PROBABILISTIC COVERAGE AND CONNECTIVITY IN
WIRELESS SENSOR NETWORKS

by

Hossein Ahmadi
B.Sc., Sharif University of Technology, Tehran, Iran, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

© Hossein Ahmadi 2007
SIMON FRASER UNIVERSITY
Summer 2007

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
APPROVAL

Name: Hossein Ahmadi
Degree: Master of Science
Title of thesis: Probabilistic Coverage and Connectivity in Wireless Sensor Networks

Examinig Committee: Dr. Richard Vaughan
Chair

Dr. Mohamed Hefeeda, Senior Supervisor

Dr. Qianping Gu, Supervisor

Dr. Jiangchuan Liu, Examiner

Date Approved: ________________________________
Abstract

Several empirical studies have shown that sensing and communication ranges of sensors are not regular disks. Rather, they follow probabilistic models. Yet, many current coverage and connectivity protocols continue to assume the disk model for ease of analysis, which may lead to incorrect operation of these protocols in real environments. We propose a distributed coverage and connectivity maintenance protocol that explicitly accounts for the probabilistic nature of communication and sensing ranges. Through analytical analysis, we show that our protocol guarantees a target packet delivery rate in the network, while ensuring the monitored area is covered with a probability exceeding a given threshold. Using large-scale simulations, we compare our protocol against others in the literature and show that it activates fewer nodes, consumes much less energy, and significantly prolongs the network lifetime. We also demonstrate the robustness of our protocol against random node failures, node location inaccuracy, and imperfect time synchronization.
To my wife Maryam, with love.
“Not everything that counts can be counted, and not everything that can be counted counts.”

— Albert Einstein
I am deeply indebted to my senior supervisor, Dr. Mohamed Hefeeda, for his continuous support, encouragement and guidance through my research. Mohamed provided valuable insights, and a lot of help during my graduate career. This thesis would not have been possible without him.

I would like to thank my supervisor Dr. Qianping Gu and my thesis examiner Dr. Jiangchuan Liu for being on my committee and reviewing this thesis. I would like to thank Dr. Richard Vaughan for taking the time to chair my thesis defense. I would also like to extend my gratitude to the faculty and staff in the school of computing science at SFU.

I would like to thank my family who inspired me to strive beyond the ordinary, encouraged me, and gave me the confidence I needed to make it through life. I also would like to thank my friends who helped me through these years of study and made it fun.

Last but certainly not least, I would like to thank my wife Maryam, for her love, care and support. This thesis is dedicated to her.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval</td>
<td>ii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Quotation</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vi</td>
</tr>
<tr>
<td>Contents</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
</tbody>
</table>

## 1 Introduction and Background

1.1 Introduction ........................................ 1
1.2 Wireless Sensor Networks ......................... 5
1.3 Thesis Contributions ............................... 6
1.4 Thesis Organization ............................... 7

## 2 Probabilistic Coverage

2.1 Related Work ....................................... 8
2.2 Probabilistic Coverage ........................... 10
   2.2.1 Coverage using Disk Sensing Model ........... 10
   2.2.2 Coverage using Probabilistic Sensing Models 11
5.5 Comparing PCMP against another Integrated Coverage and Connectivity Protocol ......................................................... 59

6 Conclusions and Future Work .................................................. 61
6.1 Conclusions ................................................................. 61
6.2 Future Work .............................................................. 62

A Least-covered Point Under the Exponential Model .............. 64

Bibliography ........................................................................ 67
List of Tables

5.1 Parameters used for the log-normal shadowing communication model. . . . . 49
List of Figures

1.1 Some of the sensing models used in the literature. ........................................ 3

2.1 The structure of the optimal coverage using the disk sensing model. This structure minimizes the overlap between the sensing ranges of nodes. ................. 9

2.2 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................................... 11

2.3 The node activation process in PCP. Activated nodes try to form a triangular lattice over the area in a way that ensures that the least-covered point in each triangle has a probability of being sensed more than or equal to $\theta$. ............ 12

2.4 Location of the least-covered point in an equilateral triangle formed by three sensors. ............................................................... 13

2.5 The state diagram of the PCP protocol. In each state, we mark which of the sensing, sending, and receiving modules is on. ................................. 17

2.6 Choosing the closest node to a triangle vertex. ........................................... 18

3.1 Validation of the PCP protocol: (a) Achieved coverage and (b) Connectivity of the activated nodes............................................................... 27

3.2 Savings in number of active nodes because of using the exponential sensing model for different values of $\alpha$ and $\theta$. .................................................... 27

3.3 Fraction of sensors activated by PCP: Theory versus simulation. (a) Different values for the coverage threshold parameter $\theta$ and (b) Different values for the exponential decay factor $\alpha$. For simulations, we report the minimum, average, and the maximum values. .................................................... 28
3.4 Convergence time of PCP: Theory versus simulation. (a) Different values for the coverage threshold parameter $\theta$ and (b) Different values for the exponential decay factor $\alpha$. .................................................. 29
3.5 Robustness of the PCP protocol against: (a) inaccurate node locations, and (b) imperfect time synchronization. Notice that there are two y-axes in each plot. ................................................................. 30
3.6 Robustness of the PCP protocol against random node failures. .............................................. 32
3.7 Uniform energy consumption and network lifetime using PCP protocol. .................. 33
3.8 Impact of multiple starting nodes on the performance of the PCP protocol: (a) convergence time and fraction of activate nodes, and (b) network lifetime. ........................................... 33
3.9 Comparison between number of activated nodes by PCP and CCANS for different values of: (a) $\alpha$ and (b) $\theta$. .................................................. 34
3.10 Comparison between PCP and CCANS: (a) Total remaining energy in all nodes, and (b) Network life time. ................................................................. 35
3.11 Comparison between PCP, OGDC, and CCP: (a) Total remaining energy in all nodes, and (b) Energy consumption per millisecond on a smaller time scale. 36
3.12 State transitions with time for three deterministic coverage protocols: (a) PCP, (b) OGDC, and (c) CCP. .................................................. 37

4.1 Probabilistic connectivity in triangular mesh. ................................................................. 41
4.2 The triangular mesh construction used in proving Theorem 5. .............................. 42
4.3 The square mesh construction process used in proving Theorem 6. ..................... 44

5.1 Network delivery rate in triangular mesh: Theorem 5 vs. Simulation. For the simulation data, we show the minimum, average, and maximum values. 50
5.2 Network delivery rate in square mesh: Theorem 6 vs. Simulation. For the simulation data, we show the minimum, average, and maximum values. 51
5.3 Network delivery rate in uniform distribution: Theorem 7 vs. Simulation. For the simulation data, we show the minimum, average, and maximum values. 51
5.4 Potential benefits of using probabilistic communication models over the deterministic disk model: (a) Achieved network delivery rate, (a) Saving in number of activated nodes. .................................................. 53
5.5 PCMP achieves the requested network delivery rate in all cases. The minimum, average, and maximum values are shown. ......................... 54
5.6 Probabilistic coverage and connectivity achieved by PCMP. 54
5.7 Robustness of PCMP against: (a) Location inaccuracy, and (b) Clock drifts. 55
5.8 Robustness of PCMP against random node failures. 56
5.9 The deterministic connectivity of SPAN, PCMP and GAF. 57
5.10 Comparing PCMP against SPAN and GAF: (a) Energy consumption, and
    (b) Network lifetime. 57
5.11 Comparing PCMP against CCP-SPAN: (a) Number of activated nodes, and
    (b) Energy consumption. 59
Chapter 1

Introduction and Background

In this chapter, we introduce the coverage and connectivity problems in wireless sensor networks and give a brief background. Then, we summarize the contributions of this thesis. We also describe the organization of the thesis.

1.1 Introduction

Sensor networks have been proposed for many applications such as forest fire detection, area surveillance, and natural habitat monitoring [3]. A common ground for all such applications is that every sensor can detect an event occurring within its sensing range, and communicate with a sensor inside the communication range 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 sensing 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 $\epsilon$ values (see Fig. 1.1(a)). In this case, the area is covered if any arbitrary point in the area has a sensor within the range of $r_s$. The disk sensing model is appealing, because it makes coverage maintenance protocols, e.g., [30, 35, 36], less complicated to design and analyze. It also makes analytical and asymptotic analysis, e.g., [18, 26], 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 [2, 7, 19, 37, 38] 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 [7] 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 [37, 38] use an exponential sensing model, where the sensing capacity degrades according to an exponential distribution after a certain threshold, as shown in Fig. 1.1(b). Whereas the authors of [19] propose a polynomial function to model the probabilistic nature of the sensing range, as shown in Fig. 1.1(d). Furthermore, the authors of [2] 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. 1.1(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 [35], 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 [30], PEAS [33], Ottawa [29], and OGDC [35], 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
In this thesis, 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 or communication 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 and communication models. Even for the same sensor type, these models may need to be changed in different environments or when the technology changes. Designing, implementing, and testing a different coverage protocol for each 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 and communication models. In particular, our protocol requires the computation of a single parameter from the adopted models, while everything else remains the same.

We show in this thesis how this parameter can be derived in general, and we actually do the calculations for two sensing models: (i) the probabilistic exponential sensing model [37, 38], 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 [38]). 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 the robustness of our protocol against random node failures, node location inaccuracy, and imperfect time synchronization.

