4] and CCP [2].

6.1 Experimental Setup

We have implemented our PCP protocol in NS-2 [34] and in our own packet-level simulator in C++. The source code for both implementations is available at [35]. Some results from the NS-2 implementation (Fig. 6) with reasonable network sizes (up to 1,000 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 $1 \text{ km} \times 1 \text{ km}$ area. We use two sensing models: The disk sensing model with a sensing range of $r_s = 15 \text{ m}$ and the exponential sensing model with sensing capacity decay factor $\alpha = 0.05$, and we set $r_s = 15 \text{ m}$ as the threshold value below which sensing is achieved with probability 1. We employ the energy model in [7] and [4], which is based on the Mote hardware

specifications. In this model, the node power consumption in transmission, reception, idle, and sleep modes is 60, 12, 12, and 0.03 mW, respectively. The initial energy of a node is assumed to be 60 Joules, 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 1 km \times 1 km area considered in the simulation. This results in a message transmission time $\tau_m = 6.8$ ms.

We repeat each experiment 10 times with different seeds and report the averages in all of our results. We also report the minimum and maximum values if they do not clutter the figures. Note that the simulated sensor network in each experiment replica has 20,000 nodes, and the measured statistics are collected from all of them. Therefore, we believe that combining the data from 10 different replicas and each with 20,000 nodes yields statistically significant results (we did not see large variances in our results). Finally, we mention that in most experiments, each single replica took several hours of running time on a decent multicore Linux server. Furthermore, processing the huge traces created in these large-scale experiments consumed many CPU hours.

6.2 Validation and Savings Achieved by PCP

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. A point is considered covered if the probability of sensing at this point is at least θ . The dense grid points have spacing of $0.03r_s = 0.5$ m. We conduct this experiment for several values of θ : 0.9, 0.99, 0.999, and 1.0. Note that $\theta = 1$ denotes the deterministic (disk) sensing model. In all cases (the figure is not shown), PCP ensured that the whole area is covered with the requested coverage threshold θ .

In addition, we check the connectivity of the nodes activated by PCP when the communication range varies from 15 to 40 m. The maximum separation s in this experiment is set to 30 m. We measure connectivity as the fraction of active nodes that are connected. We plot the results in Fig. 6. We show the minimum, average, and maximum values obtained from the 10 iterations. Confirming our analysis in Theorem 2, our protocol achieves full connectivity when $r_c \geq s$.

Savings and Flexibility Achieved by PCP. We conduct an experiment to assess the potential savings in number of active nodes because of using the exponential sensing model instead of the disk sensing model. Fig. 7 shows the results for different values of the coverage threshold θ , and for a range of values for the sensing decay factor α . The

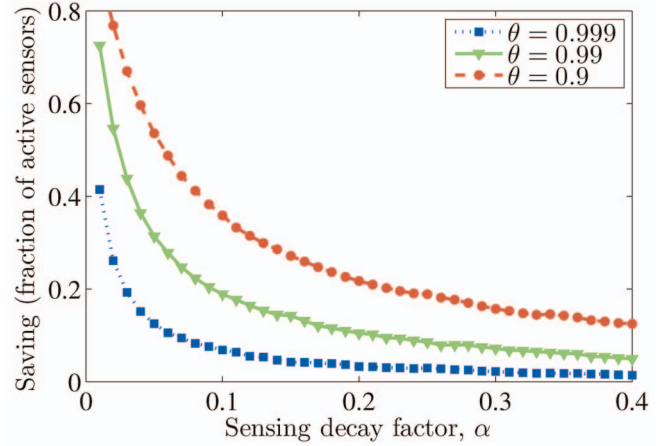


Fig. 7. Savings in number of active nodes because of using the exponential sensing model for different values of α and θ .

figure indicates that even for a conservative value of $\alpha = 0.05$ and for $\theta = 0.99$, a saving of up to 30 percent in number of active nodes can be achieved, which means less energy consumed and ultimately longer lifetimes 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, PCP provides a controllable knob: the coverage level θ , which can be employed to trade off reliability of sensing the area with the sensor network lifetime. For example, Fig. 7 shows that the number of activated sensors can be decreased if the coverage threshold θ is reduced. Reducing θ is feasible in applications that can tolerate a small probability of not detecting an event happening at a point, as it can be sensed at other points.

Theory versus Simulation. We compare the number of activated nodes and the convergence time resulted from simulation versus our theoretical analysis in Section 5. Some of the results are shown in Fig. 8, other results are similar. The results show that the upper bounds proved in Theorems 3 and 4 are only worst-case values, and our protocol performs better on the average case.

