A pragmatic approach to area coverage in hybrid wireless sensor networks

Author:
Ahmed, Nadeem

Publication Date:
2007

DOI:
https://doi.org/10.26190/unsworks/19405

A Pragmatic Approach to Area Coverage in Hybrid Wireless Sensor Networks

by

Nadeem Ahmed

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY · AUSTRALIA

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

School of Computer Science & Engineering

University of New South Wales

2007

This thesis entitled:
A Pragmatic Approach to Area Coverage in Hybrid Wireless Sensor Networks
written by Nadeem Ahmed
has been approved for the School of Computer Science & Engineering

Supervisor: Prof. Sanjay Jha

Signature ________________ Date ________________

Joint Supervisor: Dr. Salil S. Kanhere

Signature ________________ Date ________________

The final copy of this thesis has been examined by the signatory, and I find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Declaration

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signature ________________ Date ________________

COPYRIGHT STATEMENT

AUTHENTICITY STATEMENT

Abstract

Success of Wireless Sensor Networks (WSN) largely depends on whether the deployed network can provide desired area coverage with acceptable network lifetime. In hostile or harsh environments such as enemy territories in battlefields, fire or chemical spills, it is impossible to deploy the sensor nodes in a predetermined regular topology to guarantee adequate coverage. Random deployment is thus more practical and feasible for large target areas. On the other hand, random deployment of sensors is highly susceptible to the occurrence of coverage holes in the target area. A potential solution for enhancing the existing coverage achieved by random deployments involves the use of mobility capable sensors that would help fill the coverage holes. This thesis seeks to address the problem of determining the current coverage achieved by the non-deterministic deployment of static sensor nodes and subsequently enhancing the coverage using mobile sensors.

The main contributions of this dissertation are the design and evaluation of MAPC (Mobility Assisted Probabilistic Coverage), a distributed protocol for ensuring area coverage in hybrid wireless sensor networks. The primary contribution is a pragmatic approach to sensor coverage and maintenance that we hope would lower the technical barriers to its field deployment. Most of the assumptions made in the MAPC protocol are realistic and implementable in real-life applications e.g., practical boundary estimation, coverage calculations based on a realistic sensing model, and use of movement triggering thresholds based on real radio characteristics etc. The MAPC is a comprehensive three phase proto-

col. In the first phase, the static sensors calculate the area coverage using the Probabilistic Coverage Algorithm (PCA). This is a deviation from the idealistic assumption used in the binary detection model, wherein a sensor can sense accurately within a well defined (usually circular) region. Static sensors execute the PCA algorithm, in a distributed way, to identify any holes in the coverage. In the second phase, MAPC scheme moves the mobile nodes in an optimal manner to fill these uncovered locations. For different types of initial deployments, the proposed movement algorithms consume only 30-40% of the energy consumed by the basic virtual force algorithm. In addition, this thesis addresses the problem of coverage loss due to damaged and energy depleted nodes. The problem has been formulated as an Integer Linear Program and implementable heuristics are developed that perform close to optimal solutions. By replacing in-operational nodes in phase three, MAPC scheme ensures the continuous operation of the WSN.

Experiments with real mote hardware were conducted to validate the boundary and coverage estimation part of the MAPC protocol. Extensive discrete event simulations (using NS2) were also performed for the complete MAPC protocol and the results demonstrate that MAPC can enhance and maintain the area coverage by efficiently moving mobile sensor nodes to strategic positions in the uncovered area.

Dedication

To my parents for their lifelong encouragement and support.

To my wife, Ayesha, for her patience and understanding.

To my children, Rimmal, Rohail, and Rumman, for their future.

Acknowledgements

It has been an incredible journey pursuing my PhD in the School of Computer Science and Engineering (CSE), University of New South Wales (UNSW), Australia. I am indebted to many people and organizations for their help and support during my post graduate studies.

First, I would like to thank my supervisor, Professor Sanjay Jha. I have known Sanjay since 1999, when I first joined the Network Research Lab (NRL) at UNSW to study for my Masters degree. Sanjay has always been an inspiration and provided me with generous guidance and support whenever I needed it. It has been a pleasure to work with a person of his research vision and commitment.

I thank my joint-supervisor, Dr. Salil S. Kanhere, for his tremendous guidance and for the time he spent with me discussing my research. He was instrumental in keeping my research in context through his stimulating and often penetrating discussions.

I also thank all existing and past members of the NRL for making my stay a very pleasant and memorable one. In particular, I would like to mention, Sarfraz Nawaz, Tatiana Bokareva and Tuan Le Dinh, for numerous useful discussions and general advice on my work. I also thank Andreas Reinhardt, who was an exchange student from the TUD (Technische Universitat Darmstadt), Germany. Andreas was involved with the implementation of the boundary estimation and probabilistic coverage algorithms on sensor hardware introduced in Chapters 3 and 4.

This work would not have been possible without the financial support from National ICT Australia (NICTA), National University of Sciences and Technology (NUST), Pakistan, and UNSW. For this assistance, I am very grateful. I also thank the Australian Research Council Communications Research Network (ACoRN) for providing me with travel funding.

I cannot end without thanking my family. Many thanks to my parents for their encouragement, support, and understanding. I thank my loving wife, Ayesha, who has been a great source of strength all through my graduate research.

Author's Publications

1. Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha, "Ensuring Area Coverage in Hybrid Wireless Sensor Networks", The Third International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2007). In press (Relates to Chapter 5).

2. Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha, "Efficient Boundary Estimation for Deployment of Mobile sensors in Hybrid Sensor Networks", In proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2006), pages 662-667, (3rd IEEE workshop on Localized Communication and Topology Control Protocols (LOCAN'06)), Oct 2006. (Relates to Chapter 3).

3. Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha, "Probabilistic Coverage in Wireless Sensor Networks", In proceedings of 30th Local Computer Networks (LCN'05), pages (5th IEEE workshop on Wireless Local Networks (WLN'05)), Nov 2005. (Relates to Chapter 4).

4. Nadeem Ahmed, Salil S. Kanhere, Sanjay Jha, "The Holes Problem In Wireless Sensor Networks: A Survey", Published in ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), Vol 9 No 2, pages 4-18, April 2005. (Relates to Chapter 2).

5. Tuan Le, Nadeem Ahmed, Nandan Parameswaren, Sanjay Jha, "Fault Repair Framework for Mobile Sensor Networks", In proceedings of first IEEE international conference on Communication System Software and Middleware (Commsware 2006), pages 1-8, Jan 2006.

6. Tuan Le, Nadeem Ahmed, Sanjay Jha, “Location-free Fault Repair in Hybrid Sensor Networks”, In proceedings of first ACM international conference on Integrated Internet, Ad hoc and Sensor networks (Intersense, 2006), Article no 23, Vol 138, May 30-31, 2006.

Acronyms

Ack Acknowledgement

ACoRN Australian Research Council Communications Research Network

AODV Ad hoc On Demand Distance Vector

B-node Boundary Node

CCP Coverage Configuration Protocol

CDF Cumulative Distribution Function

CEA-VFA Coverage and Energy Aware Virtual Force Algorithm

CSE School of Computer Science and Engineering

Co-Fi Coverage Fidelity Protocol

Co-Grid Coordinating Grid Protocol

CRC Cyclic Redundancy Check

DDP Desired Detection Probability

DSR Dynamic Source Routing

DSS Distributed Self Spreading Algorithm

ERS Expanding Ring Search

FFT Fast Fourier Transform

GPS Global Positioning System

GPSR Greedy Perimeter Stateless Routing

Heu-Max-E-I Heuristic with Maximize Energy Introduced

Heu-Min-E Heuristic with Minimize Energy

ID Identification

IDCA Intelligent Deployment and Clustering Algorithm

ILP Integer Linear Program

ISM Intermittent Simulated Movements

LQI Link Quality Indicator

MANET Mobile Ad-hoc Networks

MAPC Mobility Assisted Probabilistic Coverage

MC-MC Most Constrained Minimally Constraining

MNN Mobile Node Neighbor List

MSC Maximum Set Cover

NICTA National ICT Australia

NINRA Neighbors in Nominal Range Area

NRL Network Research Lab

NS2 Network Simulator 2

NUST National University of Sciences and Technology

OGDC Optimal Geographic Density Control

Opt-Max-E-I Optimized with Maximize Energy Introduced

Opt-Min-E Optimized with Minimize Energy consumption

PCA Probabilistic Coverage Algorithm

PT Probability Table

RNG Relative Neighborhood Graph

RSSI Received Signal Strength Indicator

SM Simulated Movements Approach

SMART Scan Based Mobility Assisted Sensor Deployment

SNR Signal to Noise Ratio

TinyOS Tiny Operating System

TOSSIM TinyOS Simulator

UAV Un-manned Aerial Vehicle

UNSW University of New South Wales

VFA Virtual Force Algorithm

WMEWMA Windowed Mean Exponentially Weighted Moving Average

WSN Wireless Sensor Networks

Contents

Chapter

1 Introduction 1

1.1 Protocol Overview . . . 5

1.2 Thesis Contributions . . . 8

1.3 Thesis Organization . . . 9

2 Background 11

2.1 Related Work . . . 11

2.2 Boundary Estimation in WSN . . . 11

2.3 Coverage Calculations . . . 16

2.4 Coverage Enhancement Using Mobile Sensors . . . 22

2.4.1 All Mobile Sensor Networks . . . 22

2.4.2 Hybrid Sensor Networks . . . 26

2.5 Coverage Repair and Maintenance . . . 29

2.6 Summary . . . 31

3 Distributed Boundary Estimation 32

3.1 Introduction . . . 32

3.2 Boundary Estimation by Static Nodes . . . 34

3.2.1 Boundary Estimation Algorithm . . . 34

3.2.2 Simulations . . . 39

3.2.3 Experiments . . . 41

3.3 Boundary Landmarks . . . 52

3.3.1 Locating the Posts . . . 53

3.3.2 Replicating the Information of Selected Posts . . . 54

3.3.3 Discussion . . . 54

3.4 Summary . . . 56

4 Probabilistic Coverage in Sensor Networks 57

4.1 Introduction . . . 57

4.2 Technical Preliminaries . . . 59

4.3 Probabilistic Coverage Algorithm . . . 62

4.3.1 The Algorithm . . . 65

4.3.2 Identifying Deployment Points . . . 72

4.3.3 Request Aggregation . . . 74

4.4 Simulation Study . . . 75

4.5 Experiments . . . 79

4.5.1 Acoustic Experiments . . . 79

4.5.2 Implementation of PCA on Tmote Sky . . . 81

4.6 Discussion . . . 83

4.7 Summary . . . 83

5 Coverage Enhancement Using Mobile Sensor Nodes 85

5.1 Introduction . . . 85

5.2 Coverage and Energy Aware Virtual Force Algorithm (CEA-VFA) 88

5.2.1 Coverage Awareness . . . 90

5.2.2 Energy Awareness . . . 92

5.2.3 Choice of Thresholds, Th_{push} and Th_{pull} . . . 93

5.2.4 Boundary Considerations . . . 94

5.2.5 Per Round Synchronization . . . 95

5.2.6 Oscillation Control 97
5.3 Virtual Force Algorithm with Simulated Movement 98
5.4 Performance Evaluation 101
5.4.1 A Centralized Optimization 101
5.4.2 Simulations Results 102
5.5 Summary 109

6 Coverage Maintenance in Wireless Sensor Networks 110
6.1 Introduction 110
6.2 Proactive Maintenance: Replacing Low Energy Nodes 112
6.2.1 Problem Formulation 113
6.2.2 Heuristic Solutions 114
6.3 Reactive Maintenance: Replacing Dead / Faulty Nodes 120
6.4 Simulation Results 122
6.5 Summary 125

7 Conclusion and Future Research Directions 127
7.1 Conclusion 127
7.2 Future Research Directions 129
7.2.1 Relaxing Simplifying Assumptions 129
7.2.2 In-situ Calibration 129
7.2.3 Extended Experimental Studies 130
7.2.4 Topology Control 130
7.2.5 Multiple k Coverage 131
7.3 Summary 131

Bibliography 132

List of Tables

Table

1.1 Default specifications for sensor hardware. 2

2.1 Comparison of solutions to area coverage problem assuming static networks 20

2.2 Comparison of mobility based solutions to area coverage problem. 28

3.1 Simulation setup parameters 40

4.1 A sample probability table (PT) 62

4.2 Simulation setup parameters. 75

5.1 Simulation setup parameters. 102

List of Figures

Figure

1.1 Example of a coverage hole. 3

1.2 Building blocks of MAPC protocol. 5

2.1 Local minimum phenomenon in greedy forwarding. 13

2.2 Voronoi diagram. 23

3.1 Building blocks of MAPC protocol. 34

3.2 Right hand rule. 36

3.3 Selection of B-nodes. 38

3.4 Local minima. 38

3.5 Simulations: Boundary estimation for random, street and river topologies. 40

3.6 Calibration experiment setup. 42

3.7 Packet loss vs distance. 43

3.8 RSSI vs distance. 43

3.9 LQI vs distance. 44

3.10 Temporal packet loss in different directions. 45

3.11 Temporal RSSI values in different directions. 46

3.12 Anomalies due to (a) Asymmetric links (b) Unexpected long links. 48

3.13 Experiments: Boundary estimation for random, street and river topologies. 50

3.14 Virtual boundary with Posts. 55

4.1 Building blocks of MAPC protocol. 59

4.2 Change in detection probability with distance(m). 61

4.3 Detection probabilities. 63

4.4 Comparison of probabilistic and unit disk models. 64

4.5 Neighbor contribution to coverage 66

4.6 Perimeter coverage using PCA 67

4.7 Coverage calculations. 70

4.8 Identifying deployment point 72

4.9 Simulation results. 75

4.10 Coverage provided by unit disk and probabilistic models. 76

4.11 Effect of granularity on performance of PCA. 77

4.12 Coverage by unit disk and PCA. 77

4.13 2m: Time domain and power spectral analysis. 79

4.14 10m: Time domain and power spectral analysis. 79

4.15 Probability distribution function. 80

4.16 Experimental results: Uniform topology 82

4.17 Experimental results: Sparse topology 82

5.1 Building blocks of MAPC protocol. 86

5.2 Contention resolution mechanism. 92

5.3 Change in communication probability with distance. 94

5.4 Cut through paths followed by ISM2 and ISM3. 100

5.5 Total distance moved: Basic VFA vs CEA-VFA. 103

5.6 Total distance moved by mobile sensors. 104

5.7 Empirical error CDF for normal and island distributions. 105

5.8 Topology distribution achieved with CEA-VFA. 106

5.9 Topology distribution achieved with ISM3. 106

5.10 Topology distribution achieved with ISM6. 107
5.11 Percentage of area covered and energy consumed. 108

6.1 Building blocks of MAPC protocol. 112
6.2 Mobile node discovery process. 118
6.3 Special case of reactive maintenance. 121
6.4 Proactive maintenance: Minimize energy consumption. 123
6.5 Proactive maintenance: Maximize energy introduced. 123
6.6 Proactive maintenance: Performance of assignment algorithms. . . 124
6.7 Reactive maintenance: Performance of assignment algorithms. . . 124
6.8 Reactive maintenance: Minimize energy consumption. 125
6.9 Reactive maintenance: Maximize energy introduced. 125

Chapter 1

Introduction

Recent advancements in Micro-Electro-Mechanical Systems (MEMS) have augmented research in Wireless Sensor Networks (WSN). A wireless sensor network is composed of tiny sensor nodes each capable of sensing some phenomenon, doing some limited data processing and communicating wireless with each other [1]. These tiny sensor devices are deployed in large numbers, in close proximity to the phenomenon, for obtaining unprecedented high-precision, high-fidelity sensory data. WSN are typically designed to operate unattended and without the need for any pre-existing infrastructure. Nodes in a sensor network collaborate to form an ad-hoc network capable of reporting the sensed phenomenon to a data collection point called sink or base station. WSN have the potential to influence our daily lives to a great extent. These networked sensors have many potential civil and military applications i.e., they can be utilized for health applications [2] [3], military applications [4] [5], surveillance and intrusion detection [6], habitat monitoring [7] [8], environmental monitoring [9] [10], structural health monitoring [11] [12], animal tracking [13] [14], education [15], traffic control, inventory management in factory environments, underground sensing etc. [16] [17] [18].

These myriad of applications present various design, operational, and management challenges for wireless sensor networks. The challenges become even more demanding if we consider the constraints of wireless sensor networks such

	Mica2 [19]	**MicaZ [19]**	**Tmote Sky [20]**	**Intel Mote 2 [19]**
Processor (MHz)	7.37	7.37	7.37	13–416
Data Rate (Kbps)	38.4	250	250	250
Radio	ChipCon CC1000 (433, 868, 916 MHz)	ChipCon CC2420 (2.4 GHz)	ChipCon CC2420 (2.4 GHz)	ChipCon CC2420 (2.4 GHz)
Radio Range	approx. 125 m	75-100 m (outdoors), 20-30 m (indoors)	125 m (outdoors), 50 m (indoors)	30 m (with integrated antenna)
Memory	128 Kbytes program, 4 Kbytes configuration, and 512 Kbytes Measurement	128 Kbytes program, 4 Kbytes configuration, and 512 Kbytes Measurement	48 Kbytes program, 16 Kbytes configuration, 10 Kbytes RAM, and 1024 Kbytes Measurement	256 Kbytes SRAM, 32 Mbytes Measurement Flash and 32 Mbytes SDRAM

Table 1.1: Default specifications for sensor hardware.

as low processing power and bandwidth, limited memory and battery life, and short radio ranges. Table 1.1 lists some of the sensor hardware with their default specifications. The specifications illustrate that these low power wireless devices are indeed resource improvised.

Wireless sensor networks differ from ad-hoc networks in several ways. One of the distinguishing features is the introduction of the sensing component in sensor networks. A node[1] in a sensor network is thus performing two demanding tasks simultaneously, sensing and communicating. To accomplish these tasks, we normally assume that the node not only performs required sensing of the phenomenon but is also able to communicate with neighbors for onward transmission of the sensed data to sink. But this assumption is often not true in real world deployment scenarios.

[1] In this thesis, we use the terms node and sensor interchangeably

Consider a target tracking application employing WSN. The objective is to monitor the whole of a target area, by deploying wireless sensor nodes, to detect and track the movements of certain objects (e.g, enemy tanks, personnel etc). Assume that this application gathers acoustic sensory data from the deployed nodes for tracking purposes. One of the pre-requisite for success of this tracking application is that every point in the target area should be under direct observation of one of the deployed sensor nodes. As each sensor node has a limited sensing coverage (area of influence in which a node can sense), the target area that is supposed to be 100% covered by the densely deployed nodes may have **coverage holes**, areas not covered by any node. The network cannot detect and track in these grey areas of sensing and thus fails to achieve its objectives. Figure 1.1 shows an example of a coverage hole. Circles around a sensor location represent the sensor coverage. The grey area is not covered by any sensor.

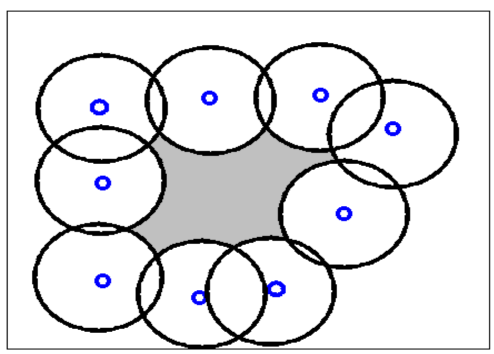

Figure 1.1: Example of a coverage hole.

Providing adequate sensing coverage of the target area is thus of paramount importance for proper functioning of a WSN. Adequate coverage of a target area can be achieved by carefully deploying sensor nodes at pre-computed locations in the topology. This deterministic deployment is only feasible for small scale networks, operating in an accessible area with known environment. For unknown, hostile, and harsh environments like enemy territory in battlefields, fire or chemical spills, the network cannot be carefully deployed to a predetermined regular topology. Random (possibly aerial) deployment of sensor nodes is a solution in such scenarios. But in this case, it is very difficult to achieve adequate coverage of the target area. This may happen due to the presence of obstacles, sloping grounds like hills, presence of strong winds or dense forestation during aerial deployment etc. A randomly deployed static sensor network, therefore, cannot guarantee adequate area coverage. Mobility capable sensors are a potential solution to enhance the existing coverage due to non-deterministic deployment of a WSN. As mobile sensors are costlier than their static counterparts, an all mobile network is economically not feasible. This dissertation employs a hybrid sensor network, where only a few of the nodes are mobility capable. This helps in reducing the overall cost of the network while providing the necessary re-configuration capabilities to reach an adequate coverage level.

Loss of coverage in the target area can also occur during the operational lifetime of a WSN. This may happen due to hardware failure, physical damage, or depletion of a node's energy resources. The mobile nodes in a hybrid network also provide an opportunity to exercise self-maintenance of the WSN. These mobile nodes can replace any faulty, damaged, or low energy node for continued network performance at the desired coverage level. Hybrid sensor networks thus provide an effective and cost-efficient self-organization capability that is vital for successful, prolonged network operation.

This thesis deals with the area coverage problem that is defined as follows. Given a set of sensors and a target area, the area is adequately covered if every point in that target area is covered by at least one sensor. There are several aspects to the area coverage problem for hybrid sensor network that need careful consideration:

- How to evaluate area coverage provided by a deployment with an objective to identify coverage holes and areas of low coverage;
- How to increase the existing coverage to the desired level by deploying mobile sensor nodes;
- How to maintain the coverage at the desired level during the lifetime of the WSN.

This thesis proposes a comprehensive Mobility Assisted Probabilistic Coverage (MAPC) protocol that deals with all these coverage related issues in entirety, from deployment to the operation of the hybrid WSN.

1.1 Protocol Overview

Our system assumes a hybrid network consisting of a large number of static sensors deployed in a non-deterministic manner. We also assume that a few mobile sensors are available for plugging the coverage holes.

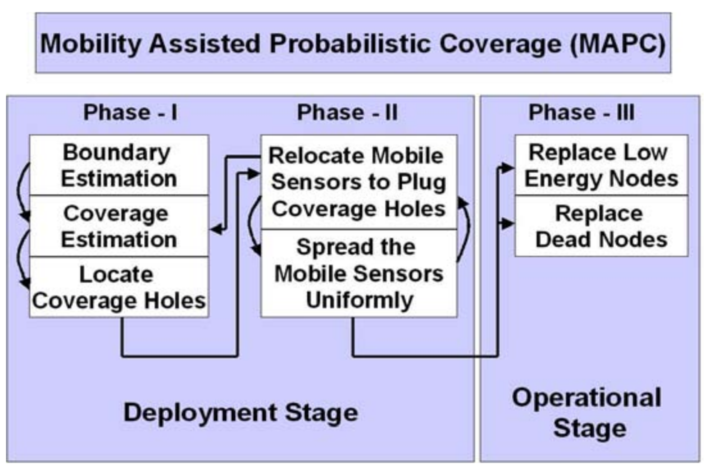

Figure 1.2: Building blocks of MAPC protocol.

Figure 1.2 illustrates the building blocks of the MAPC protocol. The protocol works in three distinct phases. Phase I and II are executed during the deployment stage of a WSN while phase III is executed during the operational lifetime of the network. Phase I consists of two sub-tasks; boundary estimation and coverage estimation. Both these tasks are performed by static nodes only. Static nodes estimate the boundary of the unknown region in Phase I. Our approach differs from the existing deployment schemes in that we assume that sensors cannot detect the physical boundary of the region in outdoor environments. This assumption is consistent with the real world capabilities of existing sensor hardware. For boundary estimation, this thesis proposes a distributed boundary estimation algorithm based on the geometric right hand rule. Boundary estimation results in the identification of **virtual boundary** formed by nodes lying on the perimeter of the deployed topology.

Once the virtual boundary of the region has been identified, static nodes estimate the area coverage. The widely used binary detection model assumes that the sensing coverage of a sensor node is uniform in all directions, often represented by a unit disk. However, in reality, the sensing capabilities of a sensor are often affected by environmental factors. In particular, for range-based sensor modalities such as acoustics, radio, etc, the signal strength of the triggering signal decays as a function of distance. This implies that the detection capabilities of these sensors would exhibit similar distance dependant characteristics as opposed to a uniform sensing range. The sensing capabilities of networked sensors are thus affected by environmental factors in real deployment and it is imperative to have practical considerations at the design stage in order to anticipate this sensing behavior. In an effort to employ more realistic models in the computation of area coverage, this thesis proposes the **Probabilistic Coverage Algorithm (PCA)**, that takes into account the probabilistic sensing behavior of sensor nodes.

Mobile nodes are introduced in the topology in phase II. They are assumed initially concentrated at one or more points on the boundary of the target area.

Phase II manages iterative relocation of the mobile sensors. For this phase, we propose a set of coverage and energy aware variants of **Virtual Force Algorithm (VFA)**. These algorithms work in rounds and manage the mobile node relocation that serves a dual purpose. Firstly, this relocation increases the coverage during deployment by allowing the mobile nodes to fill in the coverage holes with minimal expenditure of energy. Secondly, the additional mobile sensors are uniformly spread in the target area for further discovery of coverage holes. The movements of the mobile nodes are controlled by novel thresholds based on real radio characteristics. This ensures that the nodes can communicate with each other, with high probability of successful transmission, during the round-by-round movements. The mobile nodes are kept in the useful target area bounded by the virtual boundary identified by the B-nodes (phase-I). The mobile nodes relocation in phase II thus results in increase in area coverage during the deployment stage.

In phase III, the MAPC relocates the available mobile sensors during the continuous operation of the sensor network. This is to preempt any possible loss of coverage due to nodes depleting their batteries (proactive maintenance) and repair any coverage loss due to dead sensors, e.g., nodes with hardware failure or physical damage (reactive maintenance). The coverage maintenance problem, in this thesis, has been formulated as an Integer Linear Program (ILP) and implementable heuristics have been proposed that perform close to optimal solutions. We demonstrate through extensive simulations that MAPC protocol enhances the area coverage by efficiently moving mobile sensor nodes to strategic positions in the uncovered area. The individual phases of the MAPC protocol are elaborated in separate chapters of this thesis.