Network connectivity is another fundamental problem closely related to coverage in wireless sensor networks. A network is connected if every pair of nodes can communicate with each other. To study network connectivity, many previous works represent the network with an undirected unweighted graph, where network connectivity is equivalent to graph connectivity. In the graph representation, there is an edge between two nodes if they are within the communication range of each other. Furthermore, the communication range of a node is typically assumed to be a disk with radius $r_c$, where $r_c$ is referred to as the communication range of a node. We refer to this kind of connectivity as the deterministic connectivity model. The deterministic connectivity model started in wired networks, and then used widely in wireless ad hoc and sensor networks. While it is fairly accurate in wired networks, several papers, e.g. [1,17], argue that the deterministic connectivity model is not appropriate for wireless networks. This is because it has been experimentally shown that communication ranges of nodes are not nice regular disks. Rather, they follow probabilistic models. Therefore, two wireless nodes can not said to be ‘connected’ or ‘disconnected’ in the perfect sense. Instead, a link between a pair of wireless nodes should have a probability of data delivery between these two nodes. In addition, it is neither sufficient nor precise to state that the network is simply connected. Rather, a quantitative measure of the quality of communications between arbitrary nodes in the network is needed.

Despite the experimental evidence of the inaccuracy of the deterministic connectivity model, many current connectivity maintenance protocols in the literature, e.g., [31, 32], continue to use it. The deterministic connectivity model is used because it facilitates the design and performance analysis of the protocols. By relying on the deterministic connectivity model, current protocols may not function properly in real environments. In addition, these protocols fail to provide any assessment of the quality of communication between nodes in a wireless sensor network.

In this thesis, we take a first step in designing connectivity maintenance protocols for more realistic communication models. We propose a connectivity maintenance protocol (called PCMP) that explicitly accounts for the probabilistic nature of communication links and achieves a given target communication quality between nodes. Then, we integrate this connectivity maintenance protocol with our coverage protocol to form an integrated coverage and connectivity maintenance protocol. Our integrated protocol is simple to implement, and
we demonstrate its robustness against random node failures, inaccuracy of node locations, and imperfect time synchronization of node clocks using extensive simulations. We show that our integrated protocol outperforms the only other integrated coverage and connectivity protocol proposed in literature that we are aware of.

1.2 Wireless Sensor Networks

Wireless sensor networks are deployed to be used in monitoring applications such as forest fire detection, habitat monitoring, and battlefield monitoring. The nodes are small devices with a low power processor, small amount of memory, a wireless interface, and a sensing interface. The power of sensors is provided by batteries. Several sensor motes have been designed and produced such as Mica2, MicaZ, and TelosB [11]. The sensing module can be used to detect changes in temperature, light, humidity, or to detect objects using sonar or radio waves.

Sensors are expected to be available at low cost due to mass production. They can be deployed manually for small areas but for large areas, other methods like dropping nodes from a plane is used. Various node distributions such as square mesh, triangular mesh, and uniform random deployment are used for different applications. We refer to this distribution by Node Deployment Distribution. In monitoring applications, we define coverage to measure the quality of monitoring provided by deployed sensors. We say that the sensor network covers the area if all points in the area are covered by at least one sensor. This definition is extended to the case of having multiple sensors covering any arbitrary point. The area is $k$-covered by the deployed sensors if any arbitrary point is being detected by at least $k$ sensors.

The main task of a sensor network is to collect data from the surveillance area and report it to a base station. To achieve this, sensors can form a network with various architectures depending on the application, sensor types, and power constraints. Sensors can send the data directly to the base station or use a multi-hop path to deliver the data. Several routing mechanisms have been proposed to address energy constraints as well as latency in delivering data in wireless sensor networks. Sensors can also do aggregation when forwarding data to other nodes. This reduces the communication overhead and saves the energy. Regardless of the mechanism used for the delivery of data, it is essential to have all nodes in the network connected to each other to be able to deliver the data. The network is connected if every
pair of sensors can communicate with each other.

Power consumption is one of the fundamental concerns in wireless sensor networks. Sensors can last for a few weeks using their batteries. However, we are interested in extending their lifetimes into months. The solution is to deploy some extra sensors and distribute the workload between nodes to increase the lifetime. In this case, some protocols are needed to schedule activation and deactivation of nodes while keeping the coverage and connectivity quality. The protocols maintaining the area covered are often referred to as Coverage Protocols while Connectivity Protocols guarantee the communication quality between nodes. Many previous works in the literature use unrealistic assumptions on sensing capacities and wireless communications in designing coverage and connectivity protocols. In this thesis, we use probabilistic models to develop coverage and connectivity protocols and we propose a more realistic measure of connectivity.

1.3 Thesis Contributions

In this thesis, we propose a solution to coverage and connectivity problems in wireless sensor network with realistic assumptions about sensing and communication ranges. Particularly, our contributions [13,14] can be summarized as follows:

- We propose a distributed coverage protocol to guarantee coverage in the surveillance area [14]. Our protocol is simple to implement, and we demonstrate its robustness against random node failures, inaccuracy of node locations, and imperfect time synchronization of node clocks using extensive simulations. We show that our protocol minimizes the number of activated nodes and consumes much less energy than other protocols in the literature.

- We provide a quantitative measure of the quality of communication between nodes in sensor networks by defining the probability of packet delivery between arbitrary nodes in the network [13]. We analytically derive this probability for common node deployment schemes such as grid, triangular lattice, and uniform deployments.

- We propose a probabilistic connectivity maintenance protocol that explicitly accounts for the probabilistic nature of communication links and achieves a given target communication quality between nodes [13].
• We design an integrated coverage and connectivity maintenance protocol to achieve a given target communication quality between nodes while maintaining the area covered. The operation of our protocol does not depend on the specifics of the adopted sensing and communication models, which enables our protocol to be used with different models and in various environments. To the best of our knowledge, our proposed protocol is the first protocol to employ both probabilistic communication and probabilistic sensing models at the same time. Therefore, our protocol is more suitable for real sensor network environments than most others in the literature.

1.4 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 describes the probabilistic coverage problem and presents our new probabilistic coverage protocol. We analyze the correctness and performance of our proposed protocol and rigorously evaluate it through simulations in Chapter 3. Chapter 4 presents the connectivity problem and our proposed probabilistic connectivity model and protocol. In Chapter 5, we evaluate our proposed connectivity protocol against two other widely cited protocols in the literature. We also compare our integrated coverage and connectivity protocol against another integrated protocol. Chapter 6 concludes the thesis and outlines the future works. Appendix A gives the proof of one of the theorems mentioned in Chapter 2.
Chapter 2

Probabilistic Coverage

In this chapter we discuss the coverage problem in sensor networks and propose a new energy efficient protocol to solve the problem. We first summarize the related work in Section 2.1. Next, we formally define the probabilistic coverage problem, and present the key ideas behind our new probabilistic coverage protocol in Section 2.2. In Section 2.3, we present the details of our new protocol. In Section 2.4, 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 deterministic connectivity in addition to coverage. In the next chapter, we evaluate our protocol and compare it against other deterministic and probabilistic coverage protocols in the literature.

2.1 Related Work

Coverage in sensor networks has received significant research attention. The studies in [18, 26] conduct asymptotic and analytical analysis to provide necessary and sufficient conditions for coverage in various environments. In [4], 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 [35] tries to minimize the overlap between the sensing circles of activated sensors, while CCP [30] deactivates redundant sensors by checking that all intersection
points of sensing circles are covered. Other earlier protocols include PEAS [33] and Ottawa [29]. We compare our protocol against the more recent OGDC and CCP protocols, because, according to the performance evaluations in [30, 35], they outperform the earlier ones.

Probabilistic coverage with various sensing models has also been studied in [2, 19, 38]. The work in [19] analytically studies the implications of adopting probabilistic and disk sensing models on coverage, but no specific coverage protocol is presented. In [2], 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 [38], 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.

Coverage with various degrees ($k$-coverage), where every point is sensed by at least $k$ sensors, has also been studied, e.g., in [15, 30, 36]. Because of the hardness of the problem, most of these works assume disk model.
CHAPTER 2. PROBABILISTIC COVERAGE

2.2 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.

2.2.1 Coverage using Disk Sensing Model

In the disk sensing model, all events within the sensing range \( r_s \), are deterministically detected by the sensor. On the other hand, events happening farther can not be detected at all. The coverage under disk sensing model is often referred to as Deterministic Coverage and defined as following:

**Definition 1 (Deterministic Coverage)** An area \( A \) is deterministically covered by \( n \) sensors if \( \exists i, (1 \leq i \leq n) \), such that \( d_i(x) < r_s \) for every point \( x \) in \( A \), where \( d_i(x) \) is the distance between sensor \( i \) and point \( x \).

Based on the above definition each point in the area is covered by at least one sensor. However, some applications require multiple sensors monitoring every point at the same time for several reasons. For example it can be used to increase the accuracy of readings or to make the network more robust against the node failures. The \( k \)-coverage is defined as follows:

**Definition 2 (k-Coverage)** An area \( A \) is \( k \)-covered by \( n \) sensors if for every point \( x \) in \( A \), there are at least \( k \) sensors with distance of at most \( r_s \) from \( x \).

The disk sensing model simplifies the coverage problem. In fact, optimal solutions for it can be obtained efficiently. As mentioned in [4], 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 \( \sqrt{3}r_s \). This is shown in Fig. 2.1. 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.

2.2.2 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 3 (Probabilistic Coverage) An area $A$ is probabilistically covered by $n$ sensors with threshold parameter $\theta$ ($0 < \theta \leq 1$) 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 $\theta$ 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 4 (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. 2.2 demonstrates the concept of the least-covered point by the showing the sensing capacity of three nodes deployed on an equi-lateral triangle that use the exponential sensing model.