6.3 Robustness of PCP

We show that our protocol is robust against many practical aspects. 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 6.1 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 20 m, that is, a node could have as much as 20 m 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. As shown in Fig. 9a, PCP maintains the whole area covered even in the 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

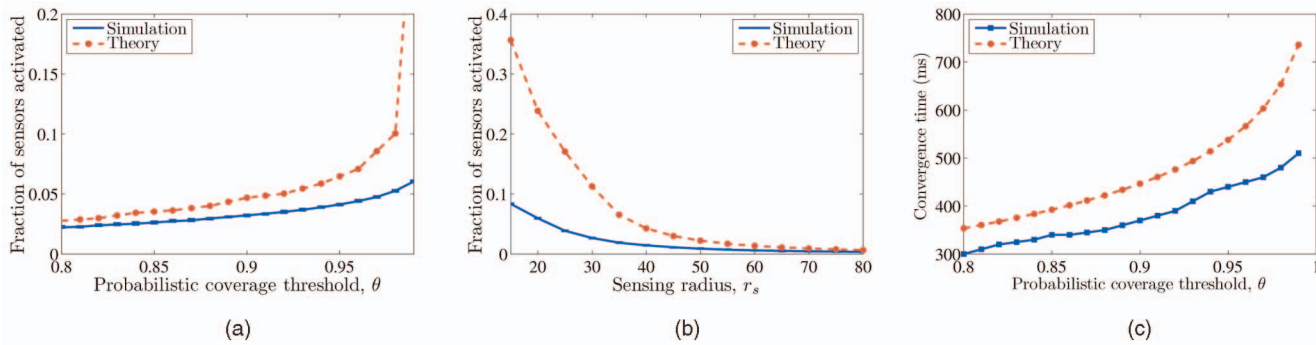


Fig. 8. Validation of the theoretical analysis: (a) and (b) fraction of sensors activated by PCP, and (c) convergence time of PCP.

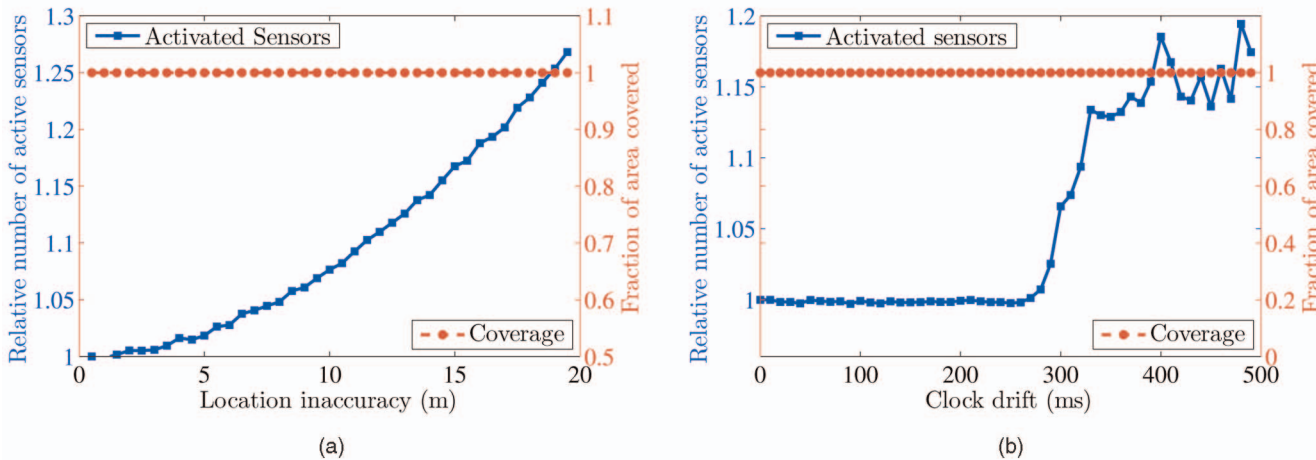


Fig. 9. Robustness of PCP against: (a) inaccurate node locations and (b) imperfect time synchronization. Note that there are two y-axes in each plot.

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. 9a (note 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 percent increase in the number of active sensors for location errors up to 10 m.

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 500 ms. For every value of d_{max} , we run our protocol till it converges, and compute the fraction of the area covered. As indicated in Fig. 9b, PCP is fairly robust against clock drifts because it can tolerate these drifts and maintain coverage. In addition, for practical clock drifts (up to 300 ms), there is virtually no increase in the number of activated sensors. For larger clock drifts, the cost is not significant as shown in Fig. 9b. Note that PCP converges in about 300 ms, on average. This explains why the number of active sensors starts to increase for clock drifts beyond 300 ms: 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.