1.2 Thesis Contributions

The main contributions of this thesis is the design and evaluation of MAPC, a distributed and comprehensive three phase protocol, that can effectively estimate, enhance, and maintain the area coverage in a hybrid sensor network. The primary contribution is a pragmatic approach to sensor coverage and maintenance that we hope would lower the technical barriers to its field deployment. Most of the assumptions made in the MAPC protocol are realistic and implementable in real-life applications, for example, practical boundary estimation, coverage calculations based on a realistic sensing model, and use of movement triggering thresholds based on real radio characteristics etc. To our knowledge, no such comprehensive solution has been presented in the literature that addresses the deployment, coverage enhancement, and coverage maintenance issues for a hybrid sensor network in its entirety.

This dissertation makes the following four specific contributions. The first contribution is the proposal and evaluation of a practical boundary estimation scheme that identifies the nodes lying on the outer boundary of a deployed network topology. The scheme employs the geometric right hand rule for identifying the boundary nodes but, on practical considerations, does not assume an idealistic unit disk based communication graph. We have conducted extensive experimentation on real sensor hardware and simulations to validate the boundary estimation scheme. The boundary estimation scheme has also been made robust to the existence of radio irregularities, a characteristic of real radio behavior.

The second contribution of this thesis is the proposal of a novel coverage estimation algorithm, PCA, that evaluates the area coverage in WSN. PCA takes into account the realistic variations in sensing behavior associated with range based sensors. This is in contrast to the simplistic binary unit disk sensing model, where an event occurring within a circular disk is always assumed detected with probability of one while any event outside this circle of influence is assumed not detected. PCA takes advantage of the sensing contributions from neighbors for accurate estimation of a node's area coverage. Simulation and experimental studies show that the PCA can accurately estimate the area coverage in the region.

The third contribution of this dissertation falls in the area of mobile nodes deployment. This thesis proposes a set of integrated deployment and coverage enhancement algorithms that are based on the virtual force algorithm. The algorithms use movement triggering thresholds that are based on real radio characteristics. These algorithms not only deploy mobile nodes from an initial concentrated position but also increase coverage by plugging the coverage holes in the topology. For different types of initial deployments, the proposed movement algorithms consume only 30-40% of the energy consumed by the basic virtual force algorithm.

Finally, the fourth contribution addresses the problem of coverage maintenance in WSNs by considering coverage loss due to damaged and energy depleted nodes. The coverage maintenance problem has been formulated as an Integer Linear Program and implementable heuristics are developed. The proposed coverage maintenance scheme is a good heuristic approximation of centralized optimal assignment that is not feasible for large scale sensor networks.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. We introduce background knowledge and discuss related work in Chapter 2. A survey of the existing literature is presented that provides proper context to the work proposed in this thesis.

In Chapter 3, we discuss a distributed boundary estimation algorithm that identifies the nodes lying on the outer boundary of a randomly deployed WSN. We evaluate the performance of the proposed algorithm through discrete event simulations and experiments on real sensor hardware. In Chapter 4, we introduce a novel coverage estimation algorithm, PCA, and show that the coverage estimated by PCA is more realistic compared to the existing algorithms assuming idealistic detection models.

We consider the problem of coverage enhancement using mobile nodes for a hybrid sensor network in Chapter 5 and propose an energy efficient integrated deployment approach that solves both the mobile sensor deployment and the coverage enhancement issues. In Chapter 6, we discuss the coverage maintenance issues and propose two heuristic solutions that are shown to perform close to optimal solutions.

Finally, we conclude our work and identify a number of future work directions in WSN coverage related issues.

Chapter 2

Background

2.1 Related Work

This dissertation proposes an integrated solution to the area coverage problem that spans from deployment to maintenance phases of the lifetime of a wireless sensor network. In this chapter, we survey the literature that is most relevant to the work covered in this dissertation. We discuss strengths as well as shortcomings of the existing protocols and show how our proposed work is different from previous schemes. In the next few sections, we will survey background and related work in each of the following areas:

- Boundary estimation in wireless sensor networks;
- Coverage estimation in wireless sensor networks;
- Coverage enhancement using mobile sensors;
- Coverage maintenance in wireless sensor networks.

2.2 Boundary Estimation in WSN

Boundary estimation in sensor networks is usually categorized as event boundary estimation and topological boundary estimation. Event boundary estimation takes sensor measurements from a deployed sensor network and determines the boundary between fields of homogeneous sensor measurements e.g., mean values for the measurements [21]. The output is the boundary delineation between homogeneous sensing regions based on the observed sensor data. On the other hand, topological boundary estimation schemes refer to the detection of nodes lying on the topological boundary of a region of interest. As the focus of our work is on topological boundary estimation, we only discuss related research efforts.

Application of the geometric right hand rule for graph traversal has been suggested in various research efforts such as [22], [23], [24], [25], and [26] etc. Jarvis [22] suggested a centralized approach to find the convex hull of a set of points. The algorithm starts by selecting an origin point lying outside the point set. It then goes through the set of all points and finds a point in the point set such that angle measured clockwise from the origin point to this point is maximum. The origin point is then replaced by this point and the process is repeated until we end up at the original origin point. All boundary nodes are thus traversed in a counter clockwise order. The algorithm is simple and effective when the number of points on the convex hull is relatively small compared to the total number of points in the points set.

Stajmenovic et al. [23] and more recently Liu et al. [24] proposed the quorum based location service. Location updates are sent to nodes in north and south directions (columns) while discovery is performed in east and west directions (rows). Intersection of rows and columns is guaranteed by adding the outer face of the network (boundary nodes) to both of them. They proposed the use of the right hand rule on edges of Gabriel graph [27] to locate the boundary nodes that act as location servers for the deployed ad hoc network.

From the routing perspective, the right hand rule is used to recover from local minimum phenomenon often faced in geographic greedy forwarding. Forwarding here is based on destination location. In Figure 2.1, a node x tries to forward the traffic to one of its 1-hop neighbors that is geographically closer to the destination than the node itself. This forwarding process stop when x cannot find any 1-hop neighbor closer to the destination than itself and the only route to destination requires that the packet moves temporarily farther from the destination to b or y. This special case is referred to as local minimum phenomenon.

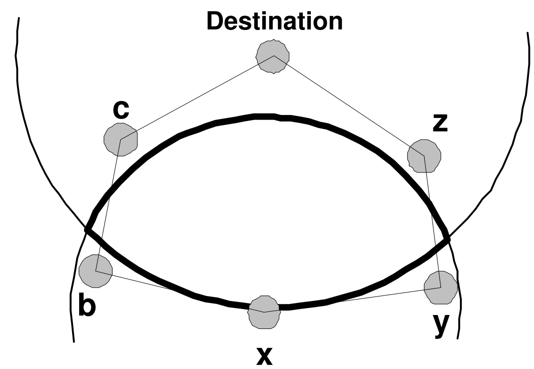

Figure 2.1: Local minimum phenomenon in greedy forwarding.

Karp et al. [28] proposed the Greedy Perimeter Stateless Routing (GPSR) for MANETs. GPSR recovers from routing holes due to local minimum phenomenon by using perimeter routing mode. In perimeter routing the right-hand rule is used. GPSR proposes either Relative Neighborhood Graph (RNG) [29] or Gabriel graph to get a planar network graph with no crossing edges. Bose et al. [25] proposed the FACE-1 and FACE-2 routing algorithms to guarantee packet delivery in MANETs. The suggested solution is also based on deriving a connected planar subgraph by using Gabriel graph and then traversing the edges of the graph using right-hand rule.

Fang et al. proposed the BoundHole Algorithm [26] using the right-hand rule to identify nodes on the boundary of geometric holes. They defined stuck nodes as nodes in the topology where packets can possibly get stuck in greedy multi-hop forwarding due to local minimum and proposed a distributed BoundHole algorithm to avoid the routing hole. The BoundHole algorithm uses the right-hand rule, after each node has identified its stuck direction, to mark the boundary of the routing hole.

Boundary estimation algorithms that do not use the right hand algorithm are presented in [30], [31], [32], [33], and [34]. All of these research efforts are focused on boundary recognition and extraction of the topology without assuming that location information is available. Fekete et al. [30] proposed an algorithm that identifies boundary nodes based on the degree of neighbors as boundary nodes are expected to have much less degree of neighbors than the non-boundary nodes. They categorized boundaries as outer and inner boundaries. Outer boundary separates the region from the unbounded portion of the outside world while inner boundaries separate the nodes from holes in the region. They assumed a dense deployment and used a combination of stochastic, topology, and geometry. This algorithm has the additional overhead of construction of a spanning tree that is utilized for computing the neighborhood size histogram. In a similar work by the same authors, Fekete et al. [31] used restricted stress centrality to identify nodes near the boundary by constructing shortest paths containing a vertex of the graph. The algorithm assumes a very dense deployment.

More recently, Kroller et al. [32] proposed an overall framework for self organization by using a combination of topology and geometry to determine the boundary nodes and the topology of the whole network. The boundary is estimated using special combinatorial structures called flowers and augmenting cycles. The proposed algorithm is distributed and deterministic. The authors assumed a street network with dense deployment and used Quasi-unit disk graph derived from the communication graph to estimate the network topology.

S. Funke [33] identified nodes near the boundary of the sensor field as well as near hole boundaries by constructing iso-contours based on the hop count from a root node. Broken contours indicate either outer boundary nodes or nodes at boundary of a hole. The algorithm is based on the connectivity information of the communication graph derived by unit disk graph assumption. They also assumed a dense deployment and used four beacon nodes that flood the network to construct the hop count information.

Wang et al. [34] also used the hop count measure from a root node to identify boundary nodes based on the observation that holes in a sensor field create irregularities in hop count distances. The algorithm is distributed and does not assume that the communication graph follows the unit disk or quasi unit disk graph model. Their work requires network wide flooding to construct shortest path tree and to disseminate information of inner boundaries to all nodes but can work with lower node density than other schemes.

The boundary estimation work presented in this dissertation is different to these proposed approaches in several ways. Our proposed boundary estimation algorithm is lightweight (and hence more suited to the energy constrained wireless sensor networks) compared to the family of computationally intensive algorithms that assume no location or inter-node distances are available. Our proposed algorithm is inspired by the Jarvis walk proposed in [22]. The Jarvis walk is a centralized approach to find the convex hull of a set of points in plane while this thesis proposes a simple and distributed boundary estimation mechanism that does not require flooding of the deployed network for gathering the topology information.

Authors in [23] and [24] also proposed the use of right hand rule as an extension of the quorum based location update mechanism. They used the FACE routing algorithm [25] on the edges of a planar Gabriel graph to locate the boundary nodes. It was shown by Kim et al. [35] that real radio often violates the unit disk graph assumption resulting in anomalies in the planarization process. Planar graphs such as the Gabriel graph that are derived from unit disk graphs may result in crossing edges due to radio irregularity. Moreover, the maintenance of planar graph at each node introduces additional overhead. While all the nodes maintain the planar graph, for boundary estimation this information is only required by the boundary nodes. Thus, in order to have a practical and efficient approach, we do not assume a planar graph rather, we introduce explicit checks to detect crossing edges among the current communication neighbors.

2.3 Coverage Calculations

The coverage problem has been interpreted in a variety of ways in existing literature. Coverage has been considered in terms of maximal support and breach paths, exposure, quality of surveillance and area coverage etc. [36] [37]. Area coverage checks whether every point in the target area is at least covered by a sensor node such that there is no coverage hole in the target area. As the focus of this thesis is on the area coverage problem with random deployments, we limit the discussion here to related work. For each of the protocols discussed in this section, a further differentiation is made according to whether it is designed to address single coverage or the multiple coverage requirement.

For a static sensor network (without mobility support), several topology /density control protocols have been proposed. These protocols select a minimal number of on-duty nodes that are active at any time out of the available densely deployed nodes. This node scheduling is feasible as long as no coverage holes appear due to nodes being turned off for energy savings. Huang et al. [38] proposed polynomial-time algorithms to verify whether every point in the target area is covered by at least the required number of nodes. Algorithm **k**-UC assumes a uniform circular sensing disk while **k**-NC assumes a non-disk sensing range for each sensor node. The node calculates its perimeter coverage by finding the sector of its coverage area occupied by the neighbor sensing range. The node thus verifies whether its whole perimeter is covered by existing neighbors to the required degree or not. To detect coverage holes, the authors suggest a central controller entity.

Wang et al. [39] presented the Coverage Configuration Protocol (CCP) that can provide flexibility in configuration of sensor network to self-configure for different degrees of coverage. Given a requested coverage degree K_s, a sensor node is scheduled to sleep if every location within its coverage range is already K_s covered by other active nodes in its neighborhood. For cases when $R_c < 2R_s$, CCP does not guarantee connectivity along with the coverage. Authors have proposed combining CCP with an existing connectivity maintenance protocol, SPAN [40], that provides the communication connectivity.

Yan et al. [41] proposed a distributed density control algorithm capable of providing differentiated coverage based on different requirements in different parts of the deployed sensor network. The algorithm is based on time synchronization among the neighbors. In the sensing phase, comprising of several rounds of equal duration, each node divides its whole area into grids and advertises its reference point and start and stop time, defined with respect to that reference point. The node sorts the neighbors covering a particular grid in ascending order of their reference points in a round. Based on the time sequence obtained, a node can decide its on-duty time such that the whole grid still gets the required degree of coverage. The results from all the covered grids are merged to find the adopted duty schedule for the node.

For single coverage requirement, Slijepcevic et al. [42] proposed a centralized heuristic solution to the NP-hard Set K-COVER problem called most-constrained minimally-constraining (MC-MC) algorithm. They divided the sensors into disjoint sets where each set can monitor the whole area independently. One such set is thus active at any time. In contrast, Cardei et al. [43] proposed heuristic solutions to the NP-complete Maximum Set Cover (MSC) problem. They assumed that the set of nodes need not be disjoint and that the sets can operate in unequal time intervals. Each set only need to cover targets with known locations.

Zhang et al. [44] proposed the Optimal Geographical Density Control (OGDC) protocol. This protocol tries to minimize the overlap of sensing areas of all sensor nodes for cases when $R_c \geq 2R_s$. A node with sufficient power is randomly chosen to start the process of node selection. This starting node broadcasts a power-on message. Nodes, on reception of this power-on message, check their power level and the existing coverage of area under their sensing range. If sufficient power is available, and the area is not fully covered, the node adds the starting node as the neighbor, sets its state to "ON" and broadcasts the power-on message again. This process continues with slightly different behavior for power-on messages received from starting and non-starting nodes. Simulation results presented by the authors shows that OGDC protocol cannot always preserve the original coverage of the network completely.

The issue of coverage with different sensing capabilities has been addressed in [45] by Tian et al. The authors discussed the topology control for both uniform and different sensing ranges. A node decides to turn off after discovering that its neighbors (sponsors) can completely cover its sensing area. The node credits the sponsored area based on sectors instead of the actual crescent formed in its sensor range. Once the sponsored area becomes equal to the sensing area under the node, the node qualifies as a candidate to be switched off. To avoid the possibility of multiple neighbors turning off and creating a coverage hole, the nodes use a random back-off algorithm before going to sleep.

Jiang et al. [46] identified two short-comings in the sponsored area approach of [45]. Firstly, neighbors lying outside the sensing range are not considered although they can contribute to the node coverage. Secondly, the sector based area calculation for coverage results in a more conservative estimate of the neighbors contributions in covering the area. The authors introduced the concept of effective neighbor nodes for calculating the nodes coverage accurately and to decide upon redundant nodes that could be put to sleep while conserving the original coverage. The neighbor set now includes all nodes within twice the sensing range of each node.

Ghrist et al. [47] proposed the use of algebraic homology for finding the topology of the union of the nodes covering disks. They provided sufficient (not necessary) conditions for establishing the presence of coverage holes in the topology. The authors assumed uniform sensing ranges and that no localization or orientation information is available. They used completion of network communication graphs to derive useful complexes. The proposed mechanism is centralized.

Table 2.1 summarizes the protocols discussed so far. Protocols have been categorized on the basis of degree of coverage. Most of the aforementioned coverage related protocols assume uniform sensing ranges. Probabilistic coverage for sensor networks has been explored in some research efforts but in different context to our work. Kuo et al. [48] proposed an error model targeting a location estimation application assuming probabilistic coverage for sensors. A signal strength based approach is used for modeling a probabilistic function that depends on the distance between the sensor and the object. The authors proposed a single value, overall weighted error degree, as a metric to evaluate the location tracking capability of a sensor network.

Ren et al. [49] provided analytical models based on probabilistic coverage to track a moving object in the sensor field. They used temporal probabilistic coverage, where any point in a sensing field is sensed with a certain probability at any time. Temporal probabilistic coverage allows sensor nodes to periodically sleep

Category	Proposed Solution	Main Assumptions	Characteristics
Multiple Coverage	CCP [39]	Location information, uniform sensing disk	Configurable degree of coverage, calculated by intersection points of sensing circles.
	k-UC, **k**-NC [38]	Location information	Perimeter coverage, non-disk sensing model supported.
	Differentiated [41]	Location information, time synchronization	Grid-based differentiated degree of coverage.
Single Coverage	MC-MC [42]	Time synchronization, uniform sensing disk	Centralized, Constant time slots for sleep/wake cycles.
	MSC [43]	Location information, time synchronization	Centralized, variable time slots for sleep/wake cycles, node sets need not be dis-joint.
	OGDC [44]	Location information, uniform sensing disk	Residual energy consideration.
	Sponsored Area [45]	Location information	Sector based coverage calculations, non-disk sensing model supported.
	Extended-Sponsored Area [46]	Location information, time synchronization	Uniform disk sensing model.
	Homology [47]	Does not require location information	Centralized, uniform disk sensing model.

Table 2.1: Comparison of solutions to area coverage problem assuming static networks

and wake up based on sensing schedules. The authors considered both random and synchronized sensing schedules. They assumed that sensor deployment is dense enough to support duty cycling of nodes to save energy at the cost of providing probabilistic coverage.

In [50], Zou et al. proposed a grid-based clustered approach to evaluate the target localization. The scheme is centralized whereby a cluster head is responsible for calculating the likely position of a target by employing a probabilistic scoring-based localization algorithm. A detection probability table is first generated by the cluster head for each grid point. This probability table is used to query suitable sensors for more detailed information once a target is initially reported to the cluster head.

Xing et al. [51] proposed a probabilistic event detection model based on data fusion. They organized the network into fusion groupings that are localized on overlapping virtual grids. Their protocol Co-Grid coordinates neighboring fusion groups to activate nodes. The performance metrics considered are the number of active nodes and (re)configuration time.

Kumar et al. in [52] studied randomized independent sleeping (RIS) where each sensor is active with a probability ρ independently from other sensors. The authors show that the network life time can be increased by a factor close to $1/\rho$. Through analytical formulations for grid, random uniform and Poisson deployment, the authors provided an estimate of appropriate number of nodes required to achieve k-coverage using duty cycling.

The work on coverage estimation proposed in this thesis is different from these research efforts in several ways. Firstly, this thesis proposes a computational geometry based approach assuming probabilistic coverage characteristics for the deployed sensor nodes. This probabilistic coverage approach is a deviation from the idealistic binary detection model and captures the real world sensing behavior of range-based sensors such as acoustic, seismic, and radio etc. Secondly, the coverage is calculated at perimeter of each node's sensing circles instead of generalized grid points. This gives us a more accurate coverage calculations for each node. Our approach extend the notion of the perimeter coverage proposed by Huang et al. [38]. Thirdly, the proposed approach is truly distributed, and all nodes run the algorithm compared to the cluster head performing the coverage calculations as in [50]. This has the added advantage of being scalable and robust in case of failures e.g., cluster head malfunctioning etc.

2.4 Coverage Enhancement Using Mobile Sensors

The protocols discussed here are classified based on the number of mobile nodes in the sensor network as (i) All Mobile sensor networks (ii) Hybrid sensor networks.

2.4.1 All Mobile Sensor Networks

Several researchers have investigated techniques to obtain maximum coverage of a target area using mobile sensors. A typical problem statement for this scenario is to maximize the coverage of a given target area with constraints on deployment time, the distance the sensors have to travel to maximize coverage and the complexity of the protocol [53].

In one of the proposed solutions, Wang et al. [53] used Voronoi diagrams to discover the existence of coverage holes once all the sensors have been initially deployed in the target area. The diagram partitions the whole space into Voronoi polygons. Each polygon has a single node with the property that every point in the polygon is closer to this node than any other node. A sensor node compares its sensing disk with the area of its Voronoi polygon to estimate any local coverage hole. Figure 2.2 shows a Voronoi diagram for a node x where $abcde$ is the Voronoi polygon and the circle centered at node x is the sensing disk. Three distributed self-deployment algorithms have been proposed: Vector based (VEC), Voronoi based (VOR) and Minimax algorithm. In VEC, a node calculates the average distance between the neighbors assuming all nodes are uniformly deployed in the given target area and tries to keep the distance with its neighbor approximately equal to this distance. In VOR, once the coverage hole is detected, the node moves toward the farthest vertex of the Voronoi polygon to cover the local maximum coverage hole. Minimax works like VOR with the additional check that while moving toward the farthest Voronoi vertex, it also keeps track of distances to other vertices and finds a target position inside the polygon from where the distance to the farthest vertex is minimized. Simulation results presented in [53] show that the Minimax algorithm outperforms the other two proposed algorithms in achieving maximum coverage with little increase in computational overhead.

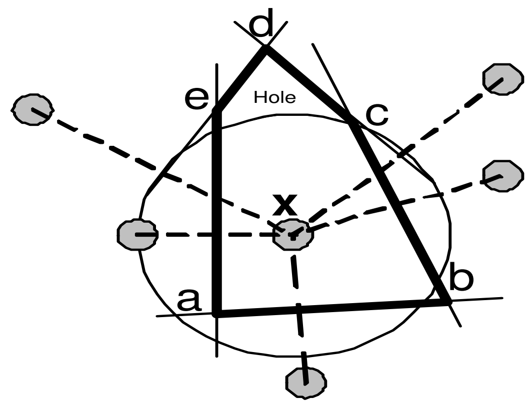

Figure 2.2: Voronoi diagram.

In other significant research, Howard et al. [54] proposed a potential fields based approach for self-deployment of mobile sensor networks. Nodes are treated as virtual particles and the virtual forces due to potential fields repel the nodes from each other and obstacles. The authors assume that each sensor is capable of determining the range and bearing of both its neighbor nodes and the obstacles. This approach does not require any communication among the nodes for movement or localization information. Instead, the nodes only use their sensed information in making the decision to move making it a cost effective solution to the coverage problem. However, extensive simulation/experimental studies have not been conducted to test the sensitivity of the approach to changes in communication and sensing ranges and different network sizes etc.

In other related work by the same authors [55], an incremental self deployment algorithm is proposed that maintains the line of sight relationships among the nodes. The line of sight constraint is necessary for localizing the nodes by using existing deployed nodes as landmarks. The target position of a node is calculated, using the deployment algorithm, at a high processing power base station. Each deployed node is responsible for communicating its local information back to the base station for utilization in the next iteration of the deployment algorithm. This implies that each node has to maintain bidirectional communication with the base station at all times or else they are deployed so that they form a multi-hop connected network at all times. The authors have conducted simulations based on achieved coverage and time. Different deployment goal selection policies such as deterministic and stochastic etc. have been proposed and evaluated. Results have shown that deterministic goal selection policy outperforms stochastic based policies but it still cannot guarantee complete coverage of the sensor network. As compared to the distributed self deployment proposals, the proposed sequential deployment scheme will take much longer time to deploy when the number of nodes is increased.

Authors in [56] proposed a potential field-based deployment approach to maximize the area coverage with the additional constraint that each of the nodes has at least k neighbors, where k is a user specified parameter. They assumed compact initial concentration of the mobile nodes and that each node can calculate the range and bearing of its neighbors and obstacles.

Heo et al. [57] proposed two schemes for addressing single coverage problem. In one scheme called Distributed Self-Spreading algorithm (DSS) the authors propose a deployment scheme similar to [54] and VEC [53]. Sensors are assumed to be randomly deployed initially. They start moving based on partial forces exerted by the neighbors. Forces exerted on each node by its neighbors depends on the local density of deployment and on the distance between the node and the neighbor.

Another scheme called Intelligent Deployment and Clustering Algorithm (IDCA) is also proposed in [57] to utilize low energy consumption characteristics of local clustering. Local density is compared with density expected when all nodes are uniformly distributed in the target area, and for close values a sensor selects the clustering mode. In this mode the relative remaining energy of the sensors dictates whether a node should move or not. The idea is to reduce the variance in remaining energy once all the nodes are uniformly distributed in the target area.

In [50], the authors propose a centralized virtual force-based algorithm. This approach assume all sensors are mobility capable and are able to communicate with a cluster head. The aim is to maximize coverage using a combination of attractive and repulsive virtual forces. The VFA is executed by the cluster head and the final locations are sent to the sensor nodes which perform a one-time movement to relocate to the new locations.

Liu et al. [58] investigate the area coverage with continuously moving mobile nodes when the location covered changes with time. Average area covered at a given time as well as during a time interval has been characterized. Wu et al. in [59] proposed SMART, a scan-based deployment scheme for mobile sensor networks. The scheme use a hybrid approach to load balance the existing irregular deployment by employing grid-based clusters. Two scans are performed in sequence, one for rows followed by another for the columns. Mobile nodes are moved from one grid to another to load balance the number of nodes in each cluster. The scheme performs well for dense networks with a high degree of clustering. Yang et al. [60] improved the performance of the SMART scheme by proposing a distributed local algorithm based on the Hungarian method. The scheme assumed that the global average load per cluster is pre-computed using SMART, and load balancing is performed based on the weighted bipartite graph matching.