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 \( \theta \). 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 \( \theta \). 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
Figure 2.4: Location of the least-covered point in an equilateral triangle formed by three sensors.

θ. Knowing this maximum length, the coverage protocol functions in the same manner as described in Section 2.2.1: It tries to activate nodes at vertices of the lattice triangles. This activation process is described in Fig. 2.3. 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 thesis. 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 [37, 38], 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 2.3 and 2.4, 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.
2.2.3 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:

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

(2.1)

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 \( \alpha \) is a factor that describes how fast the sensing capacity decays with distance. We call \( \alpha \) the sensing capacity decay factor. The exponential model is shown Fig. 1.1(b). We consider this sensing model for two reasons. First, it has been adopted before in [37,38], which allows us to conduct a fair comparison between our protocol and the protocol in [38]. 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 [2, 7, 19]. 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 (2.1), 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 \( \theta \) is \( \sqrt{3}(r_s - \ln(1 - \frac{\sqrt{3}}{2\alpha \theta})) \).

**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 the Appendix that this location is at the center of the triangle, which is at a distance of \( s/\sqrt{3} \) from each vertex, as depicted
in Fig. 2.4. The probability of sensing at the least-covered point is then \(1 - (1 - e^{-\alpha(\sqrt{3} - r_s)})^3\) which should be greater than or equal to \(\theta\). Manipulating this inequality, we get the maximum separation \(s = \sqrt{3}(r_s - \ln(1 - \frac{\sqrt{3}}{\sqrt{1-\theta}}))\).

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 [4,35].

2.3 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.

2.3.1 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 \(\theta\)) 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 $\theta$, 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 2.3.3, 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 2.3.4, 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 [12, 25], any of them can be used with our protocol. The protocols that we compare ours against [30, 35, 38] 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. 2.6):

\[ T_a = \tau_a (d_v^2 + d_a^2 \gamma^2), \]  

(2.2)

where \(d_v\), and \(d_a\) are the Euclidean distances between the node and the vertex, and the node and the activator, respectively; \(\gamma\) is the angle between the line connecting the node with the activator and the line connecting the vertex with the activator; and \(\tau_a\) 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.

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

**Definition 5 (\(\delta\)-circle)** The smallest circle drawn anywhere in the monitored area such

\(^1\)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 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 \(\gamma\) is between 0 and \(2\pi\) while the scale of \(d_v\) can change in different applications. Therefore, \(\gamma\) 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 \(\gamma\) inside a \(\delta\)-circle is proportional to \(\gamma^2\). Thus, the activation timer is formed by summation of \(d_v^2\) and scaled angle \((d_a\gamma)^2\).
that there is at least one node inside it is called the $\delta$-circle, where $\delta$ is the diameter of the circle.

The diameter $\delta$ is computed from the deployment distribution of nodes. In Section 2.3.2, we show how $\delta$ 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 state diagram in Fig. 2.5 shows the status of the sensing, sending, and receiving modules in each state of the node.) The savings in energy are significant as shown in the evaluation section. PCP achieves this optimization by making only nodes inside $\delta$-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 $\delta$-circles. Nodes inside $\delta$-circles compute activation timers, as described above, to choose the closest node the vertex to be active. As shown in Fig. 2.6, centers of $\delta$-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.

2.3.2 Computing $\delta$-circles for Different Deployment Distributions

As mentioned in the previous section, node deployment distribution determines the value of $\delta$, which is the diameter of the smallest circle with at least one node inside it. In this section, we compute $\delta$ for two common deployment schemes: grid and uniform distribution. $\delta$ 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 $\delta$, 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 $\delta$ larger than the diagonal of this small square ensures that a $\delta$-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 $\lambda$, whereas the maximum distance does not exceed $2\lambda$. Using a similar argument as in the grid distribution, $\delta$ should be $2\sqrt{2}\lambda$. To uniformly distribute $n$ nodes over an $l \times l$ square, $\lambda$ should be $l/\sqrt{n}$, which results in $\delta = 2l\sqrt{2/n}$. Note that randomness in the deployment distribution results in larger $\delta$ values.

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

### 2.3.3 Multiple Starting Nodes

In Section 2.3.1, 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 startup timer $T_s$. $T_s$ is chosen randomly between 0 and a constant $\tau_s$. Suppose that we want to compute the value of $\tau_s$ 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, $\tau_s$ 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, $\tau_s$ 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.

### 2.3.4 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.

### 2.4 Analysis 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.

#### 2.4.1 Correctness and Complexity Analysis

We carry out our analysis in terms of the input parameters $\delta, \theta, s,$ and $l$, and the protocol parameter $\tau_a$, which is the maximum value of the activation timer. $\delta$ is determined from the deployment distribution of sensors as explained in Section 2.3.2. The maximum separation between any two active nodes $s$ is computed from the adopted probabilistic sensing model as explained in Section 2.2.3. $\theta$ 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 $\tau_m$ 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 $\theta$. 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 $\theta$, unless node density is not enough to achieve coverage of the whole area.

*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 $\theta$. Therefore, the coverage probability in whole triangle is at least $\theta$.

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 (2.2)). Adding the message transmission time $\tau_m$ 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 number of steps $l/(s - \delta)$ yields the worst-case convergence time of PCP.

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.

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 \( 3/6 \times 2l^2/\sqrt{3}s^2 = l^2/\sqrt{3}s^2 \). 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 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.

2.4.2 Network Connectivity Analysis

Under the disk sensing model, previous studies [30,35,38] 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 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 \( s \).

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. By contradiction, we show that the subset \( A_{k+1} \) constructed in step \( k + 1 \) is also connected. Suppose \( A_{k+1} \)
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 $u' \in A$ be the neighbor with the least distance to $v$.

We identify two cases:

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:

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

Since $\text{dist}(u, u') \leq s$, we have $\text{dist}(u', v)^2 \leq \text{dist}(u, v)^2 + s^2 - 2 \cos(u'uv)\text{dist}(u, v)s$.

The angle between lines $uv$ and $uu'$, called $u'uv$, is less than 60 degrees. Otherwise, there is another neighbor of $u$ nearer than $u'$ to $v$. Therefore, $\cos(u'uv) \geq 0.5$ and

$$\text{dist}(u', v)^2 \leq \text{dist}(u, v)^2 + s^2 - \text{dist}(u, v)s = \text{dist}(u, v)^2 + s(s - \text{dist}(u, v))$$

$$\text{dist}(u, v) > r_c \geq s \Rightarrow s(s - \text{dist}(u, v)) < 0$$

$$\text{dist}(u', v)^2 < \text{dist}(u, v)^2 \Rightarrow \text{dist}(u', v) < \text{dist}(u, v)$$

(2.3)
In other words, $u'$ is closer to $v$ than $u$ which is a contradiction.
Chapter 3

Evaluation of the Probabilistic Coverage Protocol

In the previous chapter, we presented the formal definition of probabilistic coverage problem and proposed our PCP protocol. We also proved the correctness of PCP and analyzed its performance. In this chapter, we 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 2.4. 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 [38]. Finally, we compare our protocol versus two recent deterministic coverage protocols: OGDC [35] and CCP [30].

3.1 Experimental Setup

We have implemented our PCP protocol in NS-2 [28] and in our own packet level simulator in C++. The source code for both implementations are available at [21]. Some results from the NS-2 implementation (Figs. 3.1(a) and 3.1(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 × 1km area. With that number
Figure 3.1: Validation of the PCP protocol: (a) Achieved coverage and (b) Connectivity of the activated nodes.

Figure 3.2: Savings in number of active nodes because of using the exponential sensing model for different values of $\alpha$ and $\theta$. 
of nodes, it is not possible to use simulators like NS-2. 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 [33] and [35], 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 [35]. 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.
CHAPTER 3. EVALUATION OF THE PROBABILISTIC COVERAGE PROTOCOL

Figure 3.4: Convergence time of PCP: Theory versus simulation. (a) Different values for the coverage threshold parameter $\theta$ and (b) Different values for the exponential decay factor $\alpha$.

3.2 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 $\theta$ 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 $\theta$, and the results are shown in Fig. 3.1(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 30m. We measure connectivity as the fraction of active nodes that are connected. We plot the results in Fig. 3.1(b). We show the minimum, average, and maximum values obtained from the ten iterations. Confirming our analysis in Theorem 4, our protocol achieves
CHAPTER 3. EVALUATION OF THE PROBABILISTIC COVERAGE PROTOCOL

Figure 3.5: Robustness of the PCP protocol against: (a) inaccurate node locations, and (b) imperfect time synchronization. Notice that there are two y-axes in each plot.

full connectivity when \( r_c \geq s \).

Savings Achieved by PCP. As mentioned in Section 1.1, 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. 3.2 shows the results for different values of the coverage threshold \( \theta \), and for a range of values for the sensing decay factor \( \alpha \). 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 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 \( \theta \) is reduced, which is feasible in applications that can tolerate a small probability of not detecting an event happening at a point.