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 percent. For each value of f , we run our protocol and periodically check the coverage of the whole area. The results, shown in Fig. 10,

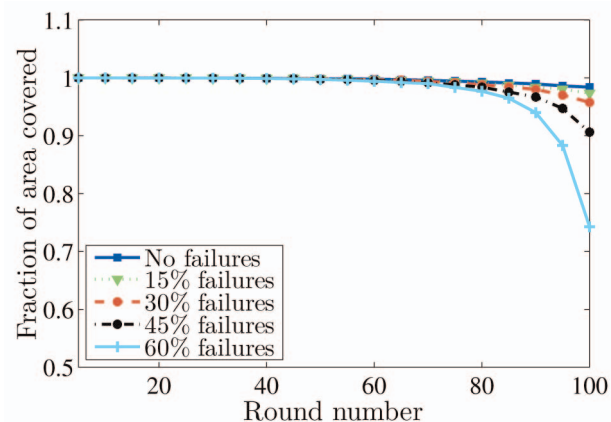


Fig. 10. Robustness of PCP against random node failures.

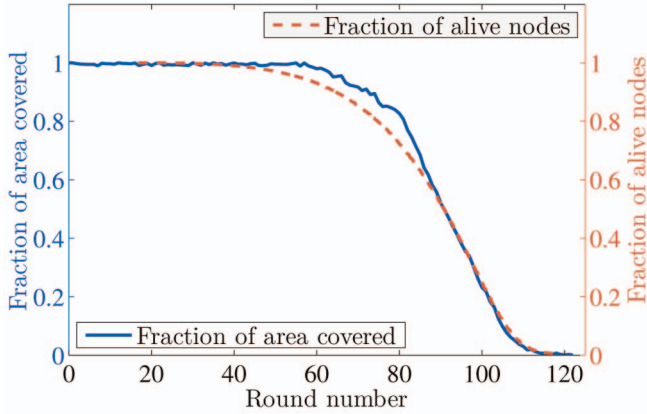


Fig. 11. Uniform energy consumption and network lifetime using PCP.

indicate that even with high failure rates, PCP maintains 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. We plot the average of the number of alive nodes versus round number in Fig. 11. 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 overutilize some nodes in early rounds, otherwise, they would have died earlier. Note that the energy of a node is enough for it to be active in about only five rounds. In addition, Fig. 11 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. In Fig. 12a, we change the number of starting

nodes k from 1 to 9 and plot the number of sensors activated by PCP to ensure coverage normalized by the number of active sensors when $k = 1$. In the same figure, we plot the normalized convergence time. As expected, increasing the number of starting points increases the number of active sensors but makes the protocol converges faster. In Fig. 12b, we repeat the same experiment but measure the normalized 80 percent lifetime, which is the time it takes for the coverage in the network to drop below 80 percent. The figure shows 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 converges, which consume more energy.

6.4 Comparing PCP versus Another Probabilistic Coverage Protocol (CCANS)

We compare our PCP protocol against the probabilistic coverage protocol (CCANS), proposed in [10], in terms of the number of activated sensors, network lifetime, and energy consumption. 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 validated our implementation of CCANS by obtaining the same results in [10]. To conduct fair comparisons, we make CCANS check only for coverage and not for connectivity.

In our comparison, we use the same exponential sensing model for CCANS and PCP with the same parameters. The parameters used for CCANS are taken from [10], and they 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 as explained in

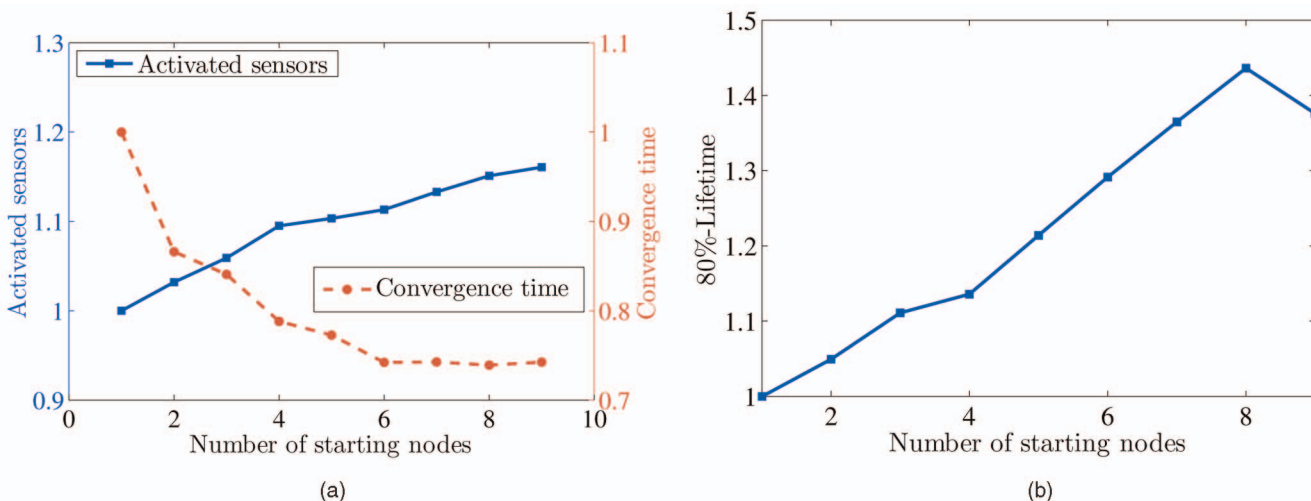


Fig. 12. Impact of multiple starting nodes on the performance of PCP: (a) convergence time and fraction of activate nodes and (b) network lifetime.