To summarize, the solutions discussed in this subsection either utilize techniques from computational geometry e.g., Voronoi diagrams or uses virtual forces, to attract or repel neighbors, or use load balancing to achieve the desired coverage of the area.

2.4.2 Hybrid Sensor Networks

The coverage scenario when only some of the sensors are capable of moving has been under active research, especially in the field of robotics for exploration purposes. The movement capable sensors can help in deployment and network repair by moving to appropriate locations within the topology to achieve desired level of coverage and connectivity, and to connect a possibly disconnected network.

Corke et al. [61] addressed the issue of network deployment with adequate connectivity using an Unmanned Aerial Vehicle (UAV). The flying robot, referred as AVATAR by the authors, can deploy the network based on a pre-computed network topology. The sensors, once deployed, compute their connectivity map and relay this information to the flying UAV. The existing network connectivity is compared with the desired topology and the separation regions are identified. Deployment points within this region of separation are computed to repair the network and the UAV again deploys more sensors at desired deployment points. The deployment of additional sensors in the target area increases the sensor density(achieving multiple coverage and connectivity) or repairs failing sensor nodes in the network. The experimental results during the network deployment stage show that it is very difficult to achieve the desired network topology using aerial deployment.

In [62], Batalin et al. suggested a combined solution for the exploration and coverage of a given target area. The coverage problem is solved with the help of a constantly moving robot in a given target area. The mobile robot first performs the network deployment in the target area as it explores the unknown environment. The deployed nodes then guide the robot, based on their local measurements, to poorly covered areas. The mobile robot, using its local sensing data and the recommended direction acquired from a deployed sensor node, decides its future direction for exploration. If the robot does not receive a direction beacon when it has traveled a predetermined distance in one direction, it deploys another sensor node to improve the local coverage of the area.

The algorithm does not consider the communications between the deployed nodes. All decisions are made by the robot by directly communicating with a neighbor sensor node. Distributed computation and in network processing is suggested in [62] as a solution for the homing problem (when a robot wishes to return to a specific point in the target area). The deployment strategy and especially the network repair policy can also benefit from the multi hop information derived out of a communicating sensor network and thus we feel that this should be explored further.

Wang et al. [63] addressed the area coverage problem by moving the available mobile sensors in a hybrid network to heal coverage holes. The static sensors detect their local coverage holes by using Voronoi diagrams. The mobile sensors also calculate coverage holes formed at their current position if they decide to leave their current position. The static sensors bid for the mobile sensors based on the size of their detected coverage hole. A mobile sensor compares the bids and decides to move if the highest bid received has a coverage hole size greater than the new hole generated in its original location due to its movement. The bids are broadcast locally up to two hops and the static sensors are able to direct neighboring eligible mobile sensors to a point close to the farthest vertex in their Voronoi polygon. However, the local broadcast may prevent the bid messages reaching mobile sensors if they are located farther than two hops.

Category	Proposed Solution	Main Assumptions	Characteristics
Voronoi Diagrams	VEC, VOR, Minmax [53]	Location information	Localized, distributed.
	Bidding protocol [63]	Hybrid network	Mobile nodes move from small coverage holes to large ones in each round.
	Proxy based [64]	Hybrid network	Logical movements in each round, static nodes act as proxies.
Virtual Forces	Centralized VFA [50]	Grid based coverage	Cluster head executes VFA, nodes perform one-time movement
	Potential Fields [54]	Range and bearing	Scalable, distributed. No local communication required for localization or movement.
	Constrained [56]	Range and bearing	Initial compact distribution, maintains k-neighbors.
	DSS,IDCA [57]	Location information	Scalable, distributed, residual energy based.
Load Balancing	SMART [59]	Localization, clustering, dense network	Scan based hybrid approach
	Localized load balancing [60]	Time synchronization, grid based clustering	Distributed Hungarian method.
Miscellaneous	Incremental [55]	Line of sight for localization	Centralized. Bidirectional communication with base station.
	UAV [61]	Predetermined topology information	Flying robot for deployment and network repair. Inaccuracies using aerial deployment
	Single Robot [62]	Location information	Distributed, no multi-hop communication for network deployment and repair.
	Continuous Movement [58]	uniform sensing model, random mobility model	Time varying coverage, mobile targets also considered, no data fusion

Table 2.2: Comparison of mobility based solutions to area coverage problem.

In another work, Wang et al. [64] proposed a proxy based sensor deployment scheme for hybrid networks as an extension of the bidding protocol [63]. Mobile nodes go through logical movements in each round while static sensor nodes are utilized as proxy nodes to advertise and calculate the next logical deployment point based on the bidding protocol. Mobile nodes only physically move when the bidding protocol results in the final deployment position.

In this thesis, we propose a set of localized and distributed variants of the virtual force algorithm for spreading out the mobile sensors as opposed to the centralized virtual force algorithm employed in [50]. We also propose the simulated movement approach for energy efficient movement of mobile sensor nodes. This simulated approach is different from the proxy based logical movement approach proposed in [64] where static sensor nodes are utilized as proxies for message passing.

2.5 Coverage Repair and Maintenance

Wang et al. [65] proposed a grid quorum-based sensor maintenance protocol. The area is partitioned into grids, each with its own grid head. Redundant node information is published by the grid head on the grid row, and requests are sent in its grid column. Closest lying redundant nodes are located so as to follow a cascaded movement approach to fill in the discovered coverage hole. In cascaded movement approach, intermediate nodes lying on the relocation path are involved in the movement to minimize delay and balance the energy consumption.

Sekhar et al. [66] proposed heuristic solutions for network repair in an all mobile network. Coverage is estimated using simplified grid coverage and one-hop neighbors are moved to cover the coverage hole, provided they can move without creating additional holes. A cascaded coverage maintenance scheme is also proposed that works for multi-hop movement scheduling. Performance of the proposed heuristic solutions depend on the deployed topology and do not guarantee complete coverage especially for sparse deployments.

Ganeriwal et al. [67] proposed a protocol called Co-Fi that uses mobility capable sensors for repairing the loss of coverage due to energy depletion. Low energy nodes, on predicting death, broadcast a Panic request message. Nodes with high energy level respond with the Panic reply message if they can move without losing existing coverage. The Panic reply message contains residual energy and the mobility cost (shortest distance the helper node has to travel to reach its final destination). The dying node receives multiple Panic reply messages and it chooses a node with maximum utility (maximum residual energy after the movement) and notifies the selected node to move. As the protocol relies on broadcast of a Panic request message by a dying node, it will not work when nodes get physically destroyed creating irreparable coverage holes.

Mei et al. proposed the use of mobile robots for replacing failed sensor nodes [68]. These robots carry functional sensor nodes and they are capable of dropping these sensor nodes at specific locations to repair coverage. The authors proposed three algorithms, centralized manager, fixed distributed manager, and dynamic distributed manager algorithm, to report failure and locate the optimal robot for network repair.

Li and Santoro proposed an integrated framework that includes deployment and coverage maintenance [69]. The framework assumes an all mobile network and comprises four different algorithms: Algorithm to calculate the number of required redundant mobile nodes (NRC), a virtual force based deployment algorithm (VFSD), a zone based sensor relocation protocol (ZONER), and a sensor replenishment protocol (SRP). VFSD assumes different sensing ranges for nodes and applies virtual forces to keep the inter-node distance equal to the sum of their sensing ranges. ZONER algorithm use a quorum publish-subscribe mechanism to locate the redundant nodes for coverage maintenance. SRP introduce redundant nodes in the topology to compensate for the coverage loss due to sensing range diminishment. The authors assume that sensing range decreases with time due to battery depletion for example, but the communication range is constant.

This thesis addresses the problem of coverage loss due to both energy depletion and physical damage to the nodes. The coverage maintenance problem has been formulated as an Integer Linear Program (ILP) and heuristic solutions for coverage maintenance have been proposed that perform close to the optimal solutions.

2.6 Summary

This chapter reviewed the existing literature and provided background knowledge on areas that are most relevant to the work presented in this dissertation. A secondary objective of this literature survey was to place the work proposed in this dissertation in proper context.

Chapter 3

Distributed Boundary Estimation

3.1 Introduction

Boundary estimation in sensor networks is usually categorized as event boundary estimation and topological boundary estimation. Event boundary estimation takes sensor measurements from a deployed sensor network and determines the boundary between fields of homogeneous sensor measurements e.g., mean values for the measurements [21]. The output is the boundary delineation between homogeneous sensing regions based on the observed sensor data. On the other hand, topological boundary estimation schemes refer to the detection of nodes lying on the topological boundary of a region of interest. In this chapter, we focus on the distributed topological boundary nodes estimation with the aim to make the nodes lying on the outer boundary of a deployed topology aware of their special topology position.

Assume that we have a 2-D region of interest in which we have to deploy a hybrid sensor network consisting of both static and mobile sensor nodes. The static nodes have been deployed uniformly at random in the region. Also assume that all the nodes are aware of their position coordinates through some GPS-less sensor network localization scheme such as [70]. The questions that we wish to answer are: How can we detect the nodes lying on the outer boundary of the deployed topology in a distributed manner? Can we use the nodes lying on the outer boundary to deploy mobile nodes in the unknown region of interest?

Detection of nodes on the outer boundary is essential for self-organization of a sensor network in numerous situations:

- For deployment of mobile sensors in the region, it is imperative to provide a notion of boundary that defines the extent of the area. This helps in keeping the mobile nodes confined in the area.
- From a coverage perspective, boundary estimation defines and limits the region where coverage is calculated and maintained.
- Finally, boundary nodes help in keeping track of objects/events entering or leaving the region boundary.

Most of the existing hybrid sensor deployment schemes either assume regular, known deployment regions without taking into account practical boundary estimation, or assume indoor deployment where the mobile sensors are able to detect the walls as obstructions. This thesis proposes a distributed boundary estimation scheme. In the proposed scheme, static nodes estimate the boundary of the unknown region by using the right-hand rule. Boundary estimation results in identification of the boundary nodes (**B-nodes**), nodes lying on the outer boundary of a deployed topology. Neighboring B-nodes form the virtual boundary of the region.

This chapter details the boundary estimation scheme that forms part of phase I of the MAPC protocol (Figure 3.1). We corroborate the proposed boundary estimation scheme through simulations and experiments conducted with real hardware using Tmote-sky motes [20]. Experiments with real sensor hardware show that observed radio irregularity has a significant effect on correct operation of theoretical protocols. In light of these experimental observations, we have

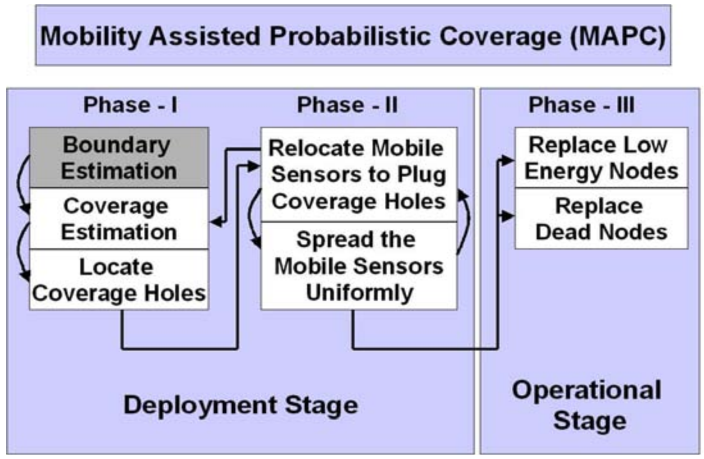

Figure 3.1: Building blocks of MAPC protocol.

adopted the boundary estimation algorithm to ensure that it is resilient to untoward radio propagation effects.

3.2 Boundary Estimation by Static Nodes

A naive approach for boundary detection is to collect the deployed topology information at the sink/base station by flooding probing packets over the whole network. The topology information once gathered can help in identification of boundary nodes by geometric computation for convex hull [22]. Nodes identified to be on the convex hull are then conveyed this information. This approach is centralized, involves high communication and computation cost, and does not scale well with the size of the network. A localized and distributed boundary node selection algorithm is suited for the given scenario.

3.2.1 Boundary Estimation Algorithm

The proposed B-node selection algorithm is based on the simple geometric right-hand rule. It consists of two parts; application of right-hand rule to compute the next B-node, and sending of a special **B-message** to the next selected B-node. After neighbor information has been exchanged through local broadcast of **Hello** messages, a node that receives a B-message marks itself as a B-node. It then sweeps clock-wise from the direction of the sending node and selects the first node that intersects the sweeping line from its neighbor list (by simple geometric calculations as location information of neighbors is available). The node then unicasts a B-message to the next selected node (referred as **forward B-node**) and stores the information about **previous B-node** (node from which the B-message was received) and the forward B-node. The node that initially originated the B-message is referred as **OB-node** (originating B-node) and each B-message contains the OB-node ID. Each B-node also stores the OB-node ID from the received B-message. The B-message processing continues through the boundary nodes of the deployed topology until the B-message reaches back to the OB-node.

Note that our algorithm is different from the Jarvis walk [22] in that we apply the right hand rule for neighbors in communication range only as compared to all points in the plane. A similar scheme is also proposed in [23] wherein the right hand rule is applied on edges of a planar Gabriel graph. It was shown by Kim et al. [35] that real radio often violates the unit disk graph assumption resulting in anomalies in the planarization process. Planar graphs such as the Gabriel graph that are derived from unit disk graphs may result in crossing edges due to radio irregularity. In order to have a more practical approach, we do not assume a planar graph; rather, we introduce explicit checks to detect crossing edges as explained in a later section detailing our experiments. An added advantage of our approach is the lesser processing overhead. The maintenance of planar graph at each node introduces additional overhead. While all the nodes maintain the planar graph, for boundary estimation this information is only utilized by the boundary nodes. Explicit checks for edge crossing at boundary nodes instead of maintaining a planar graph is more efficient, if the number of boundary nodes is relatively small as compared to the total number of nodes in the topology. The B-node selection algorithm is thus simple, distributed and is only executed by nodes lying on the outer boundary of the topology. Some of the related protocol details are discussed below.

3.2.1.1 Selection of the Originating B-node

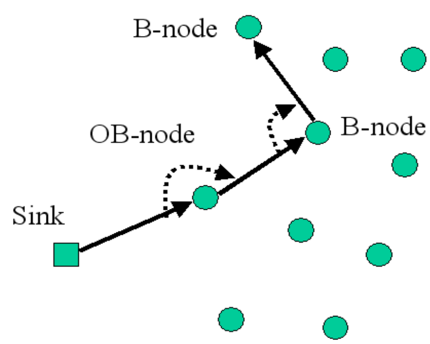

Figure 3.2: Right hand rule.

The data from a deployed WSN is often collected at a base station or sink for analysis. Without loss of generality, we assume that a sink node is available (outside the WSN convex region) and that it can communicate with a set of deployed nodes. After learning the location information of the neighboring static nodes that are within its communication range, the sink selects the nearest neighbor (with the least Euclidean distance) as the OB-node. The sink unicasts a B-message to the selected node to trigger the B-nodes selection algorithm. The OB-node applies the right-hand rule by sweeping clock-wise from the direction of the sink (see Figure 3.2) to select the forward B-node. Note that the sink is only used for starting the B-node selection process and it does not participate any further in the selection process. Also the OB-node does not mark the sink as its previous B-node, the correct previous B-node will connect with the OB-node when the B-node selection terminates.

The B-node selection algorithm described above assumes the presence of a single sink located outside the WSN region. We now show how we can relax this assumption to include the cases of multiple sinks and sinks located inside the WSN region. For multiple sinks, assumed located outside the WSN region, each sink selects its own OB-node and generates B-messages with their respective OB-node ID's. A node that receives a B-message marks itself as a B-node and stores the OB-node ID. A B-node compares the OB-node ID for each additional B-message that it receives. Normal processing of the B-message continues only if the B-message has a different OB-node ID otherwise the B-message is simply discarded. The algorithm thus runs independently for the multiple sink scenarios and terminates when the B-message is received at the correct OB-node. For the case when sink is assumed located inside the WSN region, a scheme based on the quorum based location service [23], [24] can be utilized. The sink initiates a **Locate** message in both east and west directions. The message reaches the east-most and west-most deployed static nodes, which become the OB-nodes. The OB-nodes convert the Locate message into B-message and start the boundary estimation process. The process terminates when B-messages are received at corresponding OB-nodes. Details of the quorum based location service scheme can be found in [23] and [24].

3.2.1.2 Selection of B-nodes

The set of nodes that get selected as B-nodes depends on the communication range and are a subset of all boundary nodes. Figure 3.3 illustrates this and shows part of a deployed sensor network topology. Let us assume that node X has nodes Y, A, B, C, and D in its neighbor list. Node X receives a B-message from node Y and after applying the right-hand rule, selects node D as the forward B-node

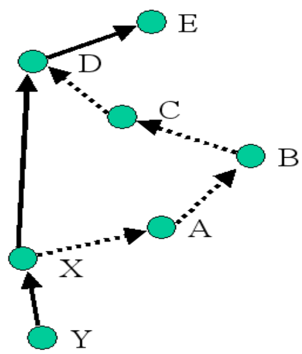

Figure 3.3: Selection of B-nodes.

from its neighbor list. The first few nodes that get selected as B-nodes are $YXDE$ because node D is within the communication range of node X. Note that nodes A, B, and C have not been selected, although the boundary can also be traced as $YXABCDE$ if we assume a shorter communication range.

In the case when multiple neighbors are found at the same minimum sweeping angle from the previous B-node (e.g., prospective nodes are lying in a straight line), nearest such neighbor is selected as the forward B-node.

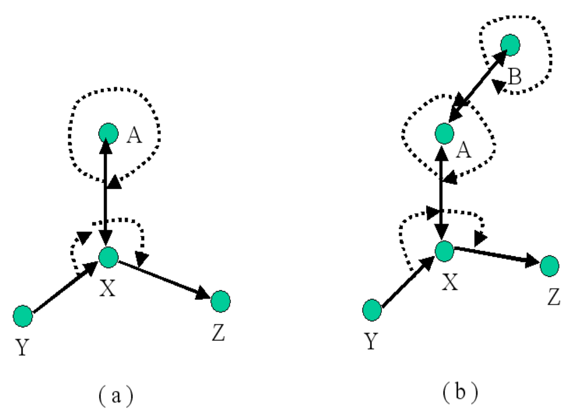

Figure 3.4: Local minima.

3.2.1.3 Local Minima

Consider Figure 3.4. Node X receives the B-message from node Y. X applies the right-hand rule and selects node A as the forward B-node. As node A has only one neighbor, node X, it cannot select any new B-node except node X. We call this special case the local minima for the algorithm and it results from limited radio transmission ranges. Recall that each node maintains the previous and forward B-node information. Node A sends the B-message, with a local minima flag, back to previous B-node X without marking itself as a B-node. Node X removes node A as the forward B-node, continues with the right hand rule and selects node Z as the new forward B-node now. In case a node receiving the B-message with local minima flag has no other neighbor to select than its previous B-node, it again sends a local minima B-message to the previous B-node (see Figure 3.4 (b)). The algorithm can thus recover from a local minima. Note that some of the nodes lying on the convex hull (e.g., Node A in Figure 3.4 (a)) are not selected as B-nodes due to the local minima cases.

3.2.1.4 Algorithm Termination

The algorithm terminates when the B-message (without the local minima flag) is received at the initiating OB-node. Recall that each OB-node is selected to be on the boundary and that each B-message contains the OB-node ID. This results in correct termination of the algorithm even when multiple OB-nodes are selected to initiate the boundary estimation process.

3.2.2 Simulations

The boundary estimation algorithm has been implemented in NS2 discrete event simulator [71]. Simulation setup parameters are listed in Table 3.1. Simulations were run on three different kind of topologies; random, street, and river.

Random and river topologies are non-deterministic while the street topology is deterministic. In the river topology, nodes are assumed randomly deployed along the banks of a meandering river. The transmit range was set at 16m based on Tmote-sky motes calibration tests discussed in Section 3.2.3.

Table 3.1: Simulation setup parameters

Parameter	**Value**
Transmit power P_t	0 dBm
Receiving threshold at sensor	-90 dBm
Path loss exponent n (free space)	2
Communication range	16m
Number of nodes	50

Figure 3.5 shows the simulation results by running the boundary estimation algorithm. Filled circles represent the selected B-nodes and the dashed line between the B-nodes indicate the virtual boundary for different kinds of topologies. OB-node has been marked with solid perimeter. The simulation results show that the boundary estimation algorithm successfully identifies the virtual boundary of the deployed topologies. Note that in street topology, the virtual boundary has formed cut off paths instead of following the deployed pattern. This is because of the fixed transmission range and assumption that the deployment is obstruction-free.

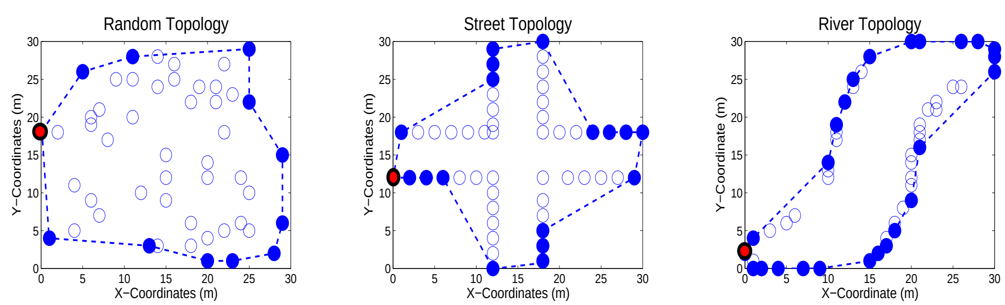

Figure 3.5: Simulations: Boundary estimation for random, street and river topologies.

3.2.3 Experiments

Several researchers have conducted experiments to evaluate and characterize the radio performance for various sensor hardware. Ganesan et al. [72] used Rene motes (with TR1000 radios [73]) while Woo et al. [74] and Zhao et al. [75] investigated Mica motes (with TR1000 radios). Zhou et al. [76] used Mica2 motes (with ChipCon CC1000 radios [77]) and Srinivasan et al. [78] and Zhao et al. [79] conducted experiments with MicaZ motes (with ChipCon CC2420 radios). More recently, Xueli et al. [80] and Holland et al. [81] employed the Telos sky motes [20] (with CC2420 radios) for their experiments. Most of these works show the existence of radio irregularity e.g., asymmetric links, grey areas where the packet reception rate has no correlation with distance, and long unexpected links etc.

Based on the aforementioned experimental studies, to validate our simulations results, we decided to run experiments with Tmote-sky wireless sensor network platform. The Tmote-sky motes use Chipcon CC2420 radio that is IEEE 802.15.4 compliant and are based on Telos revision B platform [82]. The data sheet of the sky-motes specify that the radio range is 125m for outdoor environments and that the internal antenna does not have a perfect omni-directional pattern. As perfect sender-receiver antenna orientation is impossible to achieve, especially in a random non-deterministic deployment of sensor nodes, we conducted preliminary calibration experiments to ascertain the effect of radio antenna orientation and the height above ground on radio range of Tmote-sky motes. The aim is to find a realistic value of the Tmote's transmission range in outdoor environment. This observed radio transmission range can then be utilized to decide the density of the deployment i.e., how many nodes to deploy in an area of given size in order to achieve a connected topology.

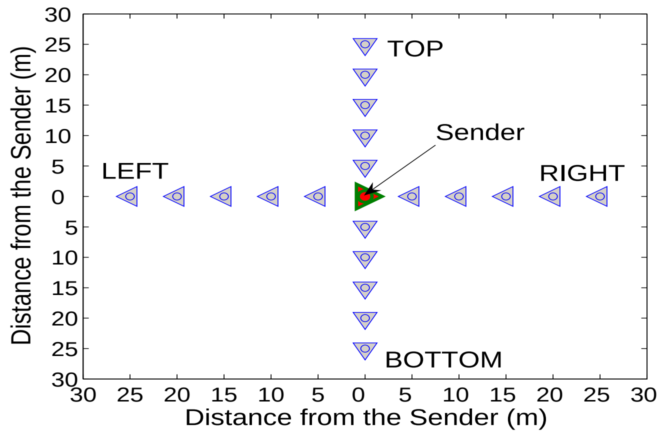

Figure 3.6: Calibration experiment setup.

3.2.3.1 Calibration Experiment Setup

Our experimental setup is shown in Figure 3.6. There is a single sender with 20 receivers arranged in orthogonal directions. All motes that are placed in a horizontal line have their internal antenna pointing towards left while motes placed in the vertical line have their antennas pointing towards bottom. The sender's internal antenna is pointing towards right. The motes are placed at 5m distance increments in four directions (arranged as to give 0, 90, 180 and 270 degree antenna orientation with respect to the sender antenna). The sender transmits at 0 dBm transmit power with a packet rate of ten packets per second for twenty five minutes. The motes were horizontally mounted at 0.1m and 1m above ground in separate set of experiments. Each receiver logs the packet sequence number, RSSI (Received Signal Strength Indicator), and LQI (Link Quality Indication) values for each packet that it receives. LQI is a metric introduced in IEEE 802.15.4 that measures the error in the incoming modulation of packets that passes the CRC check. All these experiments were conducted in an open unobstructed playing field with a new set of batteries.

3.2.3.2 Calibration Experiment Results

Figures 3.7 - 3.11 show our calibration experiment results.