Theory versus Simulation. We compare the number of activated nodes and the convergence time resulted from simulation versus our theoretical analysis in Section 2.4. Some of the results are shown in Fig. 3.3 and Fig. 3.4. The results show that the upper bounds proved in Theorems 2 and 3 are only worst-case values, and our protocol performs better on the average case.
3.3 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 3.1, 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 \([0, er_{\text{max}}]\). We vary \(er_{\text{max}}\) 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 \(er_{\text{max}}\), 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. 3.5(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. 3.5(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 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 \([0, d_{\text{max}}]\), where \(d_{\text{max}}\) is the maximum clock drift. We vary \(d_{\text{max}}\) between 0 and 500\(ms\). For every value of \(d_{\text{max}}\), 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. 3.5(b), PCP is robust against clock drifts: It achieved full coverage in all cases. 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 3.5(b). Notice that PCP converges in about 300\textit{ms} on average. This explains why the number of active sensors starts to increase for clock drifts beyond 300\textit{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\%. 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. 3.6. 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. 3.7. 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
CHAPTER 3. EVALUATION OF THE PROBABILISTIC COVERAGE PROTOCOL

Figure 3.7: Uniform energy consumption and network lifetime using PCP protocol.

Figure 3.8: Impact of multiple starting nodes on the performance of the PCP protocol: (a) convergence time and fraction of activate nodes, and (b) network lifetime.

of a node is enough for it to be active in about only five rounds. In addition, Fig. 3.7 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. 3.8(a), we change the number of starting nodes $k$ from 1 to 9 and we 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. 3.8(b), we repeat the same experiment but we measure the normalized 80%-lifetime, which is the time it takes for the coverage
Figure 3.9: Comparison between number of activated nodes by PCP and CCANS for different values of: (a) $\alpha$ and (b) $\theta$.

in the network to drop below 80%. The figure shows 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.

3.4 Comparing PCP versus another Probabilistic Coverage Protocol (CCANS)

In this section, we compare our PCP protocol against the probabilistic coverage CCANS, proposed in [38], in terms of number of activated sensors, network lifetime, and energy consumption. CCAN 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 [38].

In our comparison, we use the same sensing model for PCP with the same parameters. The parameters used for CCANS [38] are: $\xi_{th} = 1$ and $t_{max} = \tau_m$. The parameters used
CHAPTER 3. EVALUATION OF THE PROBABILISTIC COVERAGE PROTOCOL

Figure 3.10: Comparison between PCP and CCANS: (a) Total remaining energy in all nodes, and (b) Network life time.

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. \( \delta \) is computed from 2.3.2. For a uniform distribution of 20,000 nodes in a 1\text{km} \times 1\text{km} area we have \( \delta = 2 \times 1000 \sqrt{2/20,000} = 20m \)

We plot in Fig. 3.9 the average number of nodes activated by PCP and CCANS for different values of the sensing decay factor \( \alpha \) and the coverage threshold \( \theta \). 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 in Fig. 3.10(a). The figure shows 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. Finally, the lifetime of the sensor network is shown in Fig. 3.10(b), 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.
Figure 3.11: Comparison between PCP, OGDC, and CCP: (a) Total remaining energy in all nodes, and (b) Energy consumption per millisecond on a smaller time scale.

3.5 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 [35] and CCP [30] 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 $\tau_s$ 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. 3.11(a), 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. 3.11(b) 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 different energy consumption levels for transmitting, receiving, idling, or sleeping. For instance, the amount of energy consumed in receiving mode is 400 times the energy used during the sleep mode.

We plot in Fig. 3.12 the number of nodes in each state of the three 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 [35], [30]. Fig. 3.12 explains why PCP achieves the energy saving in Fig. 3.11(a). For instance, comparing OGDC (Fig. 3.12(b)) versus PCP (Fig. 3.12(a)), 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. 3.12, by finding the time at which all node states are decided. The figure shows that the convergence time of PCP and CCP is less than 500ms, while it is more than 1500ms for OGDC.
Chapter 4

Probabilistic Connectivity

In the previous two chapters, we presented the probabilistic coverage problem and our solution for it. In this chapter, we address the probabilistic connectivity problem. We summarize the related works in Section 4.1. In Section 4.2, we define the probabilistic connectivity notion and derive the probability of packet delivery in three node deployment schemes. In Section 4.3, we present our connectivity maintenance protocol. In the next chapter, we rigorously evaluate our protocol and compare it against other protocols in the literature.

4.1 Related Work

Because of its importance, the connectivity problem in sensor networks has received significant research attention. Several connectivity maintenance protocols have been proposed in the literature. We divide these protocols into two classes. In the first class, the protocol exchanges some messages to discover the connected components in the network [9,10,34]. For example, SPAN [10] maintains a list of neighbor nodes based on the received hello messages. Then, each node checks whether there exists a pair of neighbors that cannot reach each other directly or via one or two hops. If this is the case, the node becomes active; otherwise, it turns itself off to save energy. PEAS [34] and ASCENT [9] send probing messages. A node in PEAS uses probing messages to discover whether there are other working nodes in the probing range, and it goes to sleep if it finds any. PEAS uses the number of working nodes in the probing range to set the sleep duration. ASCENT [9] uses the probing messages to estimate the reachability between neighboring nodes by measuring the packet
loss rates, and uses this information to decide on which nodes should stay on. This class of protocols suffers from high communication overhead, which consumes a nontrivial fraction of node’s energy.

The second class of connectivity maintenance protocols uses information about the communication range of sensors to maintain connectivity [31, 32]. For example, the Geometric Adaptive Fidelity (GAF) protocol [32] divides the area into square cells such that all nodes inside a cell can communicate with all nodes in neighboring cells. GAF, then, keeps only one node active in each cell. These connectivity maintenance protocols rely on the assumption that the communication range is a disk, which is an over-simplification of wireless nodes in real environments [1, 17]. Our proposed protocol assumes that the communication ranges follow a probabilistic model, which is more realistic. In addition, our protocol is more general and can support the deterministic communication model as well. In this case, we compare our protocol versus the two best deterministic connectivity protocols in the literature: one from the first class, SPAN [10], and another from the second class, GAF [32].

Recently, there have been some efforts to develop realistic models for connectivity in wireless sensor networks. One approach employs a geometric random graph representation of the network to reflect the probabilistic behavior of wireless communications [6, 16]. In this case, there is an edge between each pair of nodes with a probability related to the distance between them. The work in [16] assumes that this probability is given by the log-normal shadowing model [24]. Using this model, the authors prove two theorems. The first theorem states that the graph is connected with high probability if each pair of nodes have an edge with probability at least $\frac{n}{\log n}$. The second theorem shows that the probability that the graph is $k$-connected is equal to the probability at which the minimum node degree is at least $k$. Both theorems were previously proven for geometric random graphs using the disk communication model [23]. The work in [6] derives the probability that a node in the network is isolated based on the node deployment density. The authors also show that this node isolation probability is an upper bound on the probability of having the network connected. Unlike our work, [6, 16] do not propose a distributed protocol to maintain connectivity under probabilistic communication models.

Coverage and connectivity problems are closely related to each other. When using disk model, $k$-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 [30, 35, 38]. In this thesis, we integrate our probabilistic connectivity maintenance protocol with our PCP protocol (Section 2.3) to develop an integrated protocol for probabilistic communication and probabilistic sensing models. None of the above protocols accounts for both probabilistic communication and sensing models at the same time. In addition, our protocol can provide both coverage and connectivity under deterministic models as well. Through simulation, we show that our protocol outperforms the state-of-the-art integrated coverage and connectivity protocol in the literature, CCP-SPAN [30].

4.2 Network Connectivity under Probabilistic Communication Models

In this section, we present a simple probabilistic connectivity model. Using this model, we can quantify the quality of communication between nodes in sensor networks. We start by defining a quantitative metric for communication quality. Then, we derive bounds for this metric in three node deployment schemes: triangular mesh, square mesh, and uniform.

4.2.1 Communication Quality

The main function of a sensor network is to deliver data gathered by sensors to a processing center for possible actions. Therefore, we believe that the successful data delivery between any pair of nodes in the network is a good candidate for quantifying the communication quality in a sensor network. We quantify successful data delivery from node $u$ to another node $v$ by the probability that $v$ correctly receives a packet transmitted by $u$, without any retransmission by the MAC layer. We call this probability the node-to-node packet delivery rate. We disregard MAC retransmissions because we are interested in quantifying the basic communication quality regardless of the details of the employed MAC protocol. From the sensor network design perspective, we are interested in the minimum node-to-node packet delivery rate in the network. Thus, we define the network packet delivery rate, or referred to simply as the network delivery rate, as follows.

Definition 6 (Network Delivery Rate) The network delivery rate $\alpha$ of a sensor network is the minimum packet delivery rate between any pair of nodes in the network.
CHAPTER 4. PROBABILISTIC CONNECTIVITY

Using the network delivery rate, we can define a probabilistic connectivity model for sensor networks as follows:

**Definition 7 (α-connectivity)** A sensor network is said to be α-connected if the probability of delivering a packet between any arbitrary pair of nodes (i.e., network delivery rate) is at least α, where $0 \leq \alpha \leq 1$.

In contrast to the deterministic connectivity model, the α-connectivity model provides a quantitative metric for measuring the communication quality in a sensor network. This is not only desirable, but also critical for sensor network applications that do require bounding the probability of losing a potentially important data item, such as intrusion detection systems in military applications. Furthermore, if we can determine α for a given node deployment method, we could potentially design a connectivity maintenance protocol to achieve a desired connectivity level. We derive bounds for α for common deployment methods in the following subsection. In Sec. 4.3, we propose a distributed protocol that achieves α-connectivity.