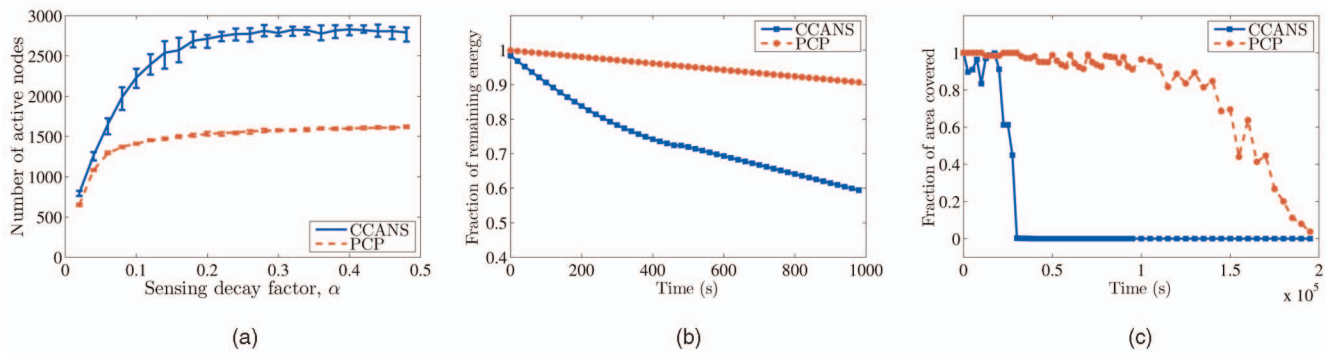


Fig. 13. Comparison between PCP and CCANS: (a) number of activated nodes, (b) total remaining energy, and (c) network lifetime.

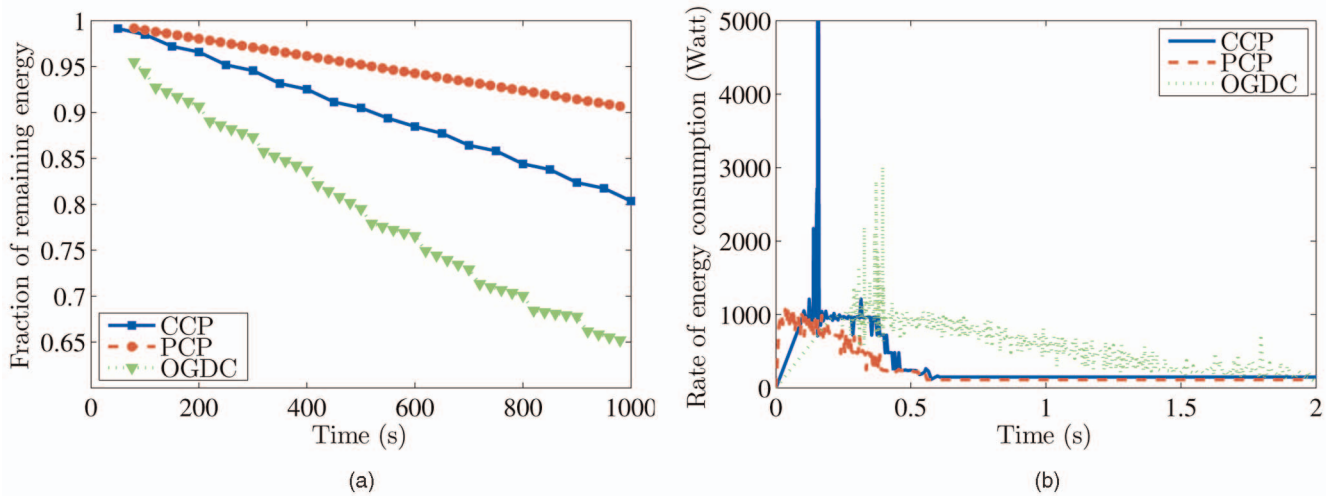


Fig. 14. Comparison among PCP, OGDC, and CCP: (a) total remaining energy in all nodes and (b) energy consumption per millisecond on a smaller timescale.

Section 3.2: For a uniform distribution of 20,000 nodes in a $1 \text{ km} \times 1 \text{ km}$ area, we have $\delta = 2 \times 1,000 \sqrt{2/20,000} = 20 \text{ m}$.