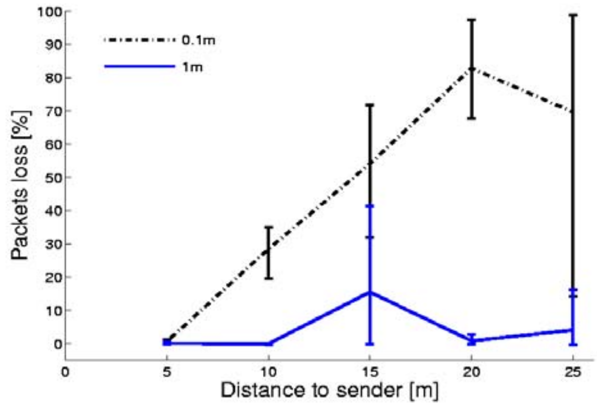

Figure 3.7: Packet loss vs distance.

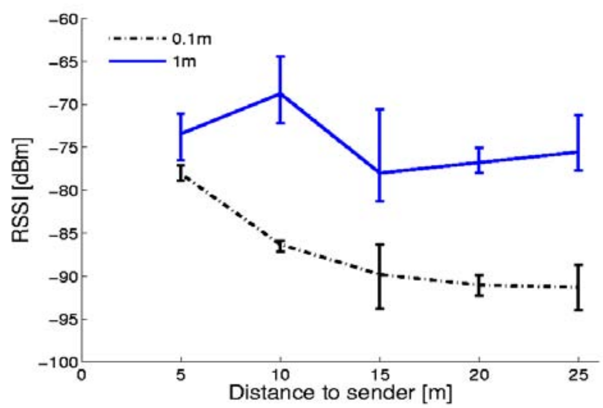

Figure 3.8: RSSI vs distance.

- Effect of Height

  Figures 3.7 and 3.8 show the percentage packet loss and RSSI values vs distance plots respectively while Figure 3.9 shows the LQI values vs distance for both 0.1m and 1m height above ground experiments. The plotted values are the average of the values measured in four directions. The error

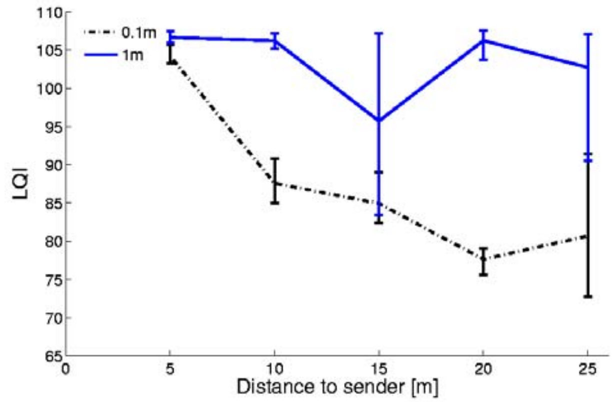

Figure 3.9: LQI vs distance.

bars show the maximum and minimum values received in any direction while the range of the error bars shows the variation due to receiver orientation in different directions. The results show that much better RSSI and LQI values are received when both the sender and the receiver are placed 1m above the ground compared to 0.1m. The reason for lower RSSI at 0.1m mounting is that more transmitted energy is absorbed by the ground at 0.1m than at 1m mounting. The set of experiments with 0.1m height above ground show RSSI and LQI values decrease with distance while for 1m, these values show irregularity with distance as higher values are received at 20-25m than 15m receiver distance. Percentage of packet loss vs distance is shown in Figure 3.7. For 0.1m mounting, average of about 52% of the transmitted packets are lost in all directions at 15m distance from the sender while for 1m mounting case low packet loss rate (less than 10%) is observed for all receivers. The results for 0.1m mounting show that the variation in packet loss in different directions increases with increase of distance from the sender. Comparing the LQI and RSSI values, low variation in RSSI values for correctly received packets is observed in different directions as compared to the LQI values. These results are consistent with those reported in [78] and [81] where the LQI values show a higher range of errors. It is clear from these calibration results that height above ground has a significant effect on the performance of the Tmote radio, affecting the transmission range and the link quality.

- Effect of Orientation

Figures 3.10 and 3.11 show the packet loss and RSSI values received in different directions at a distance of 15m from the sender. This experiment was conducted with two co-located motes placed 0.1m above ground in four orthogonal directions. The results show that a maximum difference

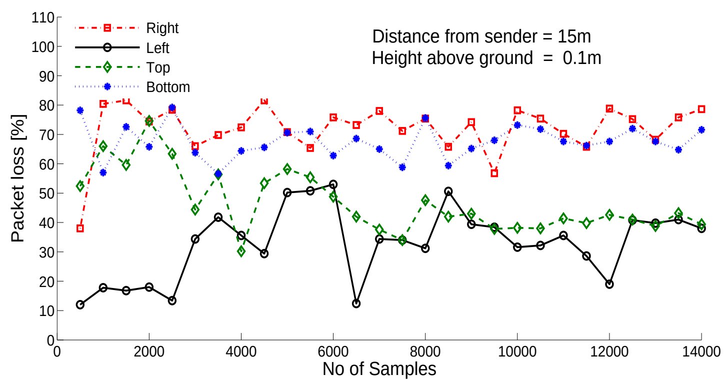

Figure 3.10: Temporal packet loss in different directions.

of about 8 dBm in RSSI values is observed in different orientations (Figure 3.11). As the data sheet for Tmote sky motes indicates the accuracy of the ChipCon radio RSSI values as $+-6$dBm, we can conclude that the observed variation in RSSI values in different directions is mainly due to the inherent hardware characteristics. The packet loss reported in Figure 3.10 has been calculated for each 500 samples interval. Results indicate that the packet loss rate not only varies spatially, different in different directions, but also has a temporally varying behavior. Low

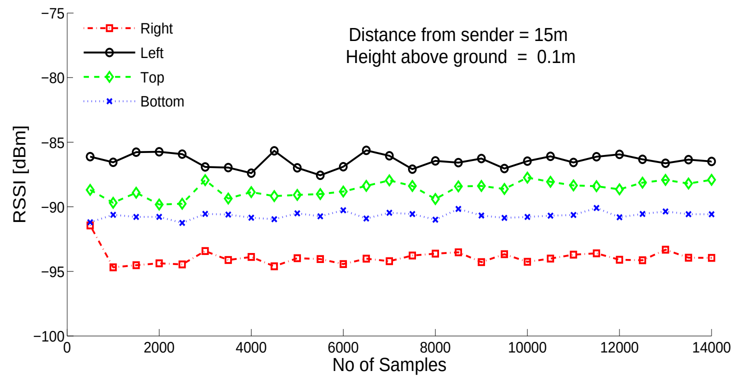

Figure 3.11: Temporal RSSI values in different directions.

packet loss rate with more fluctuations is observed when the sender and receiver antennas are aligned with each other.

As 0.1m mounting above ground is a more realistic option for many sensor network applications assuming a random non-deterministic deployment, we analyzed the calibration results with 0.1m height above ground for selecting suitable values of maximum and reliable transmission ranges. We selected 6m as the value of reliable transmission range for Tmote sky devices as this distance value gives an average packet loss rate of less than 10% (Figure 3.7). Similarly, 16m is selected as the maximum transmission range giving us an average RSSI value of about -90 dBm (Figure 3.8), a value close to the lowest receiver sensitivity of Tmotes sky motes, and an average packet loss rate of around 55%.

Once the values of maximum and reliable transmission ranges are available, we use the Neighbors in Nominal Range Area (NINRA) [83] to find out the numbers of nodes that we need to deploy in a given area.

$$NINRA = \frac{N}{A} \cdot \pi \cdot (T_r)^2 \tag{3.1}$$

where,

N = Number of nodes.

A = Deployment area.

T_r = Transmission range.

For our experiments, we chose to deploy 50 nodes in a 30m x 30m area giving us a NINRA value of about 6. The value of T_r has been set equal to the reliable transmission range (6m).

3.2.3.3 Algorithm Evolution Based on Experiments

As the calibration experiments demonstrated spatial as well as temporal variation in packet loss in different directions, we ran some initial experiments to study the effect of these radio irregularities on the correct working of boundary estimation algorithm. For these experiments, we carefully laid out the exact three topologies used in the simulations i.e., random, street, and river, in a flat playing field. All communications between deployed motes was logged in a mote directly attached to a laptop for debugging and analysis. We did not use any localization scheme, rather we hard-coded the configured location information in each Tmote node. The initial run of experiments with the implementation of basic boundary estimation algorithm indicated that most of the time the algorithm failed to correctly identify the B-nodes. Debugging the results identified the following anomalies:

- Nodes appear in the neighbor list due to the reception of Hello messages but some of these nodes cannot be sent control messages (e.g., a B-Message) due to link asymmetry.
- Links change from symmetric to asymmetric temporally resulting in failure to deliver critical protocol messages. Consider the scenario observed

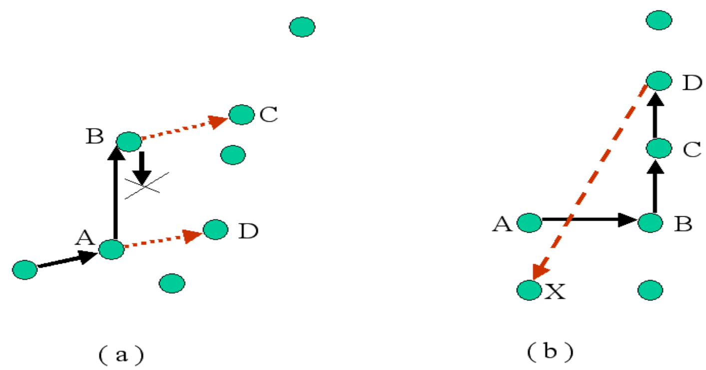

Figure 3.12: Anomalies due to (a) Asymmetric links (b) Unexpected long links.

in one of the test runs shown in Figure 3.12 (a). Node A has send a B-message to node B after applying the boundary selection algorithm. Node B receives the B-message and sends an Ack back to node A. This Ack is not received at node A. Node A times out, removes node B from the neighbor list assuming it cannot reach node B, and selects node D as a new forward B-node. Node B meanwhile proceeds with the boundary estimation algorithm assuming it has sent Ack back to node A. Two nodes are thus concurrently executing the algorithm selecting different forward B-nodes resulting in a non-deterministic behavior of the algorithm.

- Some nodes can have bi-directional communication at longer distances while other nodes cannot communicate with nodes at much shorter distances. As an example see Figure 3.12 (b). Nodes A, B and C are selected as B-nodes. Node C selects node D as the forward B-node (C cannot communicate with node X). Node D on the other hand has node X in its neighbor table and thus selects X as the forward B-node after sweeping from node C. This results in incorrect termination of the algorithm.

Based on the calibration experiment results, and initial trial of the implementation of the basic boundary estimation algorithm, we modified the boundary estimation algorithm to make it robust to the observed radio behavior and suitable for field deployment.

- Constraint on node selection.

  The edge crossing scenario shown in Figure 3.12 (b) resulted because we are not constructing a unit disk-based planar graph for this work. In the absence of such a planar graph, we need to explicitly check for crossing edges. To avoid such a situation, we introduced a constraint that the forward B-node must be further away from the previous B-node than the node making the decision i.e., distance between current node and forward B-node should be less than that between previous B-node and forward B-node. With this constraint in place, in Figure 3.12, node D cannot select X as the forward B-node.

- Neighbor discovery.

  Techniques for neighbor discovery works only if the links are symmetric [76]. To cater for asymmetric links that are likely to result due to asymmetric radio interference, for robust neighbor discovery, we employed **m out of k** strategy. The nodes broadcast k Hello messages and the neighbors only add a node to the neighbor list if m Hello messages from a neighbor has been received. Note that m and k are tunable parameters that reflect the extent of radio irregularity. Larger the ratio of m/k, more stable is the selected neighbor link. For our deployment scenario, based on the calibration tests, we selected 6 and 10 as values for m and k.

  Note that this active link quality estimation is being done at deployment time when little or no data traffic is present. Other link quality estimation techniques such as the windowed mean exponentially weighted moving average (WMEWMA), used in MintRoute multi-hop routing protocol presented by Woo et al. [74], can be utilized during normal operation of the WSN for a more granular link quality estimation.

- Reliable communication.

  To cater for temporal radio interference, we introduced a special **Probe** message in the protocol to check the link symmetry before transmitting B-messages. A node that wants to send a B-message to a forward B-node, first sends the Probe message. The B-message is sent to the forward B-node only if it receives an Ack for the Probe message from the forward B-node. In case of no Ack, the Probe message is re-sent two more times before the neighbor is removed from the neighbor list and the B-node selects another potential B-node neighbor for probing. A node is required to send an Ack back for each B-message that it receives. A B-message is also re-sent a total of three times in case no Ack is received, before the neighbor is removed from the neighbor list.

3.2.3.4 Boundary Estimation Experiment Results

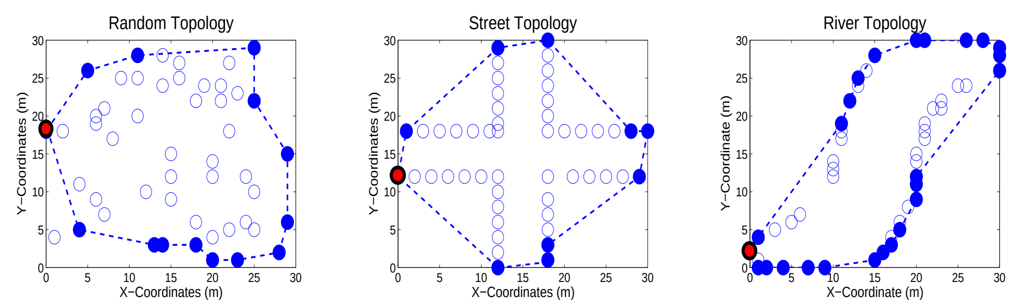

Figure 3.13: Experiments: Boundary estimation for random, street and river topologies.

The pseudo-code for the modified boundary estimation algorithm appears as Algorithm 1. Fifty of the Tmotes were programmed with this modified bound-

Algorithm 1 Distributed Boundary Estimation Algorithm

Notations :
S_{BF} = Forward B-node, S_{BP} = Previous B-node
LM Flag = Local Minima Flag
Let $S = \{S_1, S_2, \ldots, S_n\}$ represent the set of neighbors in communication range and where each sensor S_i is located at coordinate (x_i, y_i).
θ_i is the angle measured clockwise from S_{BP} to S_i.
θ_{BF} is the angle measured clockwise from S_{BP} to S_{BF}.

INIT :
Nodes broadcast k Hello messages (neighbor announcements)
if m Hello messages received from a node **then**
  node is added in S
sink sends B-message to OB-node

Process :
1: **if** B-message received **then**
2:   send Ack to sender
3:   **if** node is already a B-node/OB-node and LM Flag set **then**
4:     unmark sender as S_{BF}; remove sender node from S
5:   **else**
6:     **if** node is already a B-node and LM Flag unset **then**
7:       send B-message with local minima flag to sender
8:       terminate the algorithm
9:   **else**
10:     **if** node is OB-node and LM Flag unset **then**
11:       terminate the algorithm
12:   **else**
13:     mark sender as S_{BP}
14:   **for** each neighbor i in S **do**
15:     **if** distance to S_i < distance between S_{BP} and S_i **then**
16:       calculate θ_i
17:   $\theta_{BF} = min\{\theta_i\}$
18:   for equal min angles, select nearest S_i as S_{BF}
19:   mark yourself as B-node
20:   **if** $S_{BF} == S_{BP}$ **then**
21:     mark LM flag in outgoing B-message
22:   send Probe message to S_{BF}
23:   **if** Ack of Probe received from S_{BF} **then**
24:     send B-message to S_{BF}
25:   **else**
26:     remove S_{BF} from S after three retries; goto 15
27:   **if** Ack of B-message received from S_{BF} **then**
28:     terminate algorithm
29:   **else**
30:     remove S_{BF} from S after three retries; goto 15

ary estimation algorithm. Figure 3.13 shows our experimental results. Comparing these results with those from the simulations in Figure 3.5, we can observe some variations. Some of the boundary nodes that get selected as B-nodes are different in simulations than in experimentation results. This is because of Tmotes showing different communication ranges in different direction, a behavior that is not modelled in simulations. Also note that in the random topology experiment result, the mote in the lower left corner does not get selected as B-node. The neighbor announcements from this node fails the m out of k (6 out of 10) check and hence the node does not appear as a candidate B-node to its neighbor B-nodes. Nodes with reliable symmetric links are only included in the selected B-nodes list.

The modified boundary estimation algorithm can thus reliably identify B-nodes with stable symmetric links in the presence of realistic radio irregularities. These identified B-nodes form the virtual boundary of the region.

3.3 Boundary Landmarks

We have presented a simple and distributed boundary estimation scheme wherein the nodes lying on the outer boundary of a deployed WSN are made aware of their special topology positions. In this section, we show that it is simple to extend the algorithm to not only determine the B-nodes, but also construct an ordered list of boundary landmarks referred to as **Posts**. The idea is to select some of the B-nodes to act as Posts, collect the information of all such Posts, and then replicate this information to each of the Posts. If B represents the set of B-nodes identified to be on the outer boundary of the deployed topology, and P represents the set of B-nodes selected as Posts, then $P \leq B$. In the extreme case, all B-nodes can act as Posts when $P = B$.

Ivan et al. [23] proposed the idea of location servers, where each boundary node stores location information of all the nodes in the topology. The idea of Posts is similar with the difference that, in our scheme, only some of the boundary nodes (i.e., Posts) store the information of only a subset of all boundary nodes (i.e., set of Posts). The information stored at Posts gives a coarser view of the topological boundary. This information is useful in several ways e.g., a sink can easily retrieve a coarse view of the topology boundary, from any of the Posts, to estimate the extent of the region of interest. Also, in a hybrid network, mobile nodes can get a coarser boundary information from Posts and tight, finer boundary information from their neighboring B-nodes. This segregation of information, in coarser overall boundary and finer local boundary, helps in planning energy efficient movements for mobile nodes.

3.3.1 Locating the Posts

The information exchange required for selecting Posts, and collecting the information of all the Posts, can be piggybacked on the B-messages. The OB-node marks itself as a Post, initializes a counter called PostCount in the B-message to a predetermined value, and puts its node ID and location information as the first value in the list of Posts. It then sends the B-message to the next selected B-node by following the B-node selection algorithm described in earlier sections of this chapter.

The initial value of the PostCount counter in the B-message controls the number of B-nodes that will act as Posts. Each B-node is required to decrement this PostCount counter value by 1. A B-node marks itself as a Post if the Value of PostCount turns to zero after the decrement, and resets the counter to its initial predetermined value. Setting this counter to 1, results in all B-nodes selected as Posts.

A node that receives a B-message, checks whether the value of the PostCount is zero after decrement by one. If so, it takes the following actions:

- Marks itself as a Post;
- Appends its node ID and location information to the list of Posts contained in the received B-message;
- Reset the PostCount to its initial value (also contained in the B-message);
- Forward the B-message to the next B-node after applying the boundary node selection algorithm.

A B-node simply forwards the B-Message to the next B-node if the decremented value of PostCount is greater than zero. The B-message traverses the B-nodes, each selected Post appends its information to the B-message until this B-message reaches back the OB-node. The OB-node now has the complete list of all B-nodes that has been selected to act as Posts.

3.3.2 Replicating the Information of Selected Posts

Once the ordered list of all selected Posts are available at the OB-node, we have to replicate this information to all the Posts. The OB-node copies the list of all Posts into a **PostList** message and sends this PostList message to the forward B-node. Each B-node that receives a PostList message checks if it is a Post and that its node ID is contained in the list of Posts. In case of a match, it copies the content of the PostList message and forwards the message to the forward B-node. The PostList message thus traverses all the B-nodes and replicates the list of Posts to all selected Posts before its reaches back to the OB-node.

3.3.3 Discussion

Figure 3.14 shows the virtual boundary, for a 100 nodes topology, formed by B-nodes and Posts selected by a PostCount value of 2 i.e., every alternate B-node

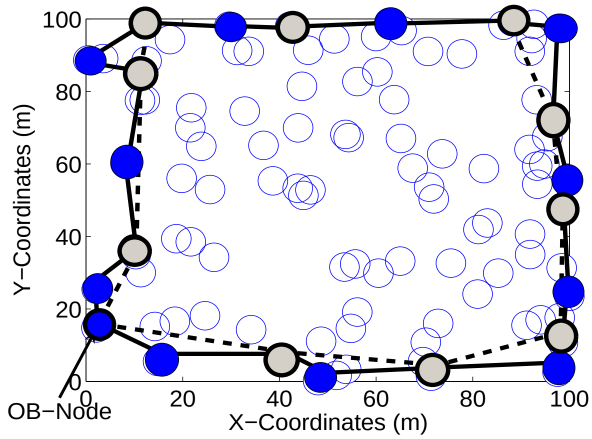

Figure 3.14: Virtual boundary with Posts.

has been selected as Post. The B-nodes are shown by dark filled circles, Posts by light filled circles with dark boundary and unfilled circles represent interior nodes. Boundary formed by Posts is shown by dashed lines.

The process of establishing the boundary landmarks, Posts, involves the overhead of state maintenance at Posts and transmission of additional PostList messages. The communication overhead is incurred only once during the deployment phase while the state maintenance overhead can be reduced by increasing the PostCount counter such that a lesser number of B-nodes are selected as Posts that in turn stores less information. On the other hand, another variant in the selection criteria is possible whereby Posts (still a subset of all B-nodes) store information of all the B-nodes. In this case, each Post has the complete information of the boundary of the region i.e., ordered list of all the B-nodes, at the cost of additional state maintenance. In Chapter 5, we will show how the coarse boundary information stored at Posts can be utilized to effectively plan the movements of mobile nodes in a hybrid sensor network.

3.4 Summary

This chapter describes the boundary estimation scheme, based on the simple right hand rule, to locate the boundary nodes in a deployed topology. The scheme is distributed and is executed only by a subset of all deployed nodes. We also present an extension to the basic boundary estimation scheme that enables the availability of coarse boundary information at several landmarks, called Posts, on the outer boundary of the deployed network.

The working of the boundary estimation scheme has been studied both through discrete event simulations and conducting experiments on real motes. Our experimental results show that undesirable radio characteristics have a significant impact on the correct operation of theoretical protocols. Based on the experimental results, we modified the boundary estimation algorithm to address these practical challenges of real deployment. This confirms the growing concern among the WSN research community that simulations alone cannot capture the true behavior of protocols in real deployment unless validated by field experiments.

This boundary estimation scheme forms Phase I of our comprehensive protocol for coverage calculation and maintenance using a combination of mobile and static sensor nodes. Other phases of the protocol are discussed in subsequent chapters.

Chapter 4

Probabilistic Coverage in Sensor Networks

4.1 Introduction

Wireless sensor networks differ from ad-hoc networks in several ways. One of the distinguishing features is the introduction of the sensing component in sensor networks. A node in a sensor network is thus performing two demanding tasks simultaneously, sensing the environment and communicating with each other to transfer useful information. Sensing is a task of paramount importance for proper functioning of wireless sensor network. The sensing coverage of a sensor node is usually assumed uniform in all directions (represented by unit disk), following the binary detection model. An event that occurs within the sensing radius of a node is always assumed detected with probability of one while any event outside this circle of influence is assumed not detected. This idealized model has been extensively used in recent research work to predict the total coverage in the target area. However, this model is based on the unrealistic assumption of perfect coverage in a circular disk for all the sensors. The sensing capabilities of networked sensors are affected by environmental factors in real deployment and it is imperative to have practical considerations at the design stage in order to anticipate this sensing behavior.

In this chapter, we explore the problem of determining the area coverage, provided by non-deterministic deployment of sensors, using a more realistic coverage model. To capture the real world sensing characteristics of sensor nodes, we assume that the signal propagation from a target to a sensor node follows a probabilistic model. In particular, for range-based sensor modalities such as acoustics, radio, etc, the signal strength of the triggering signal decays as a function of distance. This implies that the detection capabilities of these sensors would exhibit similar distance dependant characteristics as opposed to a uniform sensing range. In fact, the unit disk sensing model over-estimates the coverage achieved by the sensor network and would possibly lead to false negatives i.e., in situations where certain events are not detected, the unit disk model indicates that the location of these events are covered.

The probabilistic coverage model is only valid for certain kind of sensors e.g. radio, acoustic, seismic etc. where the signal strength decays with the distance from the source and does not hold true for sensors that only measure local point values e.g., temperature, humidity, light etc. Our work therefore targets applications like object tracking and intrusion detection that require a certain degree of confidence in the detection probability. This work is based on the path loss log normal shadowing model [84] although it can be extended to incorporate different signal decay models e.g. acoustic signal model (where signal roughly decays at inverse square of distance) for acoustic sensors. We propose the **Probabilistic Coverage Algorithm (PCA)**, a novel coverage estimation algorithm. The PCA extends the concept of perimeter coverage proposed by Huang et al. [38], to evaluate the effective coverage that can be provided to the application utilizing the sensor network. Simulation results show that coverage calculated using probabilistic coverage algorithm is more accurate than the idealistic binary detection model.

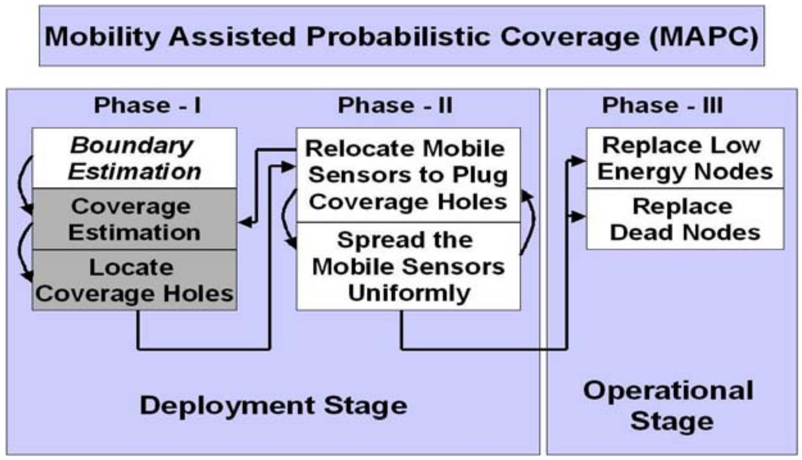

Figure 4.1: Building blocks of MAPC protocol.