### 4.2.2 Computing Network Delivery Rates

We model a sensor network as a *weighted* graph $G(V, E)$, where $V$ is the set of all nodes, and $E$ is the set of edges between nodes. Every pair of nodes $u, v \in V$ have an edge $u \rightarrow v$ that is labeled with a packet delivery rate $p(u, v)$. $p(u, v)$ represents the probability of delivering packets from $u$ to $v$ over the direct wireless channel between them. Clearly, $p(u, v)$ depends on the probabilistic communication model used for the communication ranges of sensors. In addition, packets may flow between two nodes through multiple paths. We denote the total probability of delivering packets from node $u$ to node $v$ over all possible paths as $R(u, v)$. We refer to $R(u, v)$ as the node-to-node packet delivery rate.
The above graph representation of sensor networks is fairly general. For instance, it allows the creation of links between distant nodes. It also allows sensors to employ different communication models. It is, however, quite difficult to analytically compute the exact value of the network delivery rate $\alpha$ in this general setting. Therefore, we compute lower bounds on $\alpha$ under the following assumptions.

- All sensors use the same probabilistic communication model. This is not unrealistic assumption in many applications. For example, nodes in a surveillance application deployed in open areas could use the log-normal shadowing model [24], which captures path loss, shadowing effects, and Gaussian noise. Similarly, the same model could be used by nodes in a military intrusion detection system that are deployed on the ground at the same elevation. In addition, nodes in a forest fire detection system can all use a communication model that captures the characteristics of the surrounding environment such as the signal reflections from trees. Note that this assumption does not say that all nodes are deterministically identical, rather they follow the same probabilistic model. That is, the packet delivery rates over direct links have the same average $p = p(u, v)$.

- Links starting at the same sender node have independent delivery rates. For example, in Fig. 4.1, the node-to-node delivery rates $R(u, v)$ and $R(u, x)$ are independent. This assumption is needed to make the analysis tractable, otherwise, the analysis is not possible unless the nature of the dependence between links is completely specified. In our simulations, we do not assume independence and we verify that our results still
We only consider the delivery rates between immediate neighbors. For example, in Fig. 4.1, the direct delivery rate between nodes $u$ and $z$ is assumed to be zero. Therefore, our calculation of the network delivery rate is conservative and should be viewed as a lower bound. We notice that this is not totally unrealistic, because as the distance between nodes increases the signal fades rapidly and most wireless receivers process a signal only if its level exceeds a certain threshold.

Under these assumptions, we first derive the lower bound on the network delivery rate $\alpha$ for nodes deployed on a triangular mesh as shown in Fig. 4.1. The following theorem gives this bound.

**Theorem 5** Given nodes deployed on a triangular mesh, and the average packet delivery rate between any neighboring nodes is $p$, the network delivery rate $\alpha$ is at least $(2p - 1)/p^2$.

**Proof:** We prove this by construction. First, we begin with a triangle. Then, we expand it by adding nodes one by one to make the triangular mesh as in Fig. 4.2. Now, we find the delivery rate between source and $v$ at each step. There are two links connecting $v$ to $x$ and $y$. Therefore, the accumulated delivery rate at $v$ is $1 - (1 - pR(u, x))(1 - pR(u, y))$. Since $R(u, x)$ and $R(u, y)$ are greater than or equal to $\alpha$, we get $R(u, v) \geq 1 - (1 - p\alpha)^2$. This result is true for every pair of nodes.

Now, assume two nodes, $i$ and $j$, with the least node-to-node delivery rate. By definition, we have $R(i, j) = \alpha$. On the other hand, we have $R(i, j) \geq 1 - (1 - p\alpha)^2$ from the above discussion. Therefore, we have $1 - (1 - p\alpha)^2 \leq \alpha$, or $\alpha \geq (2p - 1)/p^2$. $\square$

Next, we derive the lower bound of the network delivery rate for nodes deployed on a square mesh.

**Theorem 6** Given nodes deployed on a square mesh, and the average packet delivery rate between any neighboring nodes is $p$, the network delivery rate $\alpha$ is at least $\min\left(\frac{p + p^2 - 1}{p^2}, p^2 - 2p\right)$.

**Proof:** Again the proof is by construction. We start with a $2 \times 2$ mesh and iteratively expand it till it contains all nodes, while maintaining the lower bound in each step. The expansion process can be done in two ways. First, by adding two nodes with the structure shown in Fig. 4.3(b): We choose any two neighbor boundary nodes $x$ and $y$ and connect
them to two new nodes $w$ and $v$. In the second type of expansion, we choose two boundary nodes $x$ and $y$ with the structure shown in Fig. 4.3(c) and connect both of them to a new node $v$. Using these two types of expansion, it is easy to show that any $n \times m$ square mesh can constructed starting from a $2 \times 2$ mesh.

Now, we derive a lower bound on the network delivery rate separately for nodes in the initial $2 \times 2$ grid and nodes added during expansion process. Then, we take the minimum of them.

1. **Initial Nodes.** Without loss of generality, assume $u$ is the source (Fig. 4.3(a)) in the initial grid. We calculate the delivery rate at $v$, $w$, and $x$. There are two paths from $u$ to $v$ with lengths 1 and 3. Therefore, the data will be delivered to $v$ with probabilities $p$ and $p^3$ respectively. The accumulated probability of delivery at $v$, $R(u, v)$, is $1 - (1 - p)(1 - p^3)$. The same argument holds for $w$ due to symmetry. On the other hand, there are two paths between $u$ and $x$ both with length 2. Hence, we have $R(u, v) = 1 - (1 - p)^2$. Since $R(u, x)$ has the minimum value among these four nodes:

$$\alpha = 1 - (1 - p)^2$$

2. **Added Nodes.** In the first expansion type (Fig. 4.3(b)), data can be delivered from the source to node $v$ through two paths $yu$ and $xuv$. The delivery rates obtained from each of these paths are $R(u, y)p$ and $R(u, x)p^2$. Since $R(u, x)$ and $R(u, y)$ are both greater than or equal to $\alpha$, $R(u, v) \geq 1 - (1 - p\alpha)(1 - p^2\alpha)$. The same analysis holds.

Figure 4.3: The square mesh construction process used in proving Theorem 6.
for $w$ due to symmetry. On the other hand, it can be easily observed that for the second expansion method, we have $R(u, v) \geq 1 - (1 - pa)^2 \geq 1 - (1 - pa)(1 - p^2\alpha)$.

Now consider the two nodes with the least node-to-node delivery rate. Call them $i$ and $j$. We have $R(i, j) = \alpha$. On the other hand, from the above analysis, we have $R(i, j) \geq 1 - (1 - p\alpha)(1 - p^2\alpha)$. Therefore:

$$\alpha \geq \frac{p + p^2 - 1}{p^3}. \quad (4.2)$$

We conclude that $\alpha$ is greater than or equal to the minimum value of (4.1) and (4.2), and the proof follows.

Finally, we extend the analysis of network delivery rate to uniform random node distribution.

**Theorem 7**  
**Given nodes deployed uniformly at random with density $\rho$, the network delivery rate $\alpha$ is at least**

$$\int_0^1 \left(1 - e^{-\rho \pi d^2} \sum_{k=0}^3 \frac{\left(\rho \pi d^2\right)^k}{k!}\right)^n dx.$$  

**Proof:**  
Instead of finding $\alpha$ directly as we did in the previous cases, we first make a probability distribution function describing the probability at which the network delivery rate of the network is at least a certain value $x$. We define $F(x) = P(\alpha \geq x)$. We solve this problem by finding the probability at which a square mesh with network delivery rate $x$ is embedded in the uniformly distributed network.

Let $d$ be the spacing between nodes in a square mesh which gives us the delivery rate of $x$. As a lower bound, it is sufficient to compute the probability that each node in the random deployment is connected to at least four nodes with distance of at most $d$. Using a result from [5, Theorem 2], in a uniformly distributed network of $n$ nodes, the probability at which no randomly chosen node has four nodes with a distance $d$ is given by:

$$\left(1 - e^{-\rho \pi d^2} \sum_{k=0}^3 \frac{\left(\rho \pi d^2\right)^k}{k!}\right)^n. \quad (4.3)$$

Therefore $F(x) = 1 - e^{-\rho \pi d^2} \sum_{k=0}^3 \frac{\left(\rho \pi d^2\right)^k}{k!}$. The expected delivery rate is given by:

$$\int_{x=0}^{1} \frac{dF(x)}{dx} - x dx = \left[F(x)x\right]_0^1 - \int_{x=0}^{1} F(x) dx = \int_{x=0}^{1} \left(1 - e^{-\rho \pi d^2} \sum_{k=0}^3 \frac{\left(\rho \pi d^2\right)^k}{k!}\right)^n dx. \quad (4.4)$$
4.3 Probabilistic Connectivity Maintenance Protocol

In this section, we present a new Probabilistic Connectivity Maintenance Protocol (PCMP), which employs probabilistic communication models. We start by presenting an overview of our protocol, followed by more details. Then, we show how our protocol can provide probabilistic coverage and connectivity at the same time.

4.3.1 Overview of PCMP

The goal of PCMP is to activate a subset of deployed nodes such that the probability of delivering packets between any arbitrary nodes in the network is at least $\alpha$, i.e., keep the network $\alpha$-connected. To achieve this goal, the protocol activates nodes to form an approximate triangular mesh. The spacing between nodes in the triangular mesh is computed to achieve the target network delivery rate. We use the bound proved in Theorem 5 and information from the adopted communication model in computing the spacing. The details of this computation are given in Sec. 4.3.2. For now, let us assume that the spacing between nodes is determined to be $d$. We chose to activate nodes on a triangular mesh for two reasons. First, it enables us to use PCMP with the deterministic connectivity model, in addition to the probabilistic model. In this case, activating nodes on the triangular mesh has been shown to be optimal in terms of number nodes activated [4]. Second, our analysis for the triangular mesh in Sec. 4.2.2 provides a simpler and tighter lower bound than the analysis for the square mesh, as confirmed by our simulations. PCMP works the same as PCP (Section 2.3) to activate nodes on vertices of a triangular mesh with spacing $d$. The spacing between nodes in the triangular mesh is determined form the adopted communication model and is based on our analysis in Sec. 4.2.2.