We plot in Fig. 13a the average number of nodes activated by PCP and CCANS for different values of the sensing decay factor α . 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 1,000 seconds in Fig. 13b. The figure shows that as CCANS activates more nodes and exchanges more messages than PCP, the node energy is depleted at a much faster rate. For example, after 1,000 seconds, the average energy of a node is 60 percent of its original energy if the sensor network uses CCANS to maintain coverage, while this average is 90 percent if our PCP protocol is used. Finally, the lifetime of the sensor network is shown in Fig. 13c, where we plot the fraction of the area covered with time. The lifetime of the network under our protocol is much larger than under CCANS.

6.5 Comparing PCP versus Other Deterministic Coverage Protocols (OGDC and CCP)

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 papers. 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. 14a, 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. To better understand the dynamics of energy consumption in the network, we plot in Fig. 14b the *rate* of energy consumption in terms of energy units per millisecond. The peak in the energy consumption for CCP represents the sending of HELLO messages. We notice that the main reason that makes OGDC consumes more energy than CCP and PCP is that it takes longer time to converge, which is shown by the high energy consumption over a longer period. To study this issue further, we analyze the dynamic change in node states with time. Recall that the energy model that we use in the comparison assigns

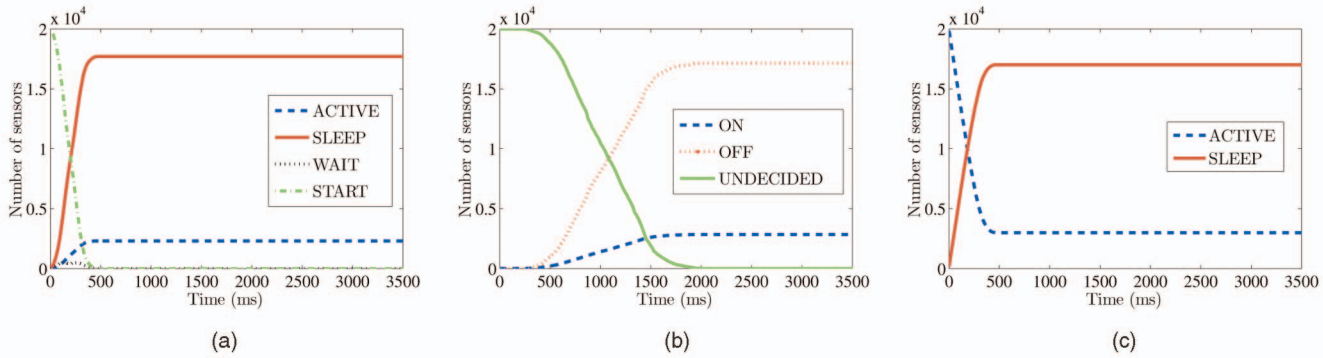


Fig. 15. State transitions with time for three deterministic coverage protocols: (a) PCP, (b) OGDC, and (c) CCP.

different energy consumption levels for transmitting, receiving, idling, or sleeping. For instance, the amount of energy consumed in receiving mode is 400 times more than the energy used in the sleep mode.

We plot in Fig. 15 the number of nodes in each state of the execution of these protocols versus the time. At the beginning of the execution of these protocols, a large amount of energy is consumed because all sensors in the field are active [4], [2]. Fig. 15 explains why PCP achieves the energy saving in Fig. 14a. For instance, comparing OGDC (Fig. 15b) versus PCP (Fig. 15a), we see that nodes decide to go to sleep much faster in PCP than in OGDC. Since the total number of deployed nodes is much larger than the activated subset of them, even a small difference in convergence time will make a significant difference in the energy consumption.

Finally, the convergence time of PCP, OGDC, and CCP can be inferred from Fig. 15, by finding the time at which the states of all nodes are decided. The figure shows that the convergence time of PCP and CCP is less than 500 ms, while it is more than 1,500 ms for OGDC.

7 CONCLUSION AND FUTURE WORK

In this paper, 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. In addition, we showed through simulation that a probabilistic coverage model may result in significant savings in the number of activated sensors, which reduces energy consumption and extends the network lifetime. It also provides a flexible way to control the number of activated sensors versus the level of coverage achieved by the protocol.

The work in this paper can be extended in several directions. For example, the analysis and design of our coverage protocol can be extended to the probabilistic k -coverage case. k -coverage is needed in several sensor network applications to enhance reliability and accuracy of the network. Using probabilistic sensing models in the k -

coverage case is expected to yield even higher savings in the number of activated sensors than in the 1-coverage case. Another extension is to consider probabilistic *communication* models, in addition to the probabilistic sensing models, in the design and operation of the protocol.