Figure 4.1 illustrates that the coverage estimation is performed by the static sensors after the boundary nodes have been identified. This coverage computation together with boundary estimation forms phase I of the MAPC protocol. The remainder of this chapter is organized as follows. We introduce the problem area by discussing some technical preliminaries in Section 4.2. Section 4.3 elaborates the probabilistic coverage algorithm. Simulation results are presented in Section 4.4. Experimental work is discussed in Section 4.5 and Section 4.7 summarizes the chapter.

4.2 Technical Preliminaries

The probability of detection of a target by a sensor decreases exponentially with increase in distance between the target and the sensor. Using the log-normal shadowing model, the path loss PL(in dB) at a distance d is given by Equation 4.1.

$$\overline{PL(d)} = \overline{PL(d_0)} + 10 \cdot n \cdot log(\frac{d}{d_0}) + X_\sigma \tag{4.1}$$

where,

d_0 = Reference distance

n = Path loss component, indicating the rate at which the path loss increases with distance

X_σ = Zero-mean Gaussian distributed random variable (in dB) with σ-variance (shadowing, also in dB)

$\overline{PL(d_0)}$ = Mean path loss at reference distance d_0.

X_σ in Equation 4.1 captures various environmental factors resulting in different received signal values at different locations although the distance between the target and sensor is the same. Values of n and X_σ can be measured experimentally [85]. Similarly $PL(d_0)$ can be measured experimentally for given event and sensor characteristics or can be calculated using free space path loss model [84].

Each sensor has a receive threshold value γ that describes the minimum signal strength that can be correctly decoded at the sensor. The probability P_{ri} that the received signal level, P_{rec}, at a sensor S_i will be above this receive threshold, γ, is given by Equation 4.4, requiring Q-function to compute probability involving the Gaussian process. The Q-function is defined as

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty exp(-\frac{x^2}{2})dx \tag{4.2}$$

where

$$Q(z) = 1 - Q(-z) \tag{4.3}$$

$$P_{ri}[P_{rec}(d) > \gamma] = Q[\frac{\gamma - P_{rec}(d)}{\sigma}] \tag{4.4}$$

For a given transmit power and receive threshold value, we can calculate the probability of receiving a signal above the receive threshold value, γ, at a given distance using Equations 4.2 and 4.4.

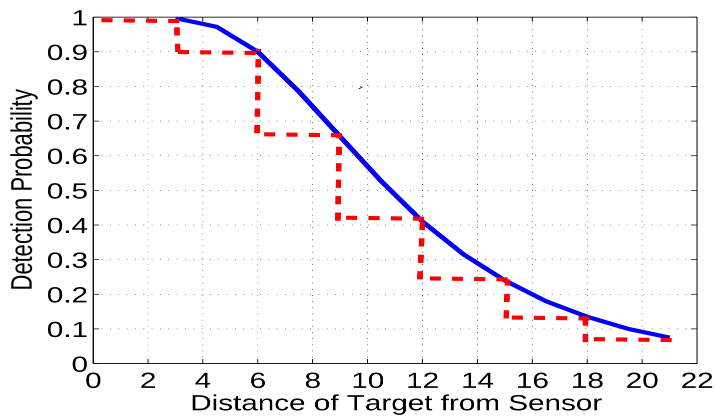

Figure 4.2: Change in detection probability with distance(m).

Figure 4.2 shows the decrease in detection probability for a sensor based on the shadowing model for parameters shown in Table 4.2. The continuous change in detection probabilities with distance can be approximated by discrete values at constant distance increments around the sensor location (shown by dashed lines in Figure 4.2). Assuming this rate of change in the detection probability to be constant in every direction, we can draw concentric circles around a sensor location to represent its discrete detection probabilities. Each circle thus represents the probability of correctly receiving a signal with strength above receiving threshold at distance equal to radius of the circle.

For a deployed sensor network, a point in the target region can be covered by more than a single sensor. The cumulative detection probability at a point in the region, Pr, is given by Equation 4.5.

$$Pr = 1 - \prod_{i=1}^{N}(1 - Pr_i) \tag{4.5}$$

where,

N = Number of sensor nodes covering a particular point

Pr_i = Detection probability of a point for a sensor i

4.3 Probabilistic Coverage Algorithm

The coverage not only depends on the sensing capabilities of the sensor but also on the event characteristics [86] e.g., target detection of military tanks as compared to detection of movement of soldiers depends on the nature and characteristics of the event as well as the sensitivity of the sensors involved. We therefore assume for this work that the signal transmit power, P_t (characteristic of the event) and receive threshold for sensor, γ, is known through experiments and sensor calibration. Once the transmit power and the receive threshold of sensors are known, a **Probability Table**, PT (see Table 4.1) can be precomputed (using Equations 4.1 - 4.4) that provides the discrete detection probabilities at various distances from the sensor.

Table 4.1: A sample probability table (PT)

Distance (m) from Sensor	Detection Probability
3	0.997
6	0.90
9	0.665
12	0.41
15	0.245
18	0.135
21	0.075

Definition 1 : *Effective coverage range, R_{effec}, of a sensor S_i is defined as distance of the target from the sensor beyond which the detection probability is negligible.*

For this work, R_{effec} is taken as the distance at which the probability of detection falls below 0.1. Note that adding an additional sensor with a detection probability of 0.1 in Equation 4.5 results in an overall increase of less than 10%. This increase is minimal (between 5% and 1%) if the existing detection probability is between 50-90%. We, therefore, neglect the part of the coverage that does not have a significant effect on the overall coverage calculations. Following definition 1, two sensors Si and Sj are considered effective neighbors in a region A, contributing to coverage of each other, only if the Euclidean distance between them, d_{ij}, is less than twice the effective coverage range R_{effec}.

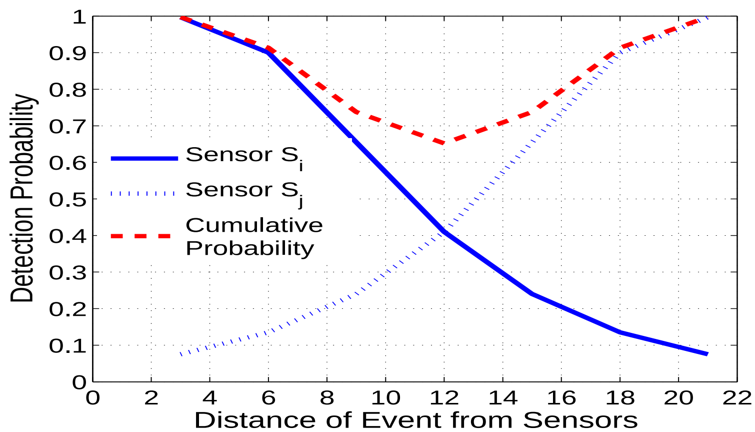

Figure 4.3: Detection probabilities.

Figure 4.3 shows the individual as well as the cumulative detection probabilities, for two effective neighbors in a region, for parameters listed in Table 4.2. The distance between the two sensors is 24m. If an event occurs at the midpoint between the two sensors S_i and S_j, the cumulative detection probability using Equation 4.5 is 0.65 while the individual detection probabilities for both S_i and S_j is 0.41. If the event is moved toward either of the sensors, the cumulative detection probability is higher than this minimal value at the midpoint.

If ρ_{reqd} represents the desired detection probability (DDP) for a region, a simple approach to calculate the coverage is to apply the perimeter coverage algorithm proposed by Huang et al. [38], assuming the sensing range of all the sensor nodes equal to a distance, d_{reqd}, from the sensor that provides ρ_{reqd} (distance and associated probability values taken from the PT). This is equivalent to using the binary detection model with sensing range set to d_{reqd}, thus restricting neighbor relationship to sensors located within twice the d_{reqd}. Figure 4.4 shows the rate of decrease of the detection probabilities with probabilistic and unit disk models. The unit disk model with sensing range equal to d_{reqd} underestimates the detection probabilities provided by a sensor (dashed region in Figure 4.4 is not considered). Clearly, the unit disk model overestimates the coverage if we use the R_{effec} as the sensing range.

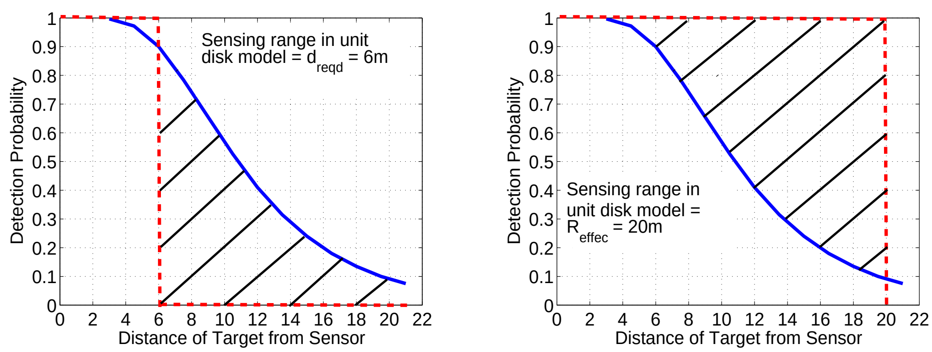

Figure 4.4: Comparison of probabilistic and unit disk models.

Referring to Figure 4.3, we observe that the detection probability at any location is increased by contributions from all the sensors covering that point and this cumulative effect is more profound if the sensors are located near each other. It is thus possible to achieve the desired detection probability at distances greater than the d_{reqd} by considering the contribution of neighbor nodes within the effective sensing range. The basic idea is to take the next higher distance from the probability table PT as d_{eval} (with lower detection probability than ρ_{reqd} as $d_{eval} > d_{reqd}$) and evaluate whether contributions from neighbors makes the perimeter at d_{eval} sufficiently covered or not.

Definition 2 : *A location in region A is said to be sufficiently covered if its cumulative detection probability, due to sensors located within the effective coverage range R_{effec} of this location, is equal to or greater than DDP, the detection probability desired by the application.*

The application utilizing the sensor network thus specifies the desired threshold for coverage probability, ρ_{reqd}, and our objective is to check whether all locations in the given region are sufficiently covered or not.

4.3.1 The Algorithm

We adopt a computational geometry based approach and propose Probabilistic Coverage Algorithm (PCA) to check whether the current deployment supports the required coverage probability or not. We make the following assumptions for this work:

- Sensors are randomly deployed in the target area.
- Location information is available to each sensor node by using some GPS-less sensor network localization scheme [87].
- Communication range of sensors is at least twice the effective coverage range, R_{effec} to ensure that effective neighbors are able to communicate with each other.
- Transmit power of target (P_t) and receive threshold (γ) for a sensor are known and γ is the same for all the sensors.
- Mean values of path loss component n and shadowing deviation σ are assumed for all the sensors.

In the initialization phase of the algorithm, a sensor Si receives **Hello** messages that contain the location information from all of its one hop communication neighbors, including the static inner nodes, B-nodes, and the Posts. It calculates the distances to all such neighbors and keeps them in a list sorted on distances. Si has two sensing circles with radius d_{reqd} and d_{eval}. d_{reqd} is the distance from the sensor providing ρ_{reqd} while d_{eval} is the next distance increment that is greater than d_{reqd} providing a lower detection probability than ρ_{reqd}. Both d_{reqd} and d_{eval} are taken from the probability table PT.

Sensor S_i now calculates the virtual boundary formed by the neighboring B-nodes/Posts known through Hello messages. If the virtual region boundary intersects the circle of S_i at d_{eval}, the sensor marks points on the perimeter that lie outside the virtual boundary of region. The segments on the perimeter that lie outside the boundary (e.g., segment of S_i between $b1$ and $b2$ in Figure 4.5) are assigned detection probability of 1, implying that the sensor does not need to calculate coverage for this part of the segment as it is out of the region boundary.

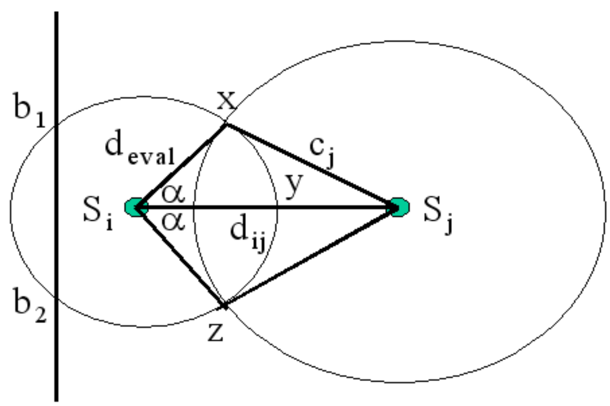

Figure 4.5: Neighbor contribution to coverage

In the next step, neighbor's contribution towards detection probability is calculated. Although neighbors that are within a distance of $d_{eval} + R_{effec}$ from S_i contribute coverage to S_i's perimeter at d_{eval}, we only consider neighbor contribution from sensor nodes within a distance of 2 * d_{eval} ($d_{eval} < R_{effec}$). Note that we have adopted a conservative approach by discarding the coverage contributions from circles greater than d_{eval}. This not only helps in reducing the number of nodes that need to be considered for coverage calculations but also ensure adequate coverage in region between d_{reqd} and d_{eval}.

A sensor Sj that is a neighbor of Si has several concentric circles representing regions of different detection probabilities (Figure 4.6). These circles can be evaluated at fixed distance increments or at fixed detection probability decrements from the node, the value of distance increment (or probability decrement) being a tradeoff between the computational time and detection granularity. Sensor S_i

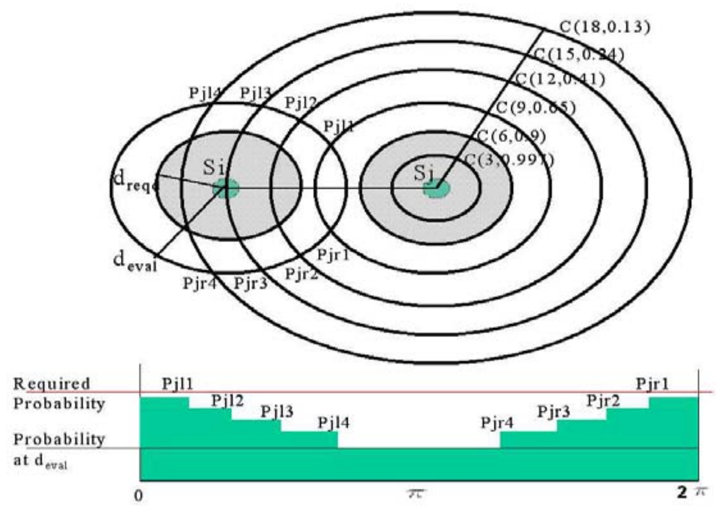

Figure 4.6: Perimeter coverage using PCA

calculates the cumulative detection probability at intersection of its circle at d_{eval} with various circles of neighbor S_j. Lets look at an example shown in Figure 4.5. S_i calculates the cumulative detection probability using Equation 4.5 at the point x, the intersection of its circle with radius d_{eval} with its neighbor S_j circle c_j. The segment on perimeter that is covered by the circle c_j is calculated using the cosines rule.

$$\cos\alpha = \frac{(d_{eval}^2 + d_{ij}^2 - c_j^2)}{(2 \cdot d_{eval} \cdot d_{ij})} \tag{4.6}$$

where, α is the angle subtended by the segment xy on perimeter of S_i. Coverage on segment yz is similar to that in segment xy as total angle subtended by segment xz is 2α. This calculation is repeated for all the circles of the neighbor S_j that are intersecting S_i circle at d_{eval} (see Figure 4.6). $C(r, p)$ in Figure 4.6 represent the circle around S_j with radius r providing probability of detection p.

The cumulative detection probability is then placed on a line segment $[0, 2\pi]$ representing the perimeter of S_i at d_{eval} (see lower part of Figure 4.6). This is repeated for each neighbor until whole perimeter is found covered by probability $\rho \geq$ DDP. If this happens, PCA can declare that region around the sensor S_i bounded by the circle with radius d_{eval} is **sufficiently covered**, otherwise, the required detection probability cannot be provided at d_{eval}, and only the region bounded by circle with radius d_{reqd} is sufficiently covered.

The pseudo-code for probabilistic coverage algorithm is listed as Algorithm 2 and some of the related details are discussed here. Line No. 5 in the pseudo-code listing sorts the neighbor list in order of increasing distance. This is to reduce the computational time for the algorithm. As the output of the PCA is a binary decision, perimeter covered or not, calculating the coverage starting from the nearest located neighbor onwards increases the likelihood of terminating the algorithm earlier in case the perimeter is found covered with neighbor contributions. This is because effective neighbors located close to the node making the decision influence the perimeter more than those located farther away.

Lines 9-12 are better explained by looking at the Figure 4.7. In Figure 4.7(a), S_i and S_j are located such that S_i circle at d_{eval} is intersecting with S_j circle with radius d_{reqd} at points a and b. We can observe that the perimeter of S_j with radius d_{reqd} is covered with ρ_{reqd} and the segment of S_i between a-b gets cumulative probability greater than ρ_{reqd}. We, thus, do not need to calculate the cumulative detection probability for segments that intersect with neighbor

Algorithm 2 Probabilistic Coverage Algorithm (PCA)

Notations :

ρ_{reqd} = Desired detection probability
d_{reqd} = Radius of circle around S_i that provides ρ_{reqd}
ρ_{eval} = Detection probability at next circle with $\rho < \rho_{reqd}$
d_{eval} = Radius of circle around S_i providing ρ_{eval}
ρcum_{ij} = Cumulative detection probability of S_i and S_j
R_{effec} = see Definition 1
$G\alpha$ = Angle subtended by the arc on perimeter of sensor S_i circle with radius d_{eval} that is covered by a neighbor
$G\rho$ = Cumulative probability of detection on perimeter of S_i circle with radius d_{eval}
$C_i(x)$ = Circle of S_i with radius x

Input :

ρ_{reqd}
Neighbor locations
Probability table (PT) of probabilities P and distances D (precomputed)

Process :

ascertain ρ_{eval} and d_{eval} from PT
check boundary intersection with circle at d_{eval}
if $C_i(d_{eval})$ lies outside the region boundary **then**
    mark segments on perimeter of $C_i(d_{eval})$ that are outside the boundary as sufficiently covered
sort the neighbor list in ascending order of distance
for each neighbor j **do**
    if $d_{ij} < 2 * d_{eval}$ **then**
        for each circle of S_j in D ($C_j(D_j)$) that intersects with $C_i(d_{eval})$ **do**
            if $D_j < d_{eval}$ **then**
                mark intersection point on perimeter of $C_i(d_{eval})$ as sufficiently covered by ρ_{reqd}
            else
                mark intersection point on perimeter of $C_i(d_{eval})$ as covered by ρcum_{ij}
        update global $G\alpha$ and $G\rho$
        sort $G\alpha$ and $G\rho$ in ascending order on $G\alpha$
        if $G\alpha$ is all covered from 0 to 2π with $G\rho \geq \rho_{reqd}$ **then**
            declare all perimeter at $C_i(d_{eval})$ is sufficiently covered
            end algorithm
declare perimeter at $C_i(d_{eval})$ is not sufficiently covered

circles with radius less than d_{eval}, and such segments can simply be marked as sufficiently covered with ρ_{reqd}. Considering the region bounded by segment a-b in Figure 4.7(a), we want to check whether the probability of detection in the

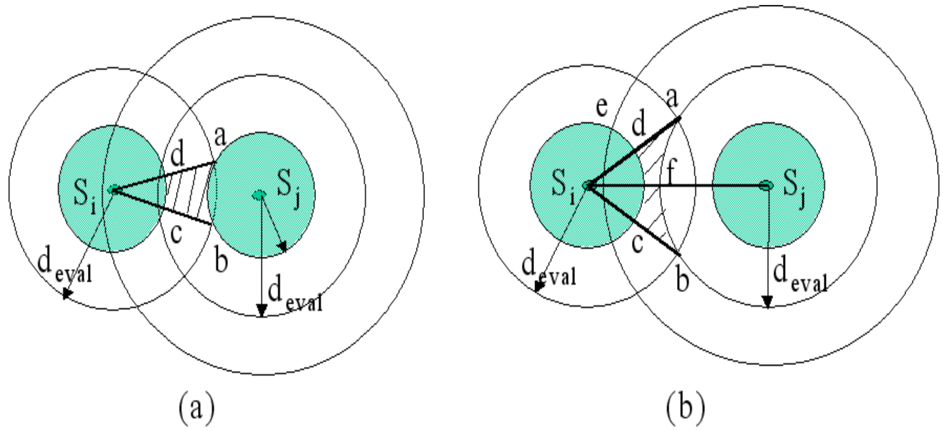

Figure 4.7: Coverage calculations.

region enclosed by a, b, c, and d (marked by slashes) is at least ρ_{reqd} or not. We observe that points a, b, c and d are all covered with probability at least ρ_{reqd} and that as we move in the slashed region from S_j towards S_i, contribution from S_i is increasing while that from S_j is decreasing at the same rate. The minimum cumulative probability in the region would thus occur at the midpoint between the two sensors. We can generalize this in mathematical form as follows. Based on Equation 4.5, we can calculate the minimum probability, ρ_{min}, required out of two sensors, that can be combined to achieve a given ρ_{reqd}.

$$\rho_{reqd} = 1 - ((1 - \rho_{min})(1 - \rho_{min})) \tag{4.7}$$

If μ_{max} represents the maximum decrement in probability such that $\rho_{min} = \rho_{reqd} - \mu_{max}$, we can calculate μ_{max} and the associated distance d_{inc}. The combined detection probability of two sensors (ρcum) located at equal distance from an event is always more than that provided by any of the two sensors (Equation 4.8).

$$\rho_{cum} = 1 - ((1 - \rho_{reqd})(1 - \rho_{reqd})) > \rho_{reqd} \tag{4.8}$$

If μ represent a small decrement in probability, the cumulative detection probability provided by two sensors is given by Equation 4.9.

$$1 - ((1 - (\rho_{reqd} - \mu))(1 - (\rho_{reqd} - \mu))) \geq \rho_{reqd} \tag{4.9}$$

Equation 4.9 holds true for value of $\mu \leq \mu_{max}$. For $\mu = \mu_{max}$, Equation attains minimum value equal to ρ_{reqd}. So, ρ_{cum} at midpoint of the region is at least ρ_{reqd}, if $d_{eval} - d_{reqd} \leq d_{inc}$. With this constraint, the slashed region in Figure 4.7 has combined detection probability of at least ρ_{reqd}. In other words, if ρ_{eval} represent the detection probability at distance d_{eval} then $\rho_{eval} \geq \rho_{min}$. This constraint can be placed on the selection of d_{eval} value from the probability table PT.

Considering the case where the intersecting circle of S_j has radius greater than or equal to d_{eval}, Figure 4.7(b), the segment between points *a*-*b* is marked covered with ρcum_{ij}, cumulative detection probability. Also segment of S_i between *c*-*d* is covered with probability greater than ρ_{reqd}. The probability inside the slashed region increases as we move from segment *a*-*b* towards segment *c*-*d*. Note that as neighbors located within a distance of $2 * d_{eval}$ are only considered in the coverage calculations, this ensures that the enclosed region (slashed) always has contributions from the neighbors.

Finally, line 15 in pseudo-code listing is an early termination check. The algorithm checks whether the desired detection probability has been achieved after calculating the influence of coverage of each neighbor and if so, the result is declared as $C_i(d_{eval})$ sufficiently covered and the algorithm terminates.

Each sensor calculates this perimeter coverage independently and can report whether the region bounded by its circle with radius d_{eval} is sufficiently covered or not. If all the sensors report sufficiently covered perimeters at $C_i(d_{eval})$, the whole region is sufficiently covered. If a sensor finds that its perimeter is not sufficiently covered, it has identified a coverage hole in the region, an area that is not covered to the required detection probability. The information from all sensors describe the current state of area coverage supported by the sensor network. In case of coverage hole detection, this information can be utilized to deploy more sensors in the topology or to guide mobility capable redundant nodes to specific locations to satisfy the detection probability constraint.

4.3.2 Identifying Deployment Points

The basic PCA can be easily extended to not only identify the presence of coverage holes in the region but also to suggest possible deployment points in the region to cover those coverage holes. An uncovered perimeter at circle with radius d_{eval} indicates a coverage hole. This information is readily available after executing the PCA.

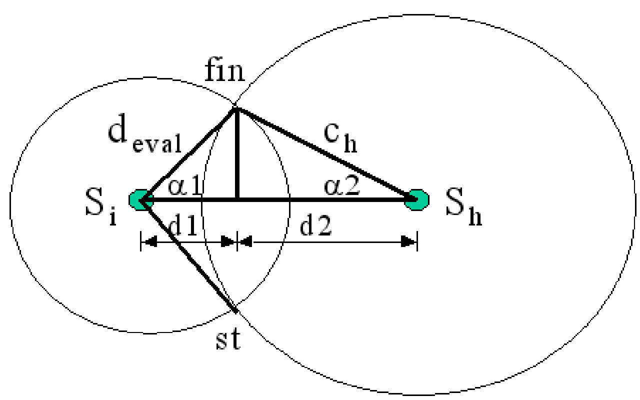

Figure 4.8: Identifying deployment point

Refer to Figure 4.8, st and fin are the start and end points of the maximum uncovered segment (having detection probability $< \rho_{reqd}$) on perimeter of S_i's circle with radius d_{eval}. There can be a number of uncovered segments in the perimeter but the one with the lowest existing detection probability is selected. The task is to determine the deployment location where a redundant helper node,

S_h, can be placed such that the perimeter coverage constraint for the current node is satisfied. Let ρ_{exist} represent the existing detection probability in the uncovered segment ($\rho_{exist} < \rho_{reqd}$), we need to calculate ρ_{help}, the probability required out of helping node that can enhance ρ_{exist} to at least ρ_{reqd}.

$$\rho_{help} = 1 - \frac{(1 - \rho_{reqd})}{(1 - \rho_{exist})} \tag{4.10}$$

ρ_{help}, given by Equation 4.10, is used to index the probability table, PT, to select appropriate distance C_h (radius of S_h), that gives $\rho \geq \rho_{help}$. We refer to probability at this distance as ρ_{select}.