4.3.2 Details of PCMP

In this section, we show how the spacing between activated nodes in the triangular mesh is computed to achieve $\alpha$-connectivity. We refer to this spacing as $d_\alpha$. To make our discussion concrete, we will derive $d_\alpha$ for the widely-used log-normal shadowing model. Computing $d_\alpha$ for other communication models can be done in a similar way.
The nodes activated by our PCMP protocol form an approximate triangular mesh. The spacing between these nodes is at most \(d_\alpha\). According to Theorem 5, the network delivery rate \(\alpha\) in the triangular mesh is at least \((2p - 1)/p^2\), where \(p\) is the average packet delivery rate on a link between two neighboring nodes. That is, \(\alpha \geq (2p - 1)/p^2\). Therefore, we need:

\[
p \geq (1 - \sqrt{1 - \alpha})/\alpha
\]

(4.5)

to meet the target network delivery rate. \(p\) is related to the spacing \(d_\alpha\) through the assumed communication model. Thus, we use the communication model to compute \(d_\alpha\) to yield the required \(p\). To illustrate, consider the log-normal shadowing model widely used in network simulators, such as NS-2 [28] and OPNET [22], and several previous works, e.g., [16]. In this model, the power of the received signal \(P_r(d)\) at a distance \(d\) from a sender transmitting at power \(P_t\) is given by [24, Sec. 3.9]:

\[
P_r(d) = P_t - (PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma),
\]

(4.6)

where \(X_\sigma\) is a zero-mean random variable with Gaussian distribution, \(n\) is a constant specified by the environment, and \(PL(d_0)\) is the mean path loss measured at the reference distance \(d_0\), which is usually set to 1 m. Note that \(P_r(d), P_t,\) and \(PL(d_0)\) are all in dBm. Wireless adapters can successfully receive data if the signal strength exceeds a certain threshold, say \(\gamma\). The probability that the signal strength exceeds \(\gamma\) is [24, Sec. 3.9]:

\[
Pr[P_r(d) > \gamma] = \frac{1}{2} \left[ 1 - \text{erf}\left(\frac{\gamma - P_r(d)}{\sigma \sqrt{2}}\right) \right]
\]

(4.7)

Assuming that the signal strength does not significantly change during the transmission of a single packet, the average packet delivery rate \(p\) is given by \(p = Pr[P_r(d) > \gamma]\). Solving (4.5) and (4.7) for the spacing \(d_\alpha\), we get:

\[
d_\alpha \leq d_0 \exp \left[ (P_t - \gamma - PL(d_0) + \sigma \sqrt{2} \text{erf}^{-1}(1 - 2\frac{1 - \sqrt{1 - \alpha}}{\alpha})) / 10n \right].
\]

(4.8)

Setting the spacing between activated nodes on the triangular mesh according (4.8) will achieve the target network delivery rate under the log-normal shadowing model. Using the new spacing, the analysis in Section 2.4 remains valid for PCMP. We note that the operation of our PCMP protocol does not depend on the adopted communication model. PCMP needs only the value of \(d_\alpha\), and the protocol functions the same regardless of the model. Thus, PCMP can be used with different communication models.
4.3.3 Integrated Probabilistic Coverage and Connectivity

The problem of connectivity is closely related to the problem of coverage: It is typically requested that sensors cover the monitored area and they form a connected network for delivering collected data. In this section, we show how our PCMP protocol can be extended to achieve probabilistic coverage at the same time as probabilistic communication.

As mentioned in Section 2.2, an area is probabilistically covered with a threshold parameter $\theta$ (or $\theta$-covered) if for every point in the area the probability that a sensor detects an event occurring at that point is at least $\theta$. To achieve this $\theta$-coverage, our PCP protocol in Section 2.3 finds the least-covered point in the area, which is the point that has the smallest probability of coverage. Then, the protocol activates sensors to assure that this least-covered point has a probability of coverage more than or equal to $\beta$. Furthermore, it is shown in Appendix that the least-covered point by three sensors deployed at vertices of an equi-lateral triangle is at the center of the triangle. Therefore, we can set the spacing between the triangle vertices to meet the target $\theta$-coverage. We denote this spacing by $d_\theta$. Now the PCMP protocol has two spacing values $d_\theta$ and $d_\alpha$, where the former assures $\theta$-coverage and the latter assures $\alpha$-connectivity. PCMP sets the spacing of the approximate triangular mesh formed by the activated sensors to $\min(d_\alpha, d_\theta)$ to make the sensor network $\alpha$-connected and the area $\theta$-covered at the same time. Notice that PCMP does not require any strict relationship between the communication model and the sensing model and therefore it is fairly general. In the evaluation section, we verify that PCMP indeed achieves both $\alpha$-connectivity and $\theta$-coverage.
Chapter 5

Evaluation of the Connectivity Protocol

In the previous chapter, we presented probabilistic connectivity problem and proposed our protocol. We also presented our integrated coverage and connectivity maintenance protocol (PCMP). In this chapter, we rigorously evaluate our proposed protocol and compare it against others. We first describe our experimental setup. Then in Section 5.2, we validate our theoretical lower bounds on network delivery rate derived in Section 4.2. In Section 5.3, we analyze the performance of our protocol and show its robustness against node location inaccuracy, node failures, and imperfect time synchronization of node clocks. In Section 5.4, we show that our protocol outperforms the best two other connectivity protocols in the literature: SPAN [10] and GAF [32]. Finally, in Section 5.5, we show that our protocol can provide coverage and connectivity under deterministic models as well, and it outperforms the state-of-the-art integrated coverage and connectivity protocol in the literature (CCP-SPAN [30]).

Table 5.1: Parameters used for the log-normal shadowing communication model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path-loss exponent $n$</td>
<td>2.2</td>
</tr>
<tr>
<td>Shadowing standard deviation $\sigma$</td>
<td>4.0</td>
</tr>
<tr>
<td>Reference distance $d_0$</td>
<td>1m</td>
</tr>
<tr>
<td>Transmission power $P_t$</td>
<td>$1mW$</td>
</tr>
<tr>
<td>Reception threshold $\gamma$</td>
<td>$10^{-9}mW$</td>
</tr>
</tbody>
</table>
5.1 Experimental Setup

We use the following setup in all of our experiments, unless otherwise specified. We use NS-2 version 2.30 [28] in the simulation. We deploy 1,000 nodes uniformly at random in a 1km × 1km area. All nodes use the log-normal shadowing model given in Eq. (4.6) for radio communications. We compute the parameters for this model based on the specifications of MicaZ motes [20], and we list these parameters in Table 5.1.

We set the wireless channel bandwidth at 40 kbps. For our PCMP protocol, we have the following parameters. The round length $R$ is 100 seconds, which is much smaller than the network lifetime. The average message size is 34 bytes. The maximum value for the startup timer $\tau_s$ is set to $1/E_r$, where $E_r$ is the fraction of the remaining energy in the node.

We employ the energy model specified in [34]. In this model, the power consumption for transmission, reception, idling, and sleeping are 60, 12, 12, 0.03 mWatt, respectively. We use an initial energy of 60 Jules for each node.

We repeat each experiment 10 times with different seeds, and we report the average values over all of them. We also show the minimum and maximum values if they do not clutter the graph.
CHAPTER 5. EVALUATION OF THE CONNECTIVITY PROTOCOL

Figure 5.2: Network delivery rate in square mesh: Theorem 6 vs. Simulation. For the simulation data, we show the minimum, average, and maximum values.

Figure 5.3: Network delivery rate in uniform distribution: Theorem 7 vs. Simulation. For the simulation data, we show the minimum, average, and maximum values.
5.2 Validation of our Analysis

In this section, we validate that our lower bounds on the network delivery rates in different node deployments still hold when the assumptions made in Sec. 4.2.2 are relaxed. We also discuss potential benefits from using probabilistic communication models.

In the first set of experiments, we measure the network delivery rate from simulation and compare it against the analytical lower bound. To measure the network delivery rate, we randomly choose a node, and make it broadcast 1,000 small packets of size 8 bytes each to all other nodes. Then, we record the number of received packets at each node. The network delivery rate is the minimum number of received packets by any node divided by 1,000.

For the triangular mesh, we activate 100 nodes (out of the 1,000 deployed) and make them form a triangular mesh over the whole area. We vary the spacing between neighboring nodes \( d \) by varying the area over which the triangular mesh is constructed. Varying the spacing between nodes corresponds to varying the probability of delivering packets between neighboring nodes \( p \). We run the simulation for each value of \( d \) 10 times and we measure the network delivery rate. We also compute the network delivery rate from Theorem 5 for each value of \( d \). We repeat the experiment again, except we vary the transmission power \( P_t \) and fix everything else in the communication model. The results shown in Fig. 5.1 confirm that our lower bound is correct and conservative.

The above experiment is repeated for the square mesh and uniform deployments. Again, the results shown in Fig. 5.2 and Fig. 5.3 validate our analysis.

Finally, we discuss potential benefits of using probabilistic communication models. Connectivity maintenance protocols that rely on the disk model typically assume a conservative value for the communication range: A value within which the signal is quite strong to be received. Otherwise, the network may become disconnected according to the deterministic model. Therefore, this model may unnecessarily activate nodes. In contrast, our probabilistic \( \alpha \)-connectivity model can consider the full communication range of nodes. It also provides the network designer with a tuning knob, which is the level of connectivity \( \alpha \). Setting \( \alpha \) to higher values will result in fewer lost packets, but will require activating more nodes, and vice versa.