APPENDIX

Here, we show that under the exponential sensing model defined in (2), the least-covered point by three sensors located at vertices of an equilateral triangle is at the center of the triangle.

From geometric properties of triangles, the following relationship holds between any interior point c at distances x, y, z from the vertices of the triangle and the triangle side s (see Fig. 16):

$$s^2(x^2y^2 + s^2z^2) + s^2(s^2y^2 + x^2z^2) + s^2(s^2x^2 + y^2z^2) - s^2(x^4 + y^4 + z^4) - s^6 = 0. \quad (3)$$

Using Definition 2, the probability of sensing at point c is

$$P(c) = 1 - (1 - e^{-\alpha(x-r_s)})(1 - e^{-\alpha(y-r_s)})(1 - e^{-\alpha(z-r_s)}). \quad (4)$$

Note that using (3), we can replace z in (4) as a function of x and y , making $P(c)$ a function of only two variables. Now we show that $P(c)$ is minimum at the center of the triangle by showing that the partial derivatives of $P(c)$ with respect to x and y are zeros, and the second derivatives are positive, when $x = y = z$. Due to symmetry, we show that only for x :

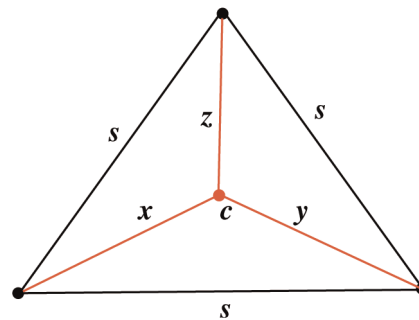


Fig. 16. Location of the least-covered point in an equilateral triangle formed by three sensors.

$$\begin{aligned} \frac{\partial P(c)}{\partial x} &= -\alpha e^{-\alpha(x-R)}(1 - e^{-\alpha(y-R)})(1 - e^{-\alpha(z-R)}) \\ &\quad - \frac{\partial z}{\partial x} \alpha e^{-\alpha(z-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(y-R)}). \end{aligned} \quad (5)$$

In order to find $\frac{\partial z}{\partial x}$, we differentiate (3) with respect to x :

$$\begin{aligned} 2xy^2 + 2\frac{\partial z}{\partial x}s^2z + 2xz^2 + 2\frac{\partial z}{\partial x}x^2z + 2s^2x + 2\frac{\partial z}{\partial x}y^2z \\ - 4x^3 - 4\frac{\partial z}{\partial x}z^3 = 0 \Rightarrow \frac{\partial z}{\partial x} = \frac{-x(y^2 + z^2 + s^2 - 2x^2)}{z(s^2 + x^2 + y^2 - 2z^2)}. \end{aligned} \quad (6)$$

At the center point, we have $x = y = z$, which yields

$$\frac{\partial z}{\partial x} = \frac{-x(x^2 + x^2 + s^2 - 2x^2)}{x(s^2 + x^2 + x^2 - 2x^2)} = -1.$$

Substituting $\frac{\partial z}{\partial x} = -1$ into (5) results in $\frac{\partial P(c)}{\partial x} = 0$. In a similar manner, it is easy to show that the second derivative of $P(c)$ with respect to x is positive:

$$\begin{aligned} \frac{\partial^2 P(c)}{\partial x^2} &= \alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(y-R)})(1 - e^{-\alpha(z-R)}) \\ &\quad - \frac{\partial z}{\partial x} \alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(y-R)})e^{-\alpha(z-R)} \\ &\quad - \frac{\partial^2 z}{\partial x^2} \alpha e^{-\alpha(z-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(y-R)}) \\ &\quad + \left(\frac{\partial z}{\partial x}\right)^2 \alpha^2 e^{-\alpha(z-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(y-R)}) \\ &\quad - \frac{\partial z}{\partial x} \alpha^2 e^{-\alpha(z-R)}e^{-\alpha(x-R)}(1 - e^{-\alpha(y-R)}). \end{aligned} \quad (7)$$

We differentiate (3) twice to find $\frac{\partial^2 z}{\partial x^2}$:

$$\begin{aligned} 2y^2 + 2\left(\frac{\partial z}{\partial x}\right)^2 s^2 + 2\frac{\partial^2 z}{\partial x^2} s^2 z + 2z^2 + 4\frac{\partial z}{\partial x} xz + 2\frac{\partial^2 z}{\partial x^2} x^2 z \\ + 4\frac{\partial z}{\partial x} xz + 2\left(\frac{\partial z}{\partial x}\right)^2 x^2 + 2s^2 + 2\frac{\partial^2 z}{\partial x^2} y^2 z + 2\left(\frac{\partial z}{\partial x}\right)^2 y^2 \\ - 12x^2 - 4\frac{\partial^2 z}{\partial x^2} z^3 - 12\left(\frac{\partial z}{\partial x}\right)^2 z^2 = 0. \end{aligned}$$