Figure 4.8 illustrates how to calculate the coordinates for helper node S_h once C_h has been selected. Required information for deployment location is the orientation and distance of deployment point from the current node. The required orientation, α_{dep}, is given by $st + \frac{(fin-st)}{2}$. Distance from the current node is divided into d_1 and d_2 (Figure 4.8). d_1 is calculated using Equation 4.11.

$$d_1 = \frac{d_{eval} \cdot \sin(\alpha 1)}{\tan(\alpha 1)} \tag{4.11}$$

For distance d_2, an additional check is made whether C_h, the circle that provide ρ_{select}, can completely cover the uncovered segment between st and fin. If C_h cannot completely cover the segment, we have to place S_h at perimeter of S_i (total distance from S_i is d_{eval}) to ensure maximum possible coverage gain. Thus if $d_{eval} \cdot \sin(\alpha 1) > \rho_{select}$ take $d_2 = (d_{eval} - d_1)$ otherwise use Equations 4.12 and 4.13 to calculate d_2.

$$\alpha 2 = \sin^{-1}(\frac{d_{eval} \cdot \sin(\alpha 1)}{\rho_{select}}) \tag{4.12}$$

$$d_2 = \frac{d_{eval} \cdot \sin(\alpha 1)}{\tan(\alpha 2)} \tag{4.13}$$

The orientation of the required deployment is known (α_{dep}) and the distance from the node is given by $d_1 + d_2$. This information can easily be resolved into the coordinates for deployment. The sensor can advertise this location for help.

4.3.3 Request Aggregation

The PCA enables each sensor to evaluate whether it has uncovered perimeter and to request deployment of an additional sensor if an uncovered perimeter is detected. Each sensor runs this algorithm in a distributed manner. Several sensors facing a common coverage hole thus discover uncovered perimeters and each can send a request for additional deployment. However, deployment of a single additional sensor may benefit other neighbors facing the same coverage hole. In such a case, we can aggregate requests to reduce the communication overhead of the greedy approach when each sensor is independently sending its own request for deployment.

A node that discovers an uncovered perimeter starts a random wait timer. Upon expiry of this timer, the node broadcasts a **Check** message announcing an uncovered perimeter and the requested deployment point. Neighboring nodes with uncovered perimeters, that are still in their random wait period, cancel their timers and ascertain whether the requested deployment point fulfills their coverage requirement or not. A node checks if the requested deployment point is within twice d_{eval} from its own location and if so, it calculates the potential contribution of a node at the requested deployment point on its uncovered perimeter. If the perimeter coverage become adequate, the node is done. Otherwise, it recalculates its own deployment point and restarts the random wait timer.

This wait and check mechanism ensures that deployment requests are aggregated. This aggregation of information not only reduces the overhead associated with additional communication but also results in deployment of fewer nodes than that required by the greedy approach.

4.4 Simulation Study

The probabilistic coverage algorithm has been implemented in the discrete event NS2 simulator. Simulation setup parameters are listed in Table 4.2. Figure 4.9 shows the number of nodes reporting sufficiently covered perimeters for PCA and unit disk binary detection model with radius d_{reqd}. d_{reqd} is 6m for ρ_{reqd} value of 0.9 while d_{eval} is 9m providing ρ_{eval} 0.665.

Table 4.2: Simulation setup parameters.

Parameter	**Value**
Transmit power P_t(Target)	24.5 dBm
Receiving threshold (γ) at sensor	-27.85 dBm
Path loss exponent n (free space)	2
σ	4 dBm
Region (A)	100m x 100m
Number of nodes	60,80,100,120

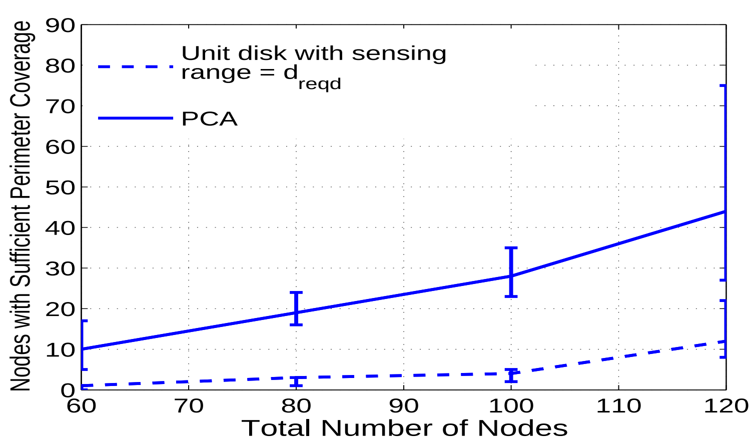

Figure 4.9: Simulation results.

With 60 nodes in 100 x 100 m region, on average PCA reports perimeter coverage at 9m circle for 10 nodes while binary detection model has only 1 node with whole perimeter covered with required probability of 0.9. At higher node density of 120, the corresponding average values are 44 for PCA and 12 for binary detection model. The results are averaged for three different randomly generated topologies for each different number of nodes.

Figure 4.10 shows the coverage provided by PCA and unit disk models by 60 node deployed in a 100m x 100m region. For unit disk with 6m sensing range, only two nodes have sufficient perimeter coverage (shown by thick perimeter). For PCA, 21 nodes have sufficient perimeter coverage due to neighbor's contributions. Note that for clarity, the virtual boundary formed by B-nodes has not been shown. It is clear that the binary detection model (with radius = d_{reqd}) underestimates the total coverage. It result in the deployment of more nodes than that are actually required to satisfy the coverage constraint.

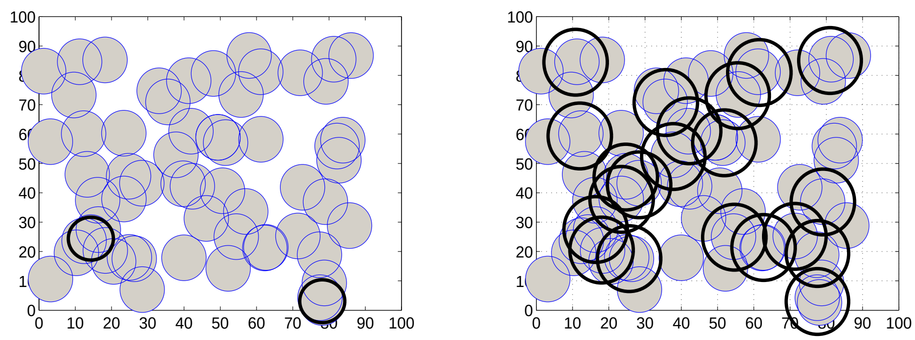

Figure 4.10: Coverage provided by unit disk and probabilistic models.

The value of distance increment used in the probabilistic coverage algorithm controls the computational complexity. Intuitively, lower distance increments for the probability table will result in higher precision in estimating the area coverage. If n represent the number of neighbors for a node, and m represent the number of discrete distances in the probability table, lines 6 and 8 in the pseudo-code for PCA results in $n \times m$ complexity (ignoring the $nlogn$ sorting part of the al-

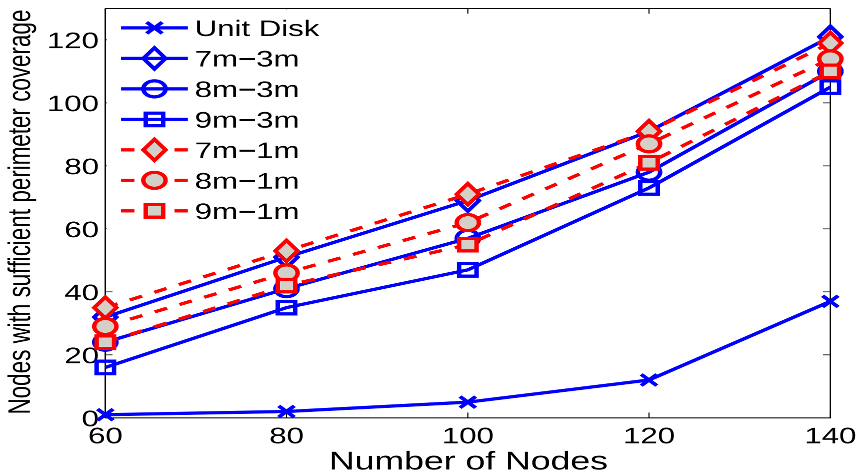

Figure 4.11: Effect of granularity on performance of PCA.

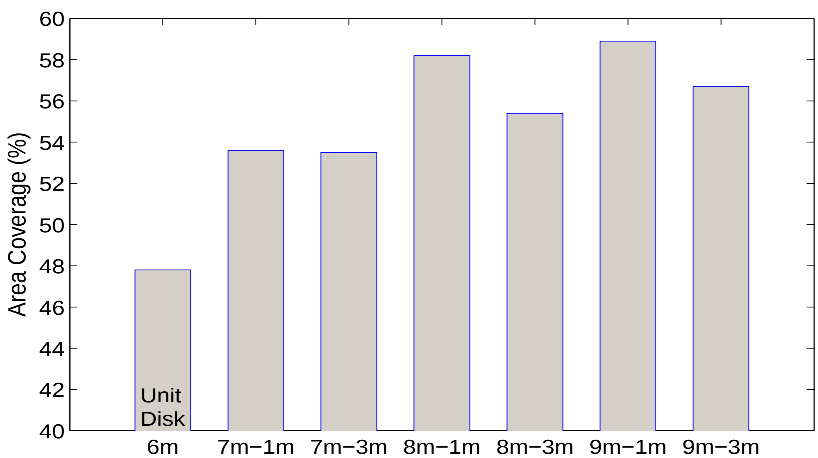

Figure 4.12: Coverage by unit disk and PCA.

gorithm). For a given number of neighbors, choosing a lower value of distance increment for circles will result in a higher value of m with corresponding increase in computational complexity. However, there is a tradeoff between the granularity/accuracy and the computational time of the algorithm. We ran different simulations to study the effect of this approximation on coverage estimation using different values of m. Distance increments used are 3m, 1m, and 0.5m for each d_{eval} values of 7m, 8m, and 9m. Note that values of d_{eval} greater than 9m are not considered as these do not satisfy the constraint set by Equations 4.7 and 4.9. Value of d_{reqd} is taken as 6m with 0.9 as the required probability of detection. Different random topologies with 60, 80, 100, 120, and 140 static nodes are generated in 100m x 100m region. Figures 4.11 and 4.12 show a subset of the simulation results. The unit disk model with sensing range equal to 6m reports the lowest number of nodes with sufficient perimeter coverage e.g., for 140 nodes topology, only 37 nodes have adequate perimeter coverage. For both 1m and 3m distance increments, the lower the value of d_{eval}, the higher the number of sufficiently covered nodes. Comparing results for 1m and 3m increments for values in the probability table, lower distance increments result in more nodes being identified as sufficiently covered due to precise probability calculations. For d_{eval} values closer to d_{reqd} (e.g., 7m), increase in precision has little effect on the overall result, while for higher d_{eval} values, using lower distance increments results in more nodes being identified to be perimeter covered.

Note that more number of nodes with a smaller radius reporting sufficient perimeter coverage does not necessarily means that larger area is covered. This is illustrated in Figure 4.12 for 60 nodes topology in a 100m x 100m region. Unit disk model with 6m as the sensing range has 47.2% of the target area as covered. Coverage provided by 7m (for both 1m and 3m increments) is about 52% while maximum coverage is estimated as 58.8% in case of d_{eval} value of 9m and distance increment of 1m. For calculating the area coverage, nodes reporting sufficient perimeter coverage are assigned the d_{eval} value as the coverage radius while nodes with insufficient coverage use d_{reqd} as the coverage radius.

4.5 Experiments

In this section, we describe results of the coverage related experiments conducted on MicaZ [19] and Tmote sky [20] devices.

4.5.1 Acoustic Experiments

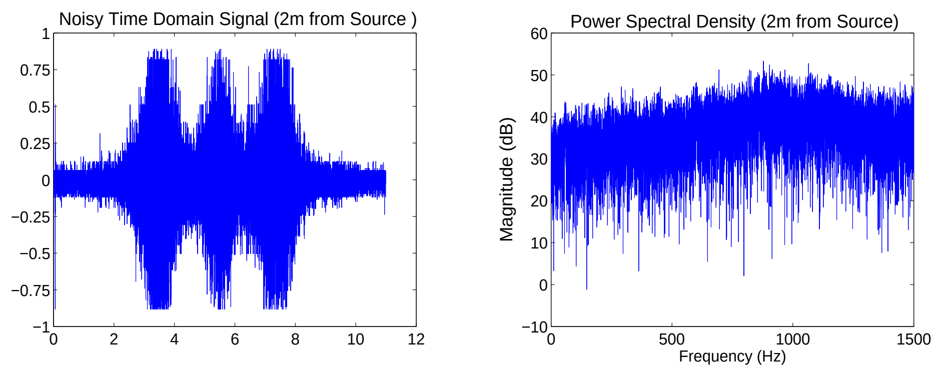

Figure 4.13: 2m: Time domain and power spectral analysis.

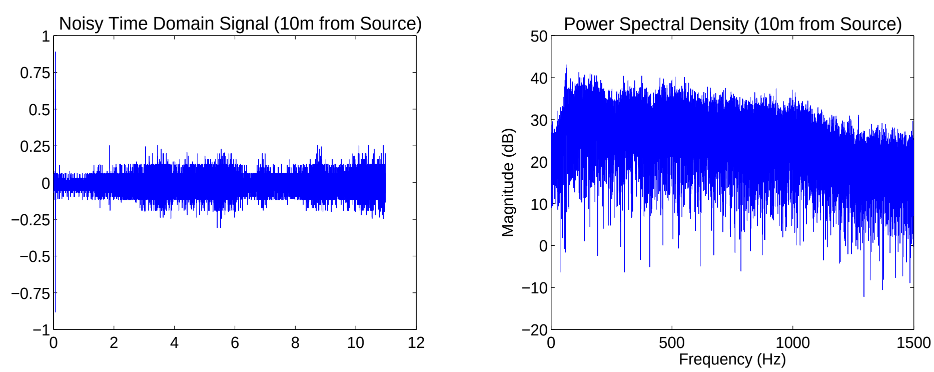

Figure 4.14: 10m: Time domain and power spectral analysis.

We conducted acoustic experiments with MicaZ Motes to ascertain the sensing range provided by onboard microphone and to verify the suitability of probabilistic model for acoustic data captured using real sensor hardware. As the highest frequency that can be analyzed is half of the sampling frequency, we

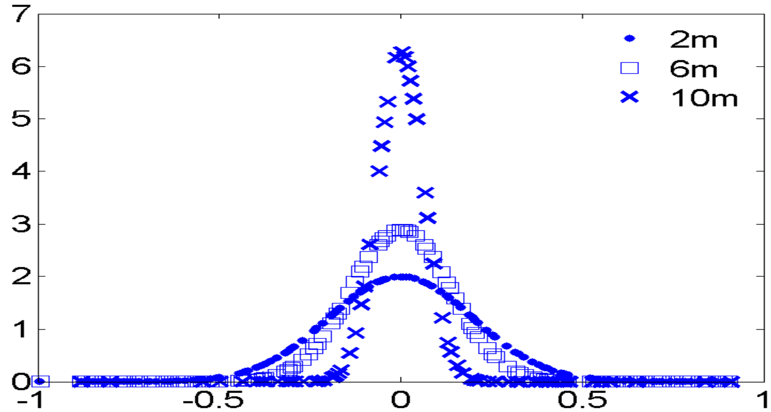

Figure 4.15: Probability distribution function.

tried to achieve the maximum possible sampling rate out of the Mote platform. In our experiments, with the maximum attainable transmission rate of about 160kbps for MicaZ Motes, the maximum achievable sampling rate was about 2000Hz. To attain a higher sampling rate, we directly plugged the MicaZ Mote into the MIB510CA serial interface board [19] that provides interface for both programming and data communications. We used 3000Hz sampling rate with this arrangement.

Acoustic data was collected in roadside experiments with the standard microphone on-board MicaZ with a constant gain value. Data was collected at 2m, 4m, 6m, 8m, and 10m distance from the road for acoustic generated by passenger cars in a 50km/h local traffic area. Three different set of experiments were conducted at each distance. The raw time domain data was normalized and off-line analysis of the data was performed by converting the time domain data into the frequency domain, by Fast Fourier Transforms, FFT (performed off-line in Matlab) to get the power spectral density. Results for 2m and 10m distances are shown in Figures 4.13 and 4.14. We did not perform any matched filtering necessary for classification purposes as we were only interested in the binary detection decision by the Motes.

At distances greater than 8m, it become extremely difficult to detect any acoustic activity due to low SNR. We constructed the normalized probability distribution function (Gaussian distribution) of the captured data at different distances. The plot in Figure 4.15 shows that data collected at short distances has higher variance than that collected at long distances. The data thus reflects that the detection capability falls considerably with increase in the distance, and beyond a certain distance (8-10m in our experiments) it become hard to distinguish between acoustic activity and background noise. Variance in same distance readings are also observed (not shown) due to various factors such as scattering, ground reflection, absorption etc. This observed behavior reflects the probabilistic nature of the acoustic data and its suitability for modelling by probabilistic coverage.

4.5.2 Implementation of PCA on Tmote Sky

An integrated implementation of PCA and the boundary estimation algorithm (Chapter 3) has been done on Tmote sky Mote platform. The PCA is computationally intensive as it involves the use of trigonometric functions to evaluate the intersection of circles and to compute the combined probabilities. One of the objectives of this implementation was to ascertain the suitability of the resource constrained Tmote sky platform to run this computationally extensive algorithm.

The integrated implementation was first tested on TOSSIM, the TinyOS simulator. After testing the implementation, small scale indoor experiments with 15 Tmotes were conducted. Each Mote applies the PCA on the list of neighbors and reports to the base station whether its perimeter at d_{eval} is sufficiently covered or not. The collected data at base station was then analyzed for the coverage.

The results with several randomly generated topologies were in accordance with the theoretical simulation results. Note that the radio irregularity discussed in Chapter 3 has little impact on the performance of PCA. This is because of the fact that all coverage calculations are performed locally by each Mote on the reliable neighbor list already maintained by the boundary estimation algorithm.

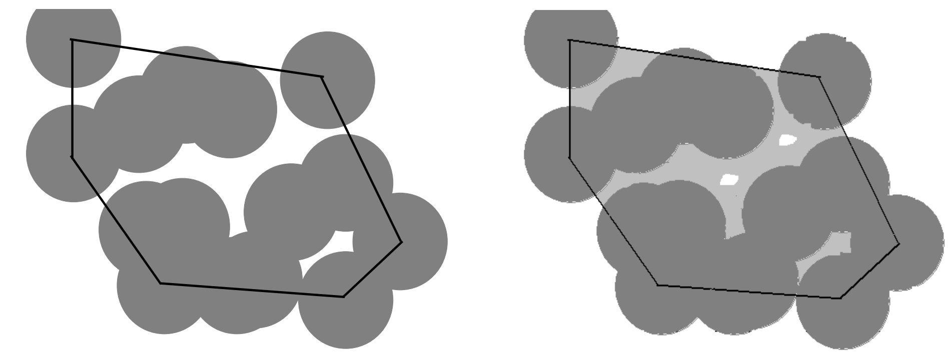

Figure 4.16: Experimental results: Uniform topology

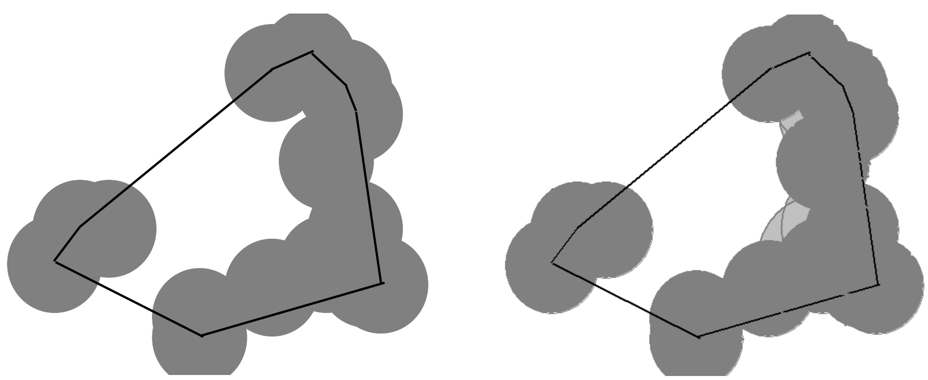

Figure 4.17: Experimental results: Sparse topology

Figures 4.16 and 4.17 show results of running the experiment with 15 Motes in uniform and sparse topologies. Coverage with both unit disk model and by applying the PCA is shown. These Figures also show the part of wasted coverage i.e., lying outside the virtual boundary. In case of uniform topology, the coverage is increased from about 70% with unit disk model to about 99% by applying the

PCA. For the sparse topology, the coverage increase is minimal as most of the Motes are placed along the boundary.

4.6 Discussion

Based on the simulation and experimental evaluation of the PCA described in the previous sections, we can highlight the following algorithm characteristics:

- The algorithm is distributed with all computations performed locally at each node.

- PCA can identify the presence of coverage holes and can request deployment of additional nodes for plugging these identified coverage holes.

- PCA cannot result in reduction of coverage compared to that provided by the unit disk model with radius set to d_{reqd}. This is because the algorithm falls back to the unit disk model if the evaluated distance does not provide any increase in detection probability. Instead, it adds the possibility to extend the radius of sufficient coverage by calculating the cumulative detection probability when multiple sensors influence the area under consideration.

- The algorithm performance depends on the distribution of the sensors in the topology. The coverage increase is higher, if sensors are uniformly distributed in the topology. Coverage increase is minimal for sparse deployment, deployments with clusters, and with a lot of boundary nodes.

4.7 Summary

This thesis has proposed a probabilistic coverage algorithm to evaluate area coverage in a randomly deployed wireless sensor network. The proposed algorithm takes into account the variations in sensing behavior of deployed sensors and adopts a probabilistic approach in contrast to widely used idealistic unit disk model. Simulation and experimental results confirm the effectiveness of the proposed algorithm in predicting the degree of confidence in detection probability supported by a given deployment.

Chapter 5

Coverage Enhancement Using Mobile Sensor Nodes

5.1 Introduction

In hostile or harsh environments such as enemy territories in battlefields, fire or chemical spills, it is impossible to deploy the sensor nodes in a predetermined regular topology. Random (possibly aerial) deployment of sensor nodes is a solution in such scenarios. However, such random deployments are highly susceptible to the occurrence of coverage holes in the target area. There are many factors that contribute to this including the presence of obstacles, sloping grounds like hills, strong winds or dense forestation during aerial deployment, etc. Coverage holes are thus expected even in dense random deployment of nodes. A potential solution for enhancing the existing coverage involves the use of mobility capable sensors [54], [63], which would help fill in the voids. In this chapter, we explore how to efficiently utilize the mobility capabilities of mobile nodes to enhance the existing coverage provided by the randomly deployed static nodes.

Mobile nodes are relocated to strategic positions in the topology. This relocation of mobile sensors serves two important purposes. Firstly, it aims at moving the mobile sensors in an optimal manner to plug the discovered coverage holes.

This increases the coverage during deployment. Secondly, the additional mobile sensors are uniformly spread in the target area for further discovery of coverage holes. Spreading out the redundant mobile sensors to form an even distribution over the target area is useful for a number of other purposes for example, clustering wherein the mobile sensors can act as the cluster heads, formation of an overlay network consisting of mobile sensors, and hierarchical routing, etc. More importantly, this uniform distribution minimizes the reaction time in relocating a mobile sensor once a coverage hole is detected in the area during the operational lifetime of the WSN. It is pertinent to note that there is no loss in existing coverage that is associated with this movement as the mobile nodes do not participate in the probabilistic coverage calculation (Phase-I of the MAPC protocol discussed in Chapter No 4).

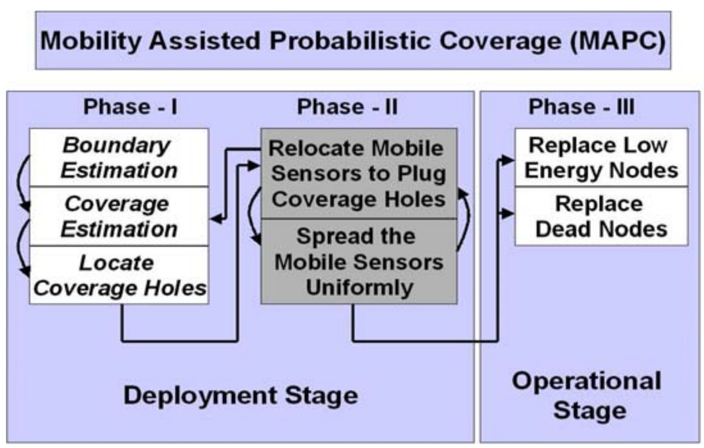

Figure 5.1: Building blocks of MAPC protocol.

This chapter details phase-II of the MAPC protocol as highlighted in Figure 5.1. Static sensors nodes execute phase-I to estimate the area coverage, discover the presence of coverage holes, and request deployment of mobile nodes to cover these identified coverage holes. In phase-II some of the mobile nodes are moved to plug these discovered coverage holes, and the rest of the mobile nodes are spread evenly in the target area.