We design an experiment to analyze this tradeoff. We use the log-normal shadowing model and vary the spacing between nodes in the triangular mesh. We measure the network delivery rate and plot the results in Fig. 5.4(a). On the same figure, we also plot the network
Figure 5.4: Potential benefits of using probabilistic communication models over the deterministic disk model: (a) Achieved network delivery rate, (a) Saving in number of activated nodes.

delivery rate when the deterministic communication model is used with a communication range set to 100m. According to the deterministic model, the network delivery rate is zero if the spacing between nodes is more than 100m. Whereas under the more realistic probabilistic model, the network delivery rate gradually decreases. For example, if the application of the sensor network can tolerate 20% loss rate (i.e., $\alpha = 0.80$), we could approximately double the spacing between activated nodes compared to the deterministic model. This leads to significant savings in number of activated nodes, and therefore prolongs the network lifetime. The potential saving in number of activated nodes for different values of $\alpha$ is plotted in Fig. 5.4(b). Many sensor network applications could benefit from this tradeoff. For example, consider a monitoring system where nodes periodically measure temperature and humidity and report them to a processing center. Since these physical phenomena do not change suddenly, the network could tolerate some losses of the reported data, because the change will persist over several measurement periods.

5.3 Performance and Robustness of PCMP

In this section, we study the performance of our PCMP protocol and assess its robustness against inaccuracy in node locations obtained by localization mechanisms, imperfect time synchronization, and random node failures.
Figure 5.5: PCMP achieves the requested network delivery rate in all cases. The minimum, average, and maximum values are shown.

Figure 5.6: Probabilistic coverage and connectivity achieved by PCMP.
We run PCMP over 1,000 uniformly deployed nodes that use the log-normal shadowing model. We vary the requested network delivery rate $\alpha$ between 0.1 and 1.0. For each value of $\alpha$, we compute the spacing between neighboring nodes $d_\alpha$ from Eq. (4.8), and we run PCMP in the simulation with this value. We measure the achieved network delivery rate by PCMP. The results shown in Fig. 5.5 demonstrate that our protocol met the requested network delivery rate in all cases.

In Sec. 4.3.3, we described how PCMP can maintain both coverage and connectivity under probabilistic sensing and communication models. We conduct an experiment to verify this. In addition to the log-normal shadowing communication model, we use the exponential sensing model [38] for coverage. The exponential model assumes that after a threshold distance $r_s$, the sensing capacity of a node decreases exponentially fast. We fix $r_s$, and we run our protocol with various spacing $d$. Then, we measure the achieved $\beta$-coverage and $\alpha$-connectivity. We repeat for a few values of $r_s$. The results shown in Fig. 5.6 indicate that our protocol achieves both goals if the spacing is set according to our discussion in Sec. 4.3.3. For example, to achieve 0.95-connectivity, we need $d_\alpha$ to be about 175m. To archive 0.9-coverage, $d_\beta$ should be around 115m. Running PCMP with $d = \min\{d_\alpha, d_\beta\} = 115m$ achieves 0.95-connectivity and 0.9-coverage.

Next, we study the robustness of PCMP against inaccuracy in node locations. We use the same setup as before except that we add errors to node locations. We add random values in the interval $[-e_{r_{\text{max}}}, e_{r_{\text{max}}}]$ to both $x$ and $y$ coordinates of the real location of each deployed node. We vary $e_{r_{\text{max}}}$ between 0 and 20m. We compute the network delivery
rate after the protocol converges. The results indicate that the network delivery rate is always maintained as shown in Fig. 5.7(a). Therefore, PCMP is robust against location inaccuracy. There is a small cost, however, with this location inaccuracy. As shown on the same figure (notice the two y-axes), the number of activated nodes slightly increases in case of inaccurate locations. There is less than 7% increase in number of activated nodes for location errors of up to $\pm 20m$.

Exact time synchronization of nodes in a large scale sensor network is costly to achieve. We study the robustness of PCMP against the granularity of time synchronization. To do this, we add random values in the interval $[0, d_{\text{max}}]$ to the clock of each node at the beginning of the simulation. We change $d_{\text{max}}$ between 0 and 500 ms. As shown in Fig. 5.7(b), the network delivery rate is ensured even with high values of clock drift. In addition, the number of active nodes does not increase if the drift is less than the convergence time of the protocol (around 75 ms). This means that our protocol is robust against fairly large clock drifts, and thus, it needs only light-weight, coarse-grained, time synchronization schemes.

Finally, we show that PCMP is robust against random node failures. We choose a fraction $f$ of all deployed nodes to be failed within the first 200 rounds of the protocol execution, and we randomly schedule them to fail. We change the fraction of failed nodes, $f$, and plot the network delivery rate as time progresses in Fig. 5.8. The results indicate that PCMP can ensure network delivery rate even with high failure rates.
CHAPTER 5. EVALUATION OF THE CONNECTIVITY PROTOCOL

Figure 5.9: The deterministic connectivity of SPAN, PCMP and GAF.

Figure 5.10: Comparing PCMP against SPAN and GAF: (a) Energy consumption, and (b) Network lifetime.
5.4 Comparing PCMP against other Connectivity Protocols

We compare our PCMP protocol against SPAN [10] and GAF [32] protocols since they are the best and widely cited other protocols in the literature. Both protocols were described in Sec. 4.1. We use the NS-2 code for SPAN which is published by its authors at [27]. The code for GAF is included in version 2.30 of the NS-2 simulator. We use the energy model (described in Sec. 5.1) for all three protocols.

First, we verify that all protocols indeed achieve the disk-model deterministic connectivity. We check this for two different node deployment densities. We set the communication range of nodes to 100 m. Based on that, the length of GAF grid cells are set to 44 m, according to the relationship presented in [32]. To measure connectivity, we run a breadth first search to find the largest connected component of nodes. We divide the size of this component by the total number of nodes. We plot the results in Fig. 5.9. As shown in the figure, all protocols achieve 100% deterministic connectivity.

Next, we compare the three protocols against a critical metric in sensor networks: energy consumption. We fix all parameters in the simulator and run the three protocols one at a time. We periodically collect the amount of remaining energy in every deployed node. Then, we sum these values and compute the fraction of energy remained in the network with respect to the initial energy at time 0. For each protocol, we run the simulator 10 times, and for long periods (35,000 seconds). The average results are shown in Fig. 5.10(a). As the figure shows, PCMP consumes much less energy than the other two protocols. For example, after 15,000 seconds from the start, nodes under SPAN and GAF have less than 20% of their initial energy, while using PCMP nodes have 60% of their initial energy. The reasons behind the energy saving of PCMP over GAF is that PCMP activates much fewer number of nodes than GAF: The average number of active nodes under PCMP was always less than 70 in all cases, while GAF activated at least 160 nodes. Nodes in active mode consumes significantly more energy than nodes in sleep mode. On the other hand, SPAN activates slightly less number of nodes than PCMP, but it has much higher communication overhead due to the frequent exchange of hello messages among nodes.

Finally, we compare the network lifetime under the three protocols. Since these are connectivity protocols, we plot the average network packet delivery rate as the time progresses. As demonstrated by Fig. 5.10(b), our protocol extends (almost doubles) the lifetime of the network. This is because of the energy saving as described above.
5.5 Comparing PCMP against another Integrated Coverage and Connectivity Protocol

As described in Sec. 4.3.3 and verified in Sec. 5.3, our protocol achieves coverage and connectivity under probabilistic communication and sensing models. In this section, we show that our protocol can be used with deterministic communication and sensing models as well, and outperforms the stat-of-the-art protocol in this category.

We compare our protocol against an integrated protocol called CCP-SPAN [30]: It uses CCP for coverage and SPAN for connectivity. CCP is a distributed coverage protocol, which tries to deactivate nodes providing redundant coverage. To achieve this, a node in CCP checks the intersection points of its sensing circle with other circles of neighboring nodes. If all intersection points are covered, the node turns itself off. A node in SPAN, on the other hand, checks whether each pair of its neighbors can reach each other either directly or through at most two hops. If this is the case, a node turns itself off. The integrated CCP-SPAN protocol checks the two conditions to turn a node off. We use the NS-2 code of CCP-SPAN provided by its authors [8]. The sensing range used in this experiment is $r_s = 50m$, and the communication range is $r_c = 100m$.

We compare the number of activated nodes by the two protocols. Again, we fix all parameters in the simulator and we run the two protocol separately for 10 times each. We repeat the whole experiment for different node deployment densities. We plot the results...

Figure 5.11: Comparing PCMP against CCP-SPAN: (a) Number of activated nodes, and (b) Energy consumption.
in Fig. 5.11(a). The figure indicates that PCMP activates almost 50% less nodes than CCP-SPAN to provide the same coverage and connectivity.

In our last experiment, we compare the energy consumption of PCMP and CCP-SPAN as time progresses. We run both protocols separately and periodically report the total amount of energy remained in all deployed nodes normalized by their initial energy. Fig. 5.11(b) shows that the energy consumed by CCP-SPAN is about four times more than that is consumed by PCMP. This implies that sensor networks using our integrated PCMP protocol will have substantially longer lifetimes than if they were to use CCP-SPAN.
Chapter 6

Conclusions and Future Work

This chapter summarizes the conclusions of the thesis and outlines possible directions for extending it.