Again we have $x = y = z$, and $\frac{\partial z}{\partial x} = -1$ at the center point:

$$\begin{aligned} 2x^2 + 2(-1)^2 s^2 + 2\frac{\partial^2 z}{\partial x^2} s^2 x + 2x^2 - 4x^2 + \frac{2\partial^2 z}{\partial x^2} x^3 \\ - 4x^2 + 2x^2 + 2s^2 + 2\frac{\partial^2 z}{\partial x^2} x^3 + 2x^2 - 12x^2 \\ - 4\frac{\partial^2 z}{\partial x^2} x^3 - 12x^2 = 0 \Rightarrow \frac{\partial^2 z}{\partial x^2} = \frac{24x^2 - 4s^2}{2s^2 x}. \end{aligned}$$

Since the triangle is equilateral, $s = \sqrt{3}x$. Thus,

$$\frac{\partial^2 z}{\partial x^2} = \frac{24x^2 - 12x^2}{6x^3} = \frac{2}{x}. \quad (8)$$

Finally, we substitute (8) into (7) and use $x = y = z$:

$$\frac{\partial^2 P(c)}{\partial x^2} = \alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)}) \left[2 - \frac{2}{\alpha x} (1 - e^{-\alpha(x-R)}) \right]. \quad (9)$$

We know that for any $\alpha, R > 0$ and $x \geq R$:

$$\begin{aligned} \alpha^2 &> 0, \\ e^{-\alpha(x-R)} &> 0, \\ (1 - e^{-\alpha(x-R)}) &> 0. \end{aligned}$$

Moreover, it is known that for any $w \in \mathcal{R}$, $e^{-w} \geq 1 - w$. Therefore,

$$\begin{aligned} 1 - e^{-\alpha(x-R)} &\leq \alpha(x-R) < \alpha x \\ \Rightarrow 2 - \frac{2}{\alpha x} (1 - e^{-\alpha(x-R)}) &> 0. \end{aligned}$$

This means that all four terms in (9) are positive. Therefore, the second derivative of coverage probability is positive.

ACKNOWLEDGMENTS

This work is partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

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, Mar. 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 Trans. Sensor Networks*, vol. 1, no. 1, pp. 36-72, Aug. 2005.
- [3] Z. Zhou, S. Das, and H. Gupta, "Connected K-Coverage Problem in Sensor Networks," *Proc. Int'l Conf. Computer Comm. and Networks (ICCCN '04)*, pp. 373-378, Oct. 2004.
- [4] H. Zhang and J. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc and Sensor Wireless Networks: An Int'l J.*, vol. 1, nos. 1/2, pp. 89-123, Jan. 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, Nov. 2005.
- [6] S. Kumar, T.H. Lai, and J. Balogh, "On K-Coverage in a Mostly Sleeping Sensor Network," *Proc. ACM MobiCom*, pp. 144-158, Sept. 2004.
- [7] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS '03)*, pp. 28-37, May 2003.
- [8] D. Tian and N. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications*, pp. 32-41, Sept. 2002.
- [9] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Trans. Embedded Computing Systems*, vol. 3, no. 1, pp. 61-91, Feb. 2004.
- [10] Y. Zou and K. Chakrabarty, "A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks," *IEEE Trans. Computers*, vol. 54, no. 8, pp. 978-991, Aug. 2005.
- [11] N. Ahmed, S. Kanhere, and S. Jha, "Probabilistic Coverage in Wireless Sensor Networks," *Proc. IEEE Conf. Local Computer Networks (LCN '05)*, pp. 672-681, Nov. 2005.
- [12] B. Liu and D. Towsley, "A Study on the Coverage of Large-Scale Sensor Networks," *Proc. IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '04)*, pp. 475-483, Oct. 2004.
- [13] Q. Cao, T. Yan, T. Abdelzaher, and J. Stankovic, "Analysis of Target Detection Performance for Wireless Sensor Networks," *Proc. Int'l Conf. Distributed Computing in Sensor Networks*, pp. 276-292, June 2005.
- [14] M. Cardei and J. Wu, "Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," *Elsevier Computer Comm.*, vol. 29, no. 4, pp. 413-420, 2006.