For spreading of the mobile sensor nodes, this thesis proposes to use the concept of Virtual Forces from robotics [54]. In this context, we propose two variants of **Virtual Force Algorithm (VFA)**: Coverage and Energy Aware VFA (CEA-VFA) and CEA-VFA with Simulated Movements (SM). These variants have been designed to efficiently locate and plug the coverage holes (hence called coverage aware) and to uniformly distribute the remaining mobile nodes in the target area with minimal energy consumption. The algorithms works in rounds. In each round, the algorithms first moves the mobile nodes to plug in the coverage holes identified by neighboring static nodes. Once this is done, remaining mobile nodes are spread out in the region to enable the discovery of more uncovered areas in subsequent rounds.

The CEA-VFA approach is different from the basic VFA in several aspects. Firstly, CEA-VFA has been made coverage aware. It locates the coverage holes and tries to fill them first before participating in the VFA. Secondly, the VFA has been made energy aware. The forces exerted on neighbors are made proportional to the current energy level of the nodes. This helps in deploying mobile sensors without depleting the battery of some of the mobile nodes. Thirdly, in CEA-VFA the movements of the mobile nodes are controlled by novel thresholds based on real radio characteristics. This ensures that the nodes can communicate with each other, with high probability of successful transmission, during the round-by-round operation of the VFA. Fourthly, we have incorporated a practical way of modelling the boundary of the region by using static nodes as the B-nodes (and Posts). The boundary nodes exert repulsive forces on mobile nodes to model the virtual boundary identified by the B-nodes (phase-I, Chapter No 3).

Random aerial deployment is extremely challenging for mobile nodes as it may result in physical damage to their locomotive parts. In realistic deployments, the mobile nodes are normally accumulated at one or more points near the target area. We therefore, consider two different initial deployment methodologies namely **Normal** and **Island** distribution. In normal distribution, mobile sensors form a single cluster at the boundary, while in island distribution they form disconnected clusters at different locations on the boundary. Note that managing the mobile node's relocation for these initial deployments is more challenging than assuming randomly distributed mobile nodes.

We make the following assumptions:

- Location information is available. Given that form factor and cost of GPS is prohibitive in the WSN, we assume any existing GPS-less WSN localization scheme for the mobile nodes such as [88].
- The target area is an unknown obstacle-free environment.
- Mobile nodes have significantly more initial energy than the static nodes. For example, the initial energy of a Robomote [89] is 4528J (3.7V Lithium battery), while that of a Mica2 motes is about 3000J (2 AA batteries).

Th two proposed movement calculation algorithms, CEA-VFA and SM are explained in detail in later sections.

5.2 Coverage and Energy Aware Virtual Force Algorithm (CEA-VFA)

This thesis presents a coverage and energy aware variant of basic VFA proposed in [54], [50]. Basic VFA attempts to iteratively spread the mobile sensors in the target area by using a combination of attractive and repulsive forces. Sensor nodes are pulled toward each other if the distance between them is more than a predefined threshold distance. If the mobile sensors are placed too close to each other, they are pushed apart, ensuring that the sensors are not clustered.

Two mobile sensors will exert virtual forces on each other if the Euclidean distance, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, between them is not between a given range of thresholds, Th_{push} and Th_{pull} (discussed in detail later in this section). This virtual force, F_{ij} is a pull or attractive force if the distance between the two mobile nodes is greater than the pull threshold, Th_{pull}, while if the distance is less than the push threshold, Th_{push}, a push or repulsive force is exerted. Equation 5.1 shows the model used for decision making.

$$\vec{F}_{ij} = \begin{cases} F_{push}, & \text{if } d_{ij} < Th_{push} \\ 0, & \text{if } Th_{push} \leq d_{ij} \leq Th_{pull} \\ F_{pull}, & \text{if } d_{ij} > Th_{pull} \end{cases} \tag{5.1}$$

where $\vec{F}_{ij}$ is the force exerted on mobile node Si by neighbor Sj. Taking the midpoint of range between Th_{push} and Th_{pull} as the desired distance between the mobile nodes, Equations 5.2 and 5.3 represent the push/pull virtual forces.

$$F_{push} = (\frac{Th_{pull} + Th_{push}}{2}) - d_{ij} \tag{5.2}$$

$$F_{pull} = d_{ij} - (\frac{Th_{pull} + Th_{push}}{2}) \tag{5.3}$$

Representing magnitude of the force in terms of distance, the share of the virtual force for a node is given by Equation 5.4. We represent $\vec{dm}_{ij}$ as the force absorbed by node i due to j, the magnitude of which is denoted by $|dm_{ij}|$.

$$\vec{dm}_{ij} = \frac{\vec{F}_{ij}}{2} \tag{5.4}$$

where $|dm_{ij}| = |(F_{ij}/2)|$.

We can express the total force exerted on a mobile sensor S_i by its n mobile neighbors, denoted by $\vec{F}_i$, as,

$$\vec{F_i} = \sum_{j=1, j \neq i}^{n} \frac{\vec{F_{ij}}}{2} = \sum_{j=1, j \neq i}^{n} \vec{d}m_{ij} \tag{5.5}$$

Note that $\vec{F_i}$ is the vector sum of all the forces acting on mobile sensor node S_i, the magnitude and orientation of which can be easily calculated, e.g., Robomote [89] uses an on-board compass combined with localization information for navigation purposes. These virtual force calculations are performed in each round of the virtual force algorithm and mobile nodes are iteratively moved to attain a more uniform distribution in the region.

Having explained the concept of virtual forces and the virtual force algorithm, we now detail the changes incorporated in the basic VFA in the following sections.

5.2.1 Coverage Awareness

We introduce coverage awareness in basic VFA to enable mobile nodes to identify and fill the existing coverage holes in the topology. In CEA-VFA, a mobile node first checks whether it is in the vicinity of a coverage hole and if so, it reacts to plug in the discovered coverage hole before participating in the virtual force calculations.

This coverage check is performed by each mobile node in each round of the CEA-VFA. The coverage check starts with mobile nodes exchanging the location information using **Hello** messages. Mobile nodes start a TM_{wait} timer. A mobile node only starts the virtual force calculations on the expiry of this timer. B-nodes also send the Hello message with a special flag identifying them as B-nodes. A Hello message contains the sender node ID, current location coordinates, and the remaining energy. A mobile node on reception of a Hello message marks the sender as its current neighbor and stores the information contained in the message. The TM_{wait} timer is reset each time on the reception of a Hello message.

Static nodes with uncovered perimeter starts a TS_{wait} timer on receiving a Hello message from the neighbor mobile nodes. This timer is reset each time a Hello message is received. On expiry of the TS_{wait} timer, a static node with uncovered perimeter selects the nearest mobile node from its mobile node neighbor list and sends a **Help** message to the selected mobile node. The Help message contains sender node ID, location, requested deployment point, and the expected gain in coverage (difference between the required detection probability ρ_{reqd} and the existing probability at uncovered perimeter ρ_{exist}).

If a mobile node receives Help message from a single static node only, the relocation decision is trivial. But intuitively, a mobile node may receive multiple Help messages from different static nodes with uncovered perimeters. A mobile node selects the requested deployment point that involves the highest gain in coverage, broadcasts a **Move** message, and starts moving toward the requested deployment point. As the Move message is a broadcast, it is received by both mobile and static sensor nodes.

Neighboring mobile nodes that receive the Move message, removes the sender node from the list of neighbor mobile nodes (as the mobile node has left its previously announced position). The TM_{wait} is reset when a new Move message is received. A mobile node starts the virtual force calculation on the expiry of the TM_{wait} timer.

Static nodes with uncovered perimeters also receive the Move message. These nodes check whether the Move message is in response to their Help message. If not, these static nodes now send a Help message to the next candidate mobile node from the neighbor list. If no Move message is received from this 2nd mobile node, the static node waits for the next round of Hello messages. This 2nd choice check is to cover static nodes that may not have received any Move messages due to mobile nodes initially selecting other preferable choices.

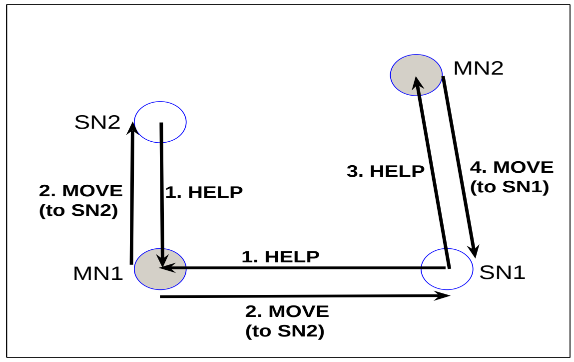

Figure 5.2: Contention resolution mechanism.

Figure 5.2 shows this contention resolution using Help and Move messages. Note that all these messages are exchanged with 1-hop neighbors only. The mobile nodes after expiry of the TM_{wait} timer use the CEA-VFA to spread out in the topology to discover more coverage holes.

5.2.2 Energy Awareness

To ensure that the mobile nodes do not exhaust all of their energy during the deployment phase, we introduce an adaptive policy based on the residual energy of the nodes. The virtual forces are made proportional to the residual node energy, with a higher energy node absorbing a greater portion of the virtual force than its low energy neighbor. This also helps in minimizing the variations in the nodes remaining energies. Let E_I represent the initial energy of a mobile node, E_{ci} and E_{cj} represent the current remaining energies of nodes S_i and S_j respectively. We have $e_{ij} = (E_{ci} - E_{cj})/E_I$, where e_{ij} is the proportional energy coefficient. Representing magnitude of the force in terms of distance, the share of the virtual force for a node is given by Equation 5.6.

$$\vec{dm}_{ij} = \frac{\vec{F}_{ij}}{2}(1 + e_{ij}) \tag{5.6}$$

where $|dm_{ij}| = |(F_{ij}/2)(1 + e_{ij})|$.

We can now express the total force exerted on a mobile sensor S_i by its n mobile neighbors, denoted by $\vec{F}_i$, as,

$$\vec{F}_i = \sum_{j=1, j \neq i}^{n} \frac{\vec{F}_{ij}}{2}(1 + e_{ij}) = \sum_{j=1, j \neq i}^{n} \vec{d}m_{ij} \tag{5.7}$$

5.2.3 Choice of Thresholds, Th_{push} and Th_{pull}

We want to ensure that the spreading of the mobile nodes results in a connected topology after termination of the virtual force algorithm. We also want to ensure that mobile nodes are able to communicate with neighbors (mobile nodes) during round by round operation of the algorithm. For this purpose, rather than using static values of movement triggering thresholds, Th_{push} and Th_{pull}, these thresholds are made dependant on the link quality. For better quality communication links, sensor nodes can be placed further apart (higher value of Th_{pull} can be used). Although other complex models can also be used, we use a simple radio propagation model based on log-normal shadowing (discussed in Chapter 4, Section 4.2) to characterize the communication link between two sensors. The path loss using log-normal shadowing model can be measured experimentally for a given event and sensor characteristics or can be calculated using the free space path loss model [84].

The path loss PL(in dB) at a distance d from a sensor is given by Equation 4.1 (Chapter 4). Each sensor has a receive threshold value γ that describes the minimum signal strength that can be correctly decoded at the sensor. The probability P_{ri} that the received signal level, P_{rec} at a sensor will be above this receive threshold, γ, is given by Equation 4.4. For a given transmit power and receive threshold value, we can calculate the probability of receiving a signal above the receive threshold value, γ, at a given distance using Equations 4.2 and 4.4 as shown in Figure 5.3.

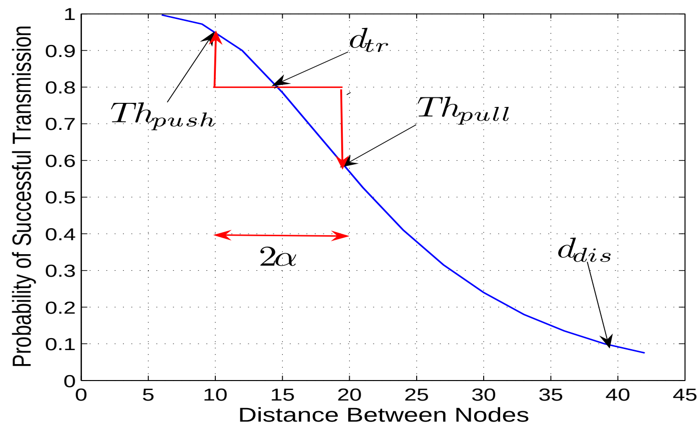

Figure 5.3: Change in communication probability with distance.

Following the terminology in [90], there are three distinct reception regions in a wireless link: connected, transitional, and disconnected. The transitional region has highly unreliable links and its region bounds can be found either by analytical or empirical methods [90]. Let d_{tr} and d_{dis} represent the points where the transitional and disconnected regions begin respectively. We define Th_{push} and Th_{pull} as $(1-\alpha)d_{tr}$ and $(1+\alpha)d_{tr}$ respectively, where α denotes the error tolerance coefficient. Note that the values of Th_{push} and Th_{pull} are bounded by d_{dis}. α is dictated by the application and it reflects the tolerance to the errors in localization and odometry during navigation of the mobile nodes. As long as the position after movement is within this range, the deviation from the ideal trajectory during movement can be tolerated by our movement algorithm. Figure 5.3 show the d_{tr}, Th_{push}, Th_{pull} and d_{dis} in terms of probability of correct packet reception.

5.2.4 Boundary Considerations

To ensure that mobile sensors are confined within the boundary of the region while moving in the unknown environment, B-nodes (identified in phase I, Chapter 3) exert virtual forces ($\vec{F}_{ib}$) on mobile sensor nodes. A repulsive virtual force $\vec{F}_{ib}$, based on the mobile node's distance from a B-node, is included in the resultant vector sum of all virtual forces. The new total force is then expressed as,

$$\vec{F_i} = \sum_{j=1, j\neq i}^{n} \frac{\vec{F_{ij}}}{2}(1 + e_{ij}) + \sum_{b=1}^{k} \vec{F_{ib}} \tag{5.8}$$

where $\vec{F_{ib}}$ is the repulsive force exerted on the mobile node Si by its k neighbor B-nodes and is modelled by Equation 5.9.

$$\vec{F_{ib}} = \begin{cases} \vec{F_b}, & \text{if } d_{ib} < \frac{(Th_{push} + Th_{pull})}{2} \\ 0, & \text{otherwise} \end{cases} \tag{5.9}$$

The repulsive force $\vec{F_b}$ is given by Equation 5.10 where d_{ib} is the distance between the node and the B-node.

$$\vec{F_b} = \frac{(Th_{push} + Th_{pull})}{2} - d_{ib} \tag{5.10}$$

As B-nodes are all static nodes, mobiles nodes absorb all of the virtual force resulting from the B-nodes. As a final check, mobile nodes should not cross the virtual boundary formed by the known B-nodes.

5.2.5 Per Round Synchronization

As the mobile nodes move variable distances in each round of the CEA-VFA, they require some kind of signaling mechanism to indicate when to start the next round. One expensive way to achieve this per round node synchronization is by flooding synchronization messages. We take a different approach to achieve this per round synchronization by restricting the maximum distance a mobile node can travel in each round of CEA-VFA. Linking this maximum distance, d_{max}, with the energy considerations, a mobile node should not move a per round distance that consumes more than $K_1 \times E_i$ energy, where K_1 is a tunable parameter that controls the maximum energy allowed to be consumed in each round of movement

Algorithm 3 Coverage and Energy Aware Virtual Force Algorithm (CEA-VFA)

Notations :
M_j = set of mobile neighbors
B_k = set of B-nodes
S_m = set of static nodes with uncovered perimeter
d_{max} = maximum distance allowed in each round
TM_{wait} = wait time before executing virtual force calculations
T_{round} = maximum time allowed for each round

INIT :
broadcast Hello message, start TM_{wait} timer
if Hello message received from a neighbor **then**
  if sender is a mobile node **then**
    store Node ID, location and energy in M_j, reset TM_{wait}
  if sender is a B-node **then**
    store Node ID, location and energy in B_k
if Help message received from a neighbor **then**
  store Node ID, location, deployment point and coverage gain in S_m
  reset TM_{wait} timer

Process :
set Round = 0
if S_m is non-empty **then**
  select node with maximum coverage gain
  broadcast Move message
if Move message received from a neighbor **then**
  remove sender from M_j
  reset TM_{wait}
if TM_{wait} expires **then**
  while Round $\leq$ MaxRounds **do**
    for each B-node k in B_k **do**
      calculate $\vec{F}_{ik}$ using d_{ik}, Th_{push}, and Th_{pull}
    for each neighbor j in M_j **do**
      calculate $\vec{F}_{ij}$ using d_{ij}, Th_{push} and Th_{pull}.
    $\vec{F}_i = \sum \frac{\vec{F}_{ij}}{2}(1 + e_{ij}) + \sum \vec{F}_{ik}, j$ in $M_j, j \neq i$, k in B_k
    calculate $(x, y)_{next}$ based on $\vec{F}_i$
    if $d((x, y)_{next}, (x, y)_{current}) \geq d_{max}$ **then**
      calculate $(x, y)_{next}$ based on d_{max}
    check $(x, y)_{next}$ is within the virtual boundary formed by B-nodes and Posts
    move toward $(x, y)_{next}$ and start timer T_{round}
    if T_{round} expires **then**
      set Round = Round + 1
      broadcast Hello message

and E_i is the initial energy. The maximum time, T_{round}, a mobile node may take to complete movement in each round can be calculated as we know the speed, v_m, of a mobile node.

$$T_{round} = \frac{d_{max}}{v_m} \tag{5.11}$$

Each mobile node waits for time T_{round} before broadcasting its new position in Hello messages to start the new round of CEA-VFA. This maximum distance constraint avoids the need for expensive per round synchronization among mobile nodes and conserves energy. More importantly, it helps in avoiding the possibility of mobile nodes reacting to incomplete or inconsistent neighbor information due to HELLO message broadcast or reception while some of the mobile nodes are still moving.

The major drawback of this approach is that each mobile node has to wait for T_{round} even when all of the nodes may have settled in their new positions in a much shorter period of time. This is intuitively expected in later rounds when all nodes move much lesser distances. Avoiding expensive per round synchronization thus comes at the cost of increased delay during the deployment phase of a WSN.

5.2.6 Oscillation Control

Node oscillation is expected without some sort of damping measures. Oscillation can occur in successive rounds of CEA-VFA because of distributed decision making. A node may discover new neighbors, after the movement, that may force it back towards its initial position. As an oscillation control measure, a node should remember its last position from the previous round of movement. If the current resultant virtual force drives it towards the same direction, it only moves half of the calculated distance i.e., resultant virtual force becomes $\vec{F_i}/2$. This will result in the mobile node rapidly converging to its final position without excessive movements due to oscillations.

The pseudo-code for CEA-VFA appears as Algorithm 3.

5.3 Virtual Force Algorithm with Simulated Movement

Mobile nodes move after each round of CEA-VFA due to the virtual force exerted by its neighbors before settling down to their final position in the topology. If we could calculate the final position of a mobile node and move directly to that final position, we can save energy by moving much lesser distances, i.e., through the cut-through paths instead of the zig-zag paths that result from the round-by-round operation of CEA-VFA (Figure 5.4).

We propose CEA-VFA with Simulated Movement (SM) approach that attempts to use the cut-through paths for movement. Nodes go through the CEA-VFA iterations and calculate new position after each round. The difference is that nodes do not physically move after each iteration, rather, they stay at their original position and simply assume the new calculated virtual position. The Hello messages at the start of the next round contain the new virtual positions enabling the recipients to use this updated position information for the upcoming round of CEA-VFA. Nodes only move once they have calculated their final positions.

This simplistic approach has a few disadvantages. B-nodes, static nodes with uncovered perimeters, and new mobile neighbors are discovered by the per round movements in CEA-VFA. If a fully simulated run of CEA-VFA is used, it is highly likely that the mobile nodes may not detect the presence of the other mobile nodes and B-nodes in the region. Similarly, it is not possible to locate all of the coverage holes.

One way to offset this disadvantage is to use intermittent simulated movement instead of fully simulated movement. In the CEA-VFA Intermittent Simulated Movement (ISM) approach, nodes simulate the movement for x number of rounds. Actual physical movement only takes place after every x number of

Algorithm 4 Intermittent Simulated Movement (ISM) Algorithm

Notations :
SimRound = number of simulated rounds before actual movement
M_j = set of mobile neighbors
B_k = set of B-nodes
P_l = set of Posts
Process :

set Round = 0, simulated flag = true
while Round $\leq$ MaxRounds **do**
    for each B-node k in B_k **do**
        calculate $\vec{F}_{ik}$ using d_{ik}, Th_{push}, and Th_{pull}
    for each neighbor j in M_j **do**
        calculate $\vec{F}_{ij}$ using d_{ij}, Th_{push} and Th_{pull}.
    $\vec{F}_i = \sum \frac{\vec{F}_{ij}}{2}(1 + e_{ij}) + \sum \vec{F}_{ik}, j$ in $M_j, j \neq i$, k in B_k
    calculate $(x, y)_{next}$ based on $\vec{F}_i$
    check $(x, y)_{next}$ is within boundary formed by Posts
    if simulated flag = false **then**
        if $d((x, y)_{next}, (x, y)_{current}) \geq d_{max}$ **then**
            calculate $(x, y)_{next}$ based on d_{max}
        start moving toward $(x, y)_{next}$ and start timer T_{round}
        if T_{round} expires **then**
            set Round = Round + 1
    else
        set simulated flag in broadcast message
        use $(x, y)_{next}$ in broadcast message
        set Round = Round + 1
    if Round $\neq$ simulated round **then**
        set simulated flag = false
    broadcast Hello message

rounds e.g., in ISM2 nodes physically move after every second round. Mobile nodes in the simulated rounds also utilize the boundary information advertised by Posts. The mobile node checks that their new calculated simulated position is within the region formed by the coarse virtual boundary announced by the Posts. This helps in safe navigation of the mobile nodes, in simulated rounds, within the area bounded by Posts. For non-simulated rounds, the mobile nodes calculates the actual virtual forces exerted by B-nodes that are within the communication range of the mobile nodes. The pseudo-code for ISM appears as Algorithm 4.

Figure 5.4 shows the paths followed by a node using CEA-VFA, ISM2 and ISM3. In CEA-VFA the node physically moves after each round to a new calcu-

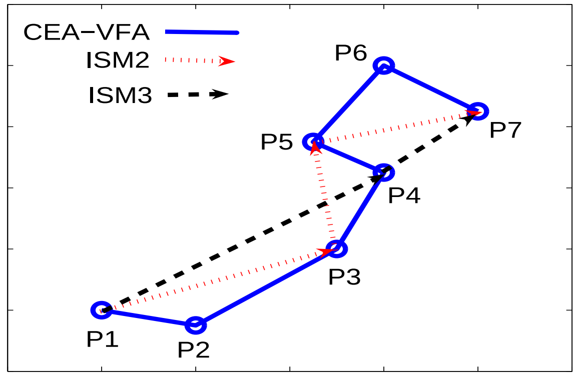

Figure 5.4: Cut through paths followed by ISM2 and ISM3.

lated position. The node thus visits all the intermediate points, P1 through P7 in various rounds of CEA-VFA. In ISM2, the node goes through the first round of virtual force calculations and finds point P2 as the next location where it should move. This node remains physically at point P1 and only advertises its simulated position P2. Virtual force calculations are again performed in the 2nd round using the simulated positions announcements of all the neighbors. The node now directly moves from P1 to P3 after the second round calculations. The node thus physically moves after every second round to points P1, P3, P5, and P7. Similarly for ISM3 the node moves from point P1 to P4 and finally to P7.

The simulated movement approach reduces energy consumption due to movement by moving the mobile sensors in fewer number of rounds. This also results in quicker deployment due to reduction in the time spent in the per round zigzag movements of the CEA-VFA.

5.4 Performance Evaluation

In this section, we discuss the simulation results that describe the performance of the ISM algorithms as compared to CEA-VFA.

5.4.1 A Centralized Optimization

Assignment of mobile nodes to coverage holes identified by static sensor nodes and uniform distribution of the remaining mobile nodes in the topology can be formulated as a matching problem. A centralized optimization can be performed, if we assume that the information of all coverage holes and the total number of mobile nodes is available. Note that this centralized optimization can provide a base line for comparison of the performance of the distributed (and iterative) virtual force-based algorithms. Essentially, the main question that we try to answer is how well does the distributed virtual force based (and simulated variants) really perform compared to optimized (centralized) assignment?

The assignment problem can be formulated as following: Let S_m represent the set of p mobile nodes, $S_m = \{S_{m1}, S_{m2}, \ldots, S_{mp}\}$ and set S_h represent the location of q coverage holes in the topology, $S_h = \{S_{h1}, S_{h2}, \ldots, S_{hq}\}$. After assigning q out of p mobile nodes, we have r remaining mobile sensors, $(r = p - q)$, that needs to be uniformally distributed in the coverage area. The set S_u represent the deployment locations of these r mobile nodes, $S_u = \{S_{u1}, S_{u2}, \ldots, S_{ur}\}$. Let $S_d = \{S_{d1}, S_{d2}, \ldots, S_{dt}\}$ represent the combined deployment locations with $S_d = S_h \cup S_u$ and $t = q + r$.

We want to optimally assign the available mobile sensors, firstly, to locations requested by static nodes and secondly, to locations in the topology so as to form a uniform distribution. This is a classical **Assignment** optimization problem, also referred to as **bipartite weighted matching** problem in graph theory. If a mobile node i is assigned to a location, j, there is a cost (energy consumption due to movement) of c_{ij}. We can minimize the total value of assignment (minimize energy consumption).

The objective function can be defined as:

"What is an assignment schedule in order to minimize energy consumption?"

The assignment problem can be formulated as an Integer Linear Program (ILP) as follows;

$$Minimize \sum_{j=1}^{t}\sum_{i=1}^{p} c_{ij}x_{ij}$$

With additional constraints that each mobile node can be assigned to exactly one location and each location must have one assigned mobile node.

- $x_{ij} \leq 1; i = 1, \ldots, p; j = 1, \ldots, t$
- $\sum x_{ij} \leq 1; j = 1, \ldots, t$
- $\sum x_{ij} \leq 1; i = 1, \ldots, p$

5.4.2 Simulations Results

We implemented the B-node selection algorithm, PCA, basic VFA, CEA-VFA and simulated variants in discrete event simulator NS2 with parameters listed in Table 5.1. Maximum number of rounds was set to 12.

Table 5.1: Simulation setup parameters.

Parameter	**Value**
Th_{push}/Th_{pull}	25m/33m
Initial Energy	4528J (3.7v, 345mAh)
Energy consumed (movement)	8.274 J/m (Robomote [89])

We first compared the performance of basic VFA and CEA-VFA. The basic VFA differs from CEA-VFA in two major aspects. Firstly, the basic VFA is

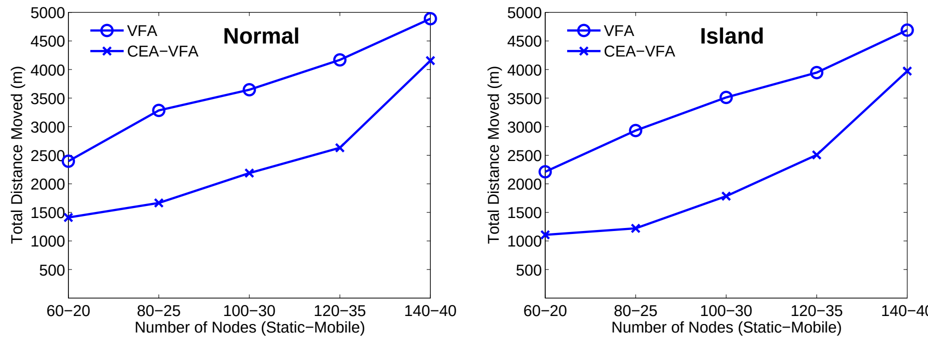

Figure 5.5: Total distance moved: Basic VFA vs CEA-VFA.

not coverage aware. The mobile nodes execute the virtual force algorithm for 12 rounds without interacting with the static nodes. Once the mobile nodes are deployed by VFA, the static nodes send the Help messages to attract their nearest mobile nodes for coverage enhancement. Secondly, in basic VFA the virtual force is always equally shared between the two neighbors irrespective of their current energy levels. In order to perform a fair comparison, basic VFA uses the same link quality-based thresholds, uses B-nodes for boundary forces and has the same oscillation control mechanism as for CEA-VFA.

In Figure 5.5, we compare basic VFA and CEA-VFA in terms of total distance travelled by all mobile nodes. Note that as energy consumption is directly proportional to the distance moved by mobile nodes, an algorithm that moves a smaller distance is more energy efficient. CEA-VFA performs better than the basic VFA for both normal and island initial deployment, for different numbers of mobile nodes. On average, CEA-VFA causes the mobile nodes to move about 63% and 57% of the total distance moved in case of basic VFA for normal and island deployments respectively. The results are primarily due to the integration of coverage awareness in the VFA that enables CEA-VFA to discover coverage holes and react to plug them in each round of the VFA.

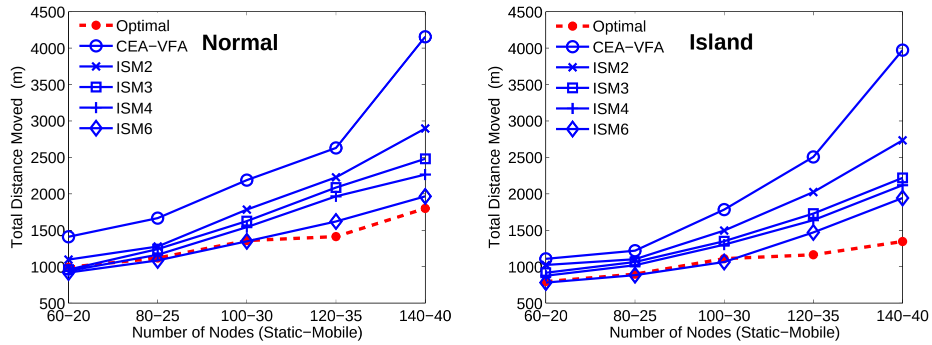

Figure 5.6: Total distance moved by mobile sensors.

Next we conducted detailed simulations for CEA-VFA and different variants of ISM (ISM2, ISM3, ISM4, and ISM6) and compared the results with an optimized assignment. The centralized optimization discussed in Section 5.4.1 is performed in MATLAB. The results are averaged over three different topologies for each type of deployment. Figure 5.6 compares the performance of virtual force based algorithms with a centralized optimized assignment. For obtaining optimized assignment the following procedure was adopted. Suppose X represents the total number of mobile nodes and Y represents the total number of deployment points requested by static nodes. We are left with Z mobile nodes after assigning Y number of nodes out of X, that needs to be uniformly distributed in the region. The region is then divided with Z number of idealized grid points. This gives us Z number of additional idealized grid deployment points. Assignment optimization is then performed with desired matching between $Y + Z$ deployment points and X number of available mobile nodes. Comparing the virtual force-based iterative algorithms with this optimized assignment gives us a measure of two aspects. Are all the coverage holes detected and plugged by the virtual force based algorithm? How well are the remaining mobile nodes spread as compared to a uniform grid deployment?

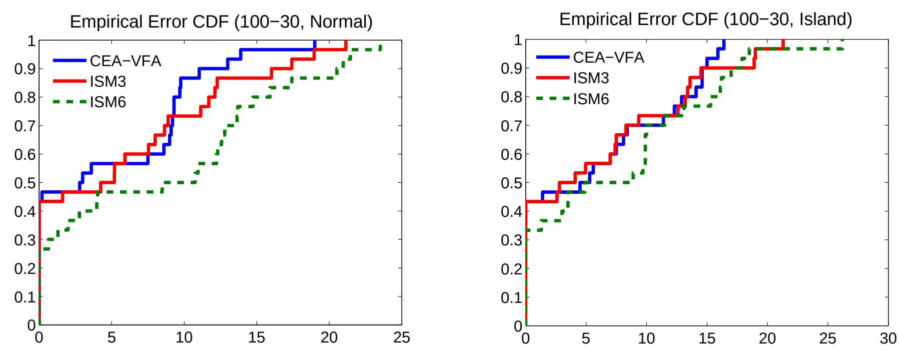

Figure 5.7: Empirical error CDF for normal and island distributions.

The results show that CEA-VFA causes the mobile sensors to move the highest total distance for all types of topologies for both normal and island initial deployment of the mobile sensors. On the other hand, mobile nodes using ISM6 variant consistently move the least distance for all types of deployment. For 140 static-40 mobile nodes topology, mobile nodes using ISM6 move about 45% of the distance moved by CEA-VFA, and about 37% of the distance moved by the basic VFA (Figure 5.5). This is because in ISM6, the mobile sensors only moves in two of the twelve rounds while performing simulations in ten rounds. Also note that in some cases, mobile nodes using ISM6 move a lower total distance than that given by optimized assignment (e.g., 60-20 topology). This is because of the fact that in these cases ISM6 not only fails to discover all of the coverage holes in the topology but it also produces a poor topology distribution by moving lesser distances.

Figure 5.7 illustrates the empirical error Cumulative Distribution Function (CDF) for both normal and island distributions for 100 static-30 mobile nodes topologies. Errors are calculated as the difference between the desired deployment points and the final topology position achieved by different virtual force based algorithms. A zero error means that either a coverage hole has been plugged or a

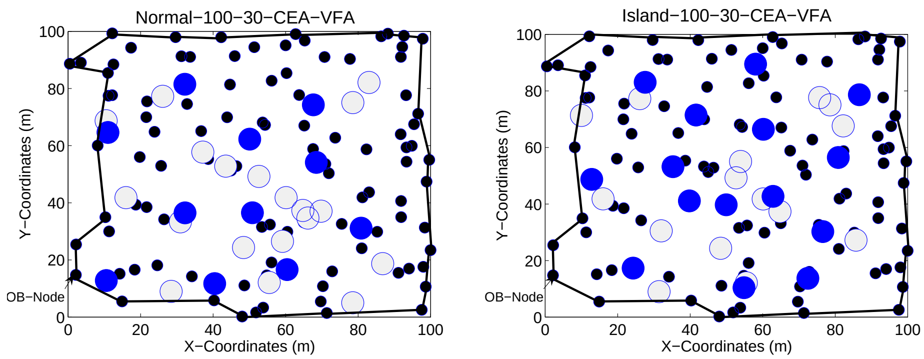

Figure 5.8: Topology distribution achieved with CEA-VFA.

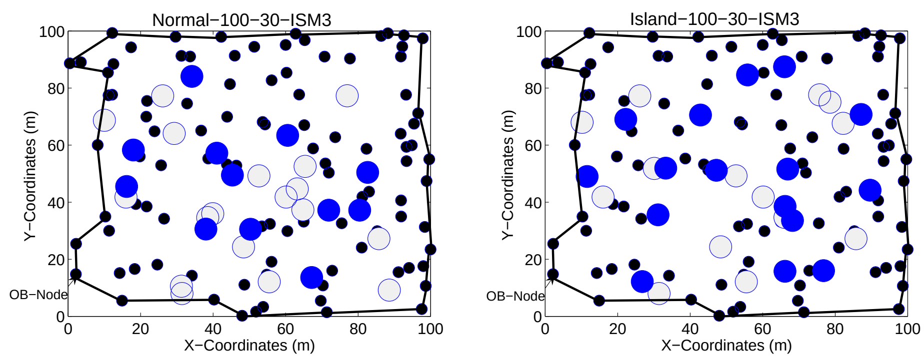

Figure 5.9: Topology distribution achieved with ISM3.

perfect grid point deployment has been achieved. For island distribution, the error CDF of ISM3 matches closely with that of CEA-VFA while ISM3 moves about 55% of the total distance moved by CEA-VFA. For ISM6, lesser number of nodes (about 27-33%) report zero deployment error than ISM3 and CEA-VFA (about 43-47%). Also the spread in error CDF is more for ISM6 than either ISM3 and CEA-VFA. Comparing the same movement algorithm for different initial deployments, we observe that island deployment always results in lesser movement.

Topology distributions achieved by different algorithms are shown in Figures 5.8, 5.9 and 5.10 (we only illustrate the 100 static-30 mobile nodes scenario).

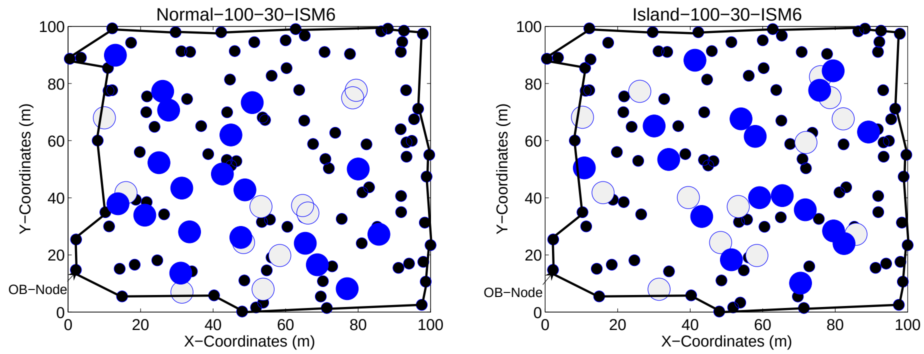

Figure 5.10: Topology distribution achieved with ISM6.

Small filled circles represent static nodes, B-nodes are shown by connected virtual boundary. The mobile nodes are initially introduced from the bottom-left corner of topology in normal distribution while in island distribution, mobile nodes are introduced from bottom-left and top-right corners of the topology. Light grey filled big circles represent the mobile nodes that have been utilized for coverage improvement while dark filled big circles are the final position of redundant mobile nodes. CEA-VFA results in a more uniform distribution of the mobile nodes (lesser spread in error CDF shown in Figure 5.7) than all other movement algorithms. ISM6 results in poor node distribution for both normal and island initial deployments.

Simulation results show that ISM variants save a considerable amount of energy by moving the mobile nodes lesser distances than the CEA-VFA for different types of initial deployment. However, this saving is at the cost of slight non-uniformity in the node distribution. Performance of ISM3 is comparable to ISM4 and ISM6 in terms of energy consumption and yet it achieves topology distribution closer to that of CEA-VFA with similar error CDF. To summarize, the simulation results show that ISM3 is a good compromise with significant savings in energy consumption.

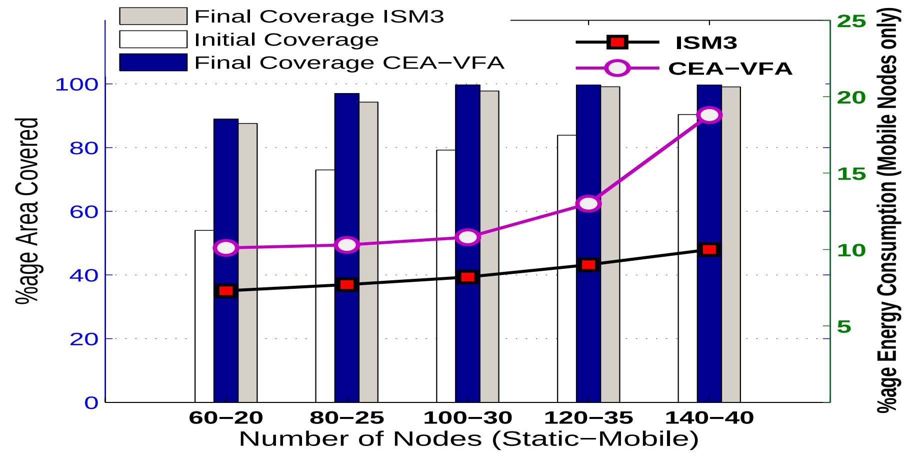

Figure 5.11: Percentage of area covered and energy consumed.

Finally, Figure 5.11 shows the initial and final percentage of area with sufficient coverage (shown by bars) for CEA-VFA and ISM3 with different topologies. Static nodes are randomly deployed in a 100m by 100m region. Static nodes estimate their coverage using PCA with required coverage probability (ρ_{reqd}) of 0.9 and request assistance from mobile nodes if they find uncovered perimeters. Mobile nodes spread out from an initial island distribution.

For 100-30, 120-35, and 140-40 static-mobile node cases, more than 99% of the area is covered after relocation of mobile nodes for both CEA-VFA and ISM3. For 80-25 configuration, coverage is enhanced from 72% to 94% after execution of 12 rounds of the ISM3 algorithm while the corresponding coverage in CEA-VFA is 96.7%. This gain in coverage is at the expense of energy consumption due to relocation of the mobile nodes. Figure 5.11 also shows that for 140-40 topology, ISM3 only consume about 10% of their total initial energy in the deployment phase as compared to close to 19% for CEA-VFA (shown by lines). Also note that for 140 static nodes random deployment in 100m x 100m area, on average the initial area coverage is about 90%. Mobile nodes assist in plugging the coverage holes that exist even in dense random deployments.

5.5 Summary

This chapter has detailed phase-II of the deployment stage of the MAPC protocol. This phase utilizes the mobility capabilities of mobile sensor nodes in a hybrid network to locate and plug the coverage holes in the topology. The mobile nodes spread out from an initial concentrated position to increase the existing coverage provided by the static sensor nodes. The mobile nodes that are still available after plugging the coverage holes are uniformly spread in the topology to help in coverage repair and maintenance during the operational lifetime of the WSN. We have proposed energy efficient virtual force based algorithms for managing the relocation of the mobile sensors nodes. We demonstrated that, for different types of initial deployment, our proposed algorithms consume only 30-40% of the energy consumed by the basic VFA. We also formulated our assignment problem as Integer Linear Program to arrive at idealistic optimal solutions that form basis of our performance comparison.

Chapter 6

Coverage Maintenance in Wireless Sensor Networks

6.1 Introduction

In Chapter 3 through 5, we have discussed algorithms that handle the coverage estimation and enhancement during the deployment phase of a wireless sensor network. This chapter focuses on coverage maintenance that is performed during the operational phase of the WSN.

Coverage loss is likely to happen due to sensor node failure during the operation of the WSN. Sensor nodes may fail due to various reasons. Nodes can malfunction, die due to physical damage, or become useless after depleting their energy resources. This can lead to several undesirable circumstances. Coverage holes can appear in the topology that degrade the quality of coverage offered by the deployed network. Node failure also introduces topology changes that in extreme cases results in network partitions. Failure of a few nodes can thus render the whole network useless if the remaining nodes cannot meet the requested application requirements. This loss should be repaired, through the coverage maintenance phase, to increase the network lifetime.

Coverage maintenance can be classified in two distinct categories; Proactive and reactive maintenance. Proactive coverage maintenance is performed before the actual loss in coverage takes place. This is typically undertaken to replace low energy nodes that have become energy constrained before other nodes in the topology. This non-uniform resource depletion usually happens due to varying traffic characteristics. Different WSN applications generate typical traffic patterns. Depending on the traffic pattern, some nodes may deplete their energy at a much faster rate than other nodes. It may be due to the edge effect, where nodes form favorable routes for increased data flow owning to reoccurring events in some locations. Similarly, nodes closer to a base station/sink consume more energy than nodes lying far away from it as they have to carry more transient traffic. Proactive maintenance thus aims at replacing such low energy nodes before they deplete all of their available energy resource.

Reactive coverage maintenance is performed once a coverage loss has already occurred in the deployed topology. This coverage loss can be due to dead or faulty nodes that have become useless without giving a warning. Note that reactive maintenance requires a different approach than the proactive maintenance. This is because in proactive maintenance, the node to be replaced is still alive and can initiate and coordinate the maintenance process, while in reactive maintenance some other node has to manage the coverage recovery process.

Given that manual replacement of either the batteries or the node itself is not an option for harsh inaccessible environments or large scale WSN, redundant mobile nodes in a hybrid network are best utilized for coverage repair. Although recharging using mobile nodes (energy harvesting) has been discussed in the literature [91], this thesis concentrates on relocating the redundant mobile nodes for replacing either low energy or dead nodes for coverage repair. However, the energy consumption for movement itself is a costly operation. Relocation thus results in lower total available energy in the network but with a more useful distribution.

Our aim is to minimize this relocation cost to improve the system utility.

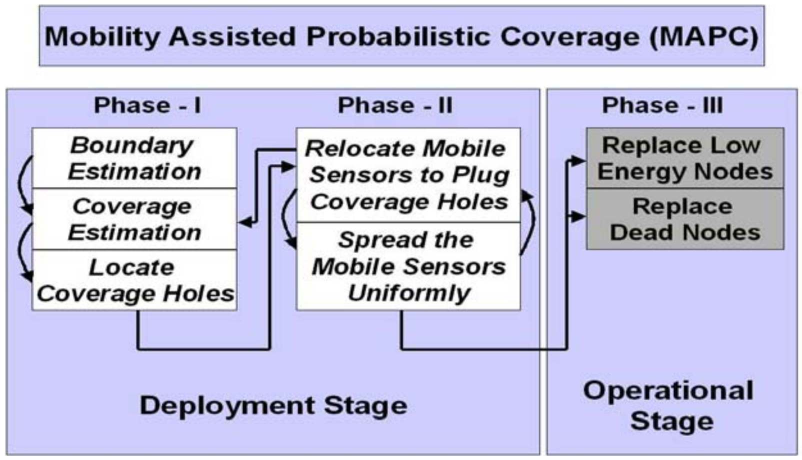

Figure 6.1: Building blocks of MAPC protocol.

In this chapter we discuss both proactive and reactive coverage maintenance and propose algorithms for node replacement. Figure 6.1 illustrates that the coverage maintenance forms phase-III of the MAPC protocol and is performed during the operational lifetime of the WSN. The remainder of this chapter is organized as follows. Section 6.2 elaborates the proactive maintenance. Reactive maintenance is discussed in Section 6.3. Section 6.4 details the simulation results and Section 6.5 summarizes the chapter.

6.2 Proactive Maintenance: Replacing Low Energy Nodes

Sensor nodes consume energy during the operation of the WSN. This energy consumption is not uniform e.g., nodes handling more data traffic consume more energy than other nodes. This non-uniform energy consumption can lead to an early loss of the energy constrained node due to battery depletion.

6.2.1 Problem Formulation

A replacement schedule should be able to deal with both iterative single node replacement and cumulative replacement of a set of low energy nodes. Thus the replacement problem can be formulated as following:

Let S_r represent the set of p redundant mobile nodes, $S_r = \{S_{r1}, S_{r2}, \ldots, S_{rp}\}$ and set E_r represent their remaining energies, $E_r = \{E_{r1}, E_{r2}, \ldots, E_{rp}\}$. There are q low energy sensors in set S_l $(q \leq p)$, $S_l = \{S_{l1}, S_{l2}, \ldots, S_{lq}\}$. We want to optimally assign the available redundant mobile sensors to replace the low energy nodes. This is a classical assignment optimization problem (also discussed in Chapter 5 Section 5.4.1). If a mobile node i is assigned to replace a low energy node, j, there is a cost (energy consumption due to movement) of c_{ij} and benefit (energy introduced in the network i.e., remaining energies of mobile nodes after the movement) of e_{ij}. We can either minimize the total value of assignment (minimize energy consumption) or maximize the total value (maximize energy introduced in the network).

The objective function can either be defined as:

"What is a replacement/assignment schedule in order to minimize energy consumption?"

or

"What is a replacement/assignment schedule in order to maximize energy introduced in the network?"

There are additional constraints that each mobile node can be assigned to replace exactly one low energy node and each low energy node must have one assigned mobile node. The assignment problem can be formulated as an Integer Linear Program (ILP) as follows;

$$Minimize \sum_{j=1}^{q} \sum_{i=1}^{p} c_{ij} x_{ij}, or Maximize \sum_{j=1}^{q} \sum_{i=1}^{p} e_{ij} x_{ij}$$

Constraints:

- $x_{ij} \leq 1; i = 1, \ldots, p; j = 1, \ldots, q$

- $\sum x_{ij} \leq 1; j = 1, \ldots, q$

- $\sum x_{ij} \leq 1; i = 1, \ldots, p$

Note that these optimized solutions using ILP are centralized and hence can not be used in a large scale WSN. They serve as benchmarks to compare the performance of our proposed distributed heuristic algorithms.

6.2.2 Heuristic Solutions

The coverage maintenance process consists of two distinct tasks. (i) How to decide whether a mobile node relocation is required or not? (ii) If relocation is required, how to locate a suitable redundant mobile node for relocation? Assuming that every node keeps track of its current energy level, task 1 basically evaluates whether the death of a low energy node has any effect on the overall coverage of the area. Once it is ascertained that the loss of a low power node will result in reduced coverage, task 2 aims to locate redundant mobile nodes and manage the relocation. Having this overall picture in place, we discuss distributed heuristic solutions to the coverage maintenance problem.

Virtual force-based algorithms described in Chapter 5 uniformly deploy the redundant mobile nodes in the topology. Once the deployment phase is over, mobile nodes announce their final positions in the topology to register with their static neighbors. Static nodes mark these mobile nodes as their one-hop neighbors. Each static node thus maintains a mobile node neighbor list (referred as MNN list).

A low energy node may not always require a replacement. This is possible where random deployment of static nodes has resulted in redundant nodes that can be turned off without affecting the area coverage e.g., nodes in a cluster. In such a case, the role of the low energy node can easily be taken by one of its neighbors with no effect on the area coverage. A static node whose remaining energy falls below a predefined threshold, E_{repl}, first checks whether its loss will result in any change in area coverage or not. E_{repl} threshold ensures that the low energy node has sufficient remaining energy that enable it to participate in the replacement procedure. For this purpose, it broadcasts a **Low** energy message. The neighbors on receiving this Low energy message recalculate their perimeter coverage by excluding the sender of the Low energy message. A neighbor that finds its perimeter coverage is insufficient after removing the contributions of the low energy node, broadcast a **Replace** message. Other neighbors with insufficient perimeter coverage suppress their broadcast of Replace message on receiving any Replace message. A Replace message indicates that the low energy node should go ahead with the replacement phase as its loss will result in coverage degradation. A low energy node that does not receive any Replace message in response to a Low energy message, eventually times-out. This establishes that the node is a redundant node and its death would not effect the area coverage.

A low energy node that has received a Replace message from one of its neighbor, initiates the mobile node discovery process. Depending on the location of the low energy node relative to the location of a redundant mobile node, there are three possible scenarios that we discuss here.

6.2.2.1 Case No 1

A low power node has one-hop mobile nodes in the MNN list.

As the deployment phase distributes the additional mobile nodes uniformally in the area, there is a high probability of finding a mobile node in the neighborhood of a low energy node. If a low energy node has one-hop mobile nodes in its MNN list, the low energy node broadcasts a **Help** message. The Help message generated has the location of the low energy node as the deployment coordinates. This Help message is received by one or more mobile nodes.

If a mobile node receives Help message from only a single static node, the relocation decision is trivial. Mobile node sends an **Avail** message to the originator of the Help message, showing its willingness to move to the desired location. This Avail message contains mobile node ID, location and its remaining energy Er. In general, a mobile node can receive Help messages from n static sensor nodes and a static node can receive Avail messages from m willing mobile sensor nodes. The task is to plan energy efficient movement of the mobile nodes by resolving contention. For multiple Help messages, the mobile node calculates the distance from its current location to the requested deployment points. It then selects the nearest requested deployment point that involves the least movement. The mobile node then sends an Avail message to the static node that has originated the corresponding Help message.

Willingness from multiple mobile nodes is then resolved at the static node that generated the Help message. For this purpose