6.1 Conclusions

In this thesis, we considered coverage and connectivity problems under probabilistic sensing and communication models, which are more realistic than the disk 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 [38]) and the other two are deterministic (OGDC [35] and CCP [30]). 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 presented a simple probabilistic connectivity model under which we could quantify the quality of communication between nodes in wireless sensor networks. We introduced the network packet delivery rate as a quantitative metric for communication
quality. We derived lower bounds for this metric in three common node deployment schemes: triangular mesh, square mesh, and uniform. Based on the probabilistic connectivity model, we designed an integrated Probabilistic Coverage and Connectivity Maintenance Protocol (PCMP). PCMP is a fairly general protocol that can employ different probabilistic as well as deterministic sensing and communication models, with minimal configuration.

Through extensive simulations in NS-2 with nodes using the log-normal shadowing model for their radio communications, we showed that: (i) PCMP achieves the target network delivery rates; (ii) PCMP is quite robust to several factors common in real environments such as node failures, drifts in node clocks, and errors in node locations; and (iii) Probabilistic communication models expose a tradeoff between packet delivery rates and number of activated nodes, which could be exploited by sensor network designers to optimize the number of deployed nodes. This tradeoff was not possible to analyze under the traditional deterministic communication model.

We compared our protocol versus two of the best connectivity maintenance protocols in the literature: SPAN [10] and GAF [32]. Our simulation results demonstrated that our protocol significantly outperforms them in several aspects, including: number of activated nodes, energy consumption, and network lifetime. Finally, we demonstrated how our protocol can provide both coverage and connectivity under the common deterministic disk model as well. In this case, our simulations showed that our integrated protocol outperforms the state-of-the-art integrated coverage and connectivity protocol in the literature, CCP-SPAN [30], by a wide margin. To the best of our knowledge, our protocol is the first to employ both probabilistic communication and probabilistic sensing models. Therefore, our protocol is more suitable for real sensor network environments than most others in the literature.

6.2 Future Work

The work in this thesis can be extended in several directions. Some of them are summarized below:

- **Variable Sensing and Communication Models.** A useful extension to this work is to consider different sensing and communication models for nodes deployed in the same area and forming one network. Different models are needed if heterogeneous nodes are deployed, or the environmental conditions vary significantly from one location to
another. For example, some nodes could be deployed on the ground while others are deployed at different heights on a mountain.

- **Adaptive Sensing Models.** Since the environment conditions are changing during the operation of the network, the sensing ranges of nodes can dynamically change. Handling changes in sensing ranges of sensors requires: (i) developing an adaptive sensing model, (ii) finding a mechanism to discover changes in sensing ranges, and (iii) incorporating the adaptive sensing models into the probabilistic coverage protocol.

- **Adaptive Communication Models.** Like sensing models, communication models can also be affected by changes in the environment during network operation. In this case, we need to solve the same problems mentioned above. One approach is to use the communication statistics in each sensor to estimate the communication range of the sensor. These statistics are based on the measured packet loss rate in transmission to other sensors.

- **Network Delivery Rate under Realistic MAC Protocols.** One of the assumptions in our analysis of probabilistic connectivity is that the transmission of packets on different links are independent. However, packets can collide during simultaneous transmissions, leading to a lower network delivery rate. Considering the effects of the MAC protocols on network delivery rate will yield tighter and more accurate packet delivery rate estimation in the network.
Appendix A

Least-covered Point Under the Exponential Model

In this appendix, we show that under the exponential sensing model defined in (2.1) 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. 2.4):

\[
s^2(x^2y^2 + s^2z^2) + s^2(s^2y^2 + x^2z^2) + s^2(s^2x^2 + y^2z^2) - s^2(x^4 + y^4 + z^4) - s^6 = 0 \quad \text{(A.1)}
\]

Using Definition 3, the probability of sensing at point \( c \) is given by:

\[
P(c) = 1 - (1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(y-R)})(1 - e^{-\alpha(z-R)}) \quad \text{(A.2)}
\]

Notice that using (A.1), we can replace \( z \) in (A.2) 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 \( x \):

\[
\frac{\partial P(c)}{\partial x} = \alpha e^{-\alpha(x-R)}(1 - e^{-\alpha(y-R)})(1 - e^{-\alpha(z-R)}) + \frac{\partial z}{\partial x} \alpha e^{-\alpha(z-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(y-R)}) \quad \text{(A.3)}
\]
In order to find $\frac{\partial^2 z}{\partial x^2}$, we differentiate (A.1) with respect to $x$:

$$2xy^2 + 2 \frac{\partial z}{\partial x} s^2 z + 2xz^2 + 2 \frac{\partial z}{\partial x} x^2 z + 2s^2 x + 2 \frac{\partial z}{\partial x} y^2 z - 4x^3 - 4 \frac{\partial z}{\partial x} z^3 = 0$$

Again we have

$$\frac{\partial z}{\partial x} (s^2 z + x^2 z + y^2 z - 2z^3) = -xy^2 - xz^2 - s^2 x + 2x^3$$

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

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

Substituting $\frac{\partial z}{\partial x} = -1$ in (A.3) 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.

$$\frac{\partial^2 P(c)}{\partial x^2} = -\alpha^2 e^{-\alpha(x-R)} (1-e^{-\alpha(y-R)}) (1-e^{-\alpha(z-R)}) + \frac{\partial z}{\partial x} \alpha^2 e^{-\alpha(x-R)} (1-e^{-\alpha(y-R)}) e^{-\alpha(z-R)} + \frac{\partial^2 z}{\partial x^2} \alpha e^{-\alpha(z-R)} (1-e^{-\alpha(y-R)}) (1-e^{-\alpha(z-R)}) + (\frac{\partial z}{\partial x})^2 \alpha^2 e^{-\alpha(x-R)} (1-e^{-\alpha(y-R)}) + \frac{\partial z}{\partial x} \alpha^2 e^{-\alpha(x-R)} e^{-\alpha(z-R)} (1 - e^{-\alpha(y-R)})$$ (A.4)

We differentiate (A.1) twice to find $\frac{\partial^2 z}{\partial x^2}$:

$$2y^2 + 2(\frac{\partial z}{\partial x})^2 s^2 + 2 \frac{\partial^2 z}{\partial x^2} s^2 z + 2z^2 + 4 \frac{\partial z}{\partial x} xz + 2 \frac{\partial^2 z}{\partial x^2} x^2 z + 2(\frac{\partial z}{\partial x})^2 x^2 + 2s^2 + 2 \frac{\partial^2 z}{\partial x^2} y^2 z + 2(\frac{\partial z}{\partial x})^2 y^2 - 12x^2 - 4 \frac{\partial^2 z}{\partial x^2} z^3 - 12(\frac{\partial z}{\partial x})^2 z^2 = 0$$

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

$$2x^2 + 2(-1)^2 s^2 + 2 \frac{\partial^2 z}{\partial x^2} s^2 x + 2x^2 - 4x^2 + 2 \frac{\partial^2 z}{\partial x^2} x^3 + 2x^2 + 2s^2 + 2 \frac{\partial^2 z}{\partial x^2} x^3 + 2x^2 - 4 \frac{\partial^2 z}{\partial x^2} x^3 - 12x^2 = 0$$

$$4s^2 + 2 \frac{\partial^2 z}{\partial x^2} s^2 x + 2 \frac{\partial^2 z}{\partial x^2} x^3 + 2 \frac{\partial^2 z}{\partial x^2} x^3 - 4 \frac{\partial^2 z}{\partial x^2} x^3 - 20x^2 = 0$$

$$4s^2 + 2 \frac{\partial^2 z}{\partial x^2} s^2 x - 20x^2 = 0$$

$$\frac{\partial^2 z}{\partial x^2} = \frac{20x^2 - 4s^2}{2s^2 x}$$
APPENDIX A. LEAST-COVERED POINT UNDER THE EXPONENTIAL MODEL

Since the triangle is equilateral, \( s = \sqrt{3}x \).

\[
\frac{\partial^2 z}{\partial x^2} = \frac{20x^2 - 12x^2}{6x^3} = \frac{4}{3x}
\]  
\( (A.5) \)

Finally, we substitute (A.5) in (A.4).

\[
\frac{\partial^2 P(c)}{\partial x^2} = -\alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)} - \\
\alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)})e^{-\alpha(x-R)} + \frac{4}{3x}\alpha e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)}) = \\
\alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(x-R)}) - \alpha^2 e^{-\alpha(x-R)}e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)}) = \\
\alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)})(1 - e^{-\alpha(x-R)}) \right] - e^{-\alpha(x-R)} + \frac{4}{3\alpha x}(1 - e^{-\alpha(x-R)}) - \\
(1 - e^{-\alpha(x-R)}) - e^{-\alpha(x-R)} \right] \\
\frac{\partial^2 P(c)}{\partial x^2} = \alpha^2 e^{-\alpha(x-R)}(1 - e^{-\alpha(x-R)})\left[ \frac{4}{3\alpha x}(1 - e^{-\alpha(x-R)}) - 2 \right]  
\]  
\( (A.6) \)

We know that

\[
\alpha^2 > 0 \\
e^{-\alpha(x-R)} > 0 \\
(1 - e^{-\alpha(x-R)}) > 0
\]

Moreover, for typical values of \( x, \alpha \) and \( R \) the following inequality holds

\[
(1 - e^{-\alpha(x-R)}) > \frac{3\alpha x}{2}
\]

All four terms in (A.6) are positive; therefore, the second derivative of coverage probability is positive.
Bibliography


BIBLIOGRAPHY