- [15] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity," *Proc. ACM MobiHoc*, pp. 131-142, May 2006.
- [16] Z. Abrams, A. Goel, and S. Plotkin, "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '04)*, pp. 424-432, Apr. 2004.
- [17] D. Tian and N. Georganas, "Location and Calculation-Free Node Scheduling Schemes in Large Wireless Sensor Networks," *Elsevier Ad Hoc Networks*, vol. 2, pp. 65-85, 2004.
- [18] T. Wu and K. Ssu, "Determining Active Sensor Nodes for Complete Coverage without Location Information," *Int'l J. Ad Hoc and Ubiquitous Computing*, vol. 1, nos. 1/2, pp. 38-46, 2005.
- [19] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Networks," *Proc. IEEE INFOCOM*, pp. 1380-1387, Apr. 2001.
- [20] D. Mehta, M. Lopez, and L. Lin, "Optimal Coverage Paths in Ad-Hoc Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '03)*, May 2003.
- [21] X. Li, P. Wan, and O. Frieder, "Coverage in Wireless Ad Hoc Sensor Networks," *IEEE Trans. Computers*, vol. 52, no. 6, pp. 753-763, June 2003.
- [22] L. Lazos and R. Poovendran, "Stochastic Coverage in Heterogeneous Sensor Networks," *ACM Trans. Sensor Networks*, vol. 2, no. 3, pp. 325-358, Aug. 2006.
- [23] C. Huang and Y. Tseng, "The Coverage Problem in a Wireless Sensor Network," *ACM Mobile Networks and Applications (MONET)*, special issue on wireless sensor networks, vol. 10, no. 4, pp. 519-528, Aug. 2005.
- [24] A. So and Y. Ye, "On Solving Coverage Problems in a Wireless Sensor Network Using Voronoi Diagrams," *Proc. Workshop Internet and Network Economics (WINE '05)*, pp. 584-593, Dec. 2005.
- [25] T. Okabe, B. Boots, K. Sugihara, and S.N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, second ed. John Wiley, 2000.
- [26] C. Huang, Y. Tseng, and H. Wu, "Distributed Protocols for Ensuring Both Coverage and Connectivity of a Wireless Sensor Network," *ACM Trans. Sensor Networks*, vol. 3, no. 1, Mar. 2007.
- [27] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Trans. Computers*, vol. 51, no. 12, pp. 1448-1453, Dec. 2002.
- [28] S. Yang, F. Dai, M. Cardei, J. Wu, and F. Patterson, "On Connected Multiple Point Coverage in Wireless Sensor Networks," *Int'l J. Wireless Information Networks*, vol. 13, no. 4, pp. 289-301, May 2006.
- [29] F. Dai and J. Wu, "An Extended Localized Algorithm for Connected Dominating Sets Formation in Ad Hoc Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 10, pp. 908-920, Oct. 2004.
- [30] H. Gupta, Z. Zhou, S. Das, and Q. Gu, "Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution," *IEEE/ACM Trans. Networking*, vol. 14, no. 1, pp. 55-67, Feb. 2006.
- [31] M. Hefeeda and M. Bagheri, "Randomized K-Coverage Algorithms for Dense Sensor Networks," *Proc. IEEE INFOCOM*, pp. 2376-2380, May 2007.
- [32] A. Savvides, C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiCom*, pp. 166-179, July 2001.
- [33] L. Doherty, L.E. Ghaoui, and K. Pister, "Convex Position Estimation in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, pp. 1655-1663, Apr. 2001.
- [34] NS-2, <http://nslam.isi.edu/nslam/>, 2009.
- [35] Network Systems Lab, <http://nsl.cs.sfu.ca/wiki>, 2009.



Mohamed Hefeeda (S'01, M'04, SM'09) received the BSc and MSc degrees from Mansoura University, Egypt, in 1994 and 1997, respectively, and the PhD degree from Purdue University, West Lafayette, Indiana, in 2004. He is an assistant professor in the School of Computing Science, Simon Fraser University, Surrey, British Columbia, Canada, where he leads the Network Systems Lab. His research interests include multimedia networking over wired and wireless networks, peer-to-peer systems, network security, and wireless sensor networks. He has served on several technical program committees of major conferences in his research areas, including ACM Multimedia, ACM Multimedia Systems, ACM/SPIE Multimedia Computing and Networking (MMCN), the IEEE Conference on Network Protocols (ICNP), and the IEEE Conference on Communications (ICC). He is an associate editor of the *International Journal of Advanced Media and Communication* and the guest editor of that journal's special issue on High-Quality Multimedia Streaming in P2P Environments. His paper on the hardness of optimally broadcasting multiple video streams with different bit rates won the Best Paper Award in the IEEE Innovations 2008 Conference. In addition to publications, he and his students develop actual systems, such as PROMISE, pCache, svcAuth, pCDN, and mobile TV testbed, and contribute the source code to the research community. The mobile TV testbed software developed by his group won the Best Technical Demo Award in the ACM Multimedia 2008 Conference. He is a senior member of the IEEE and a member of the ACM SIGCOMM and SIGMM.



Hossein Ahmadi received the BSc degree from Sharif University, Iran, in 2005, and the MSc degree from Simon Fraser University, Canada, in 2007. He is currently working toward the PhD degree at the University of Illinois, Urbana-Champaign. His research interests are in the area of wireless sensor networks.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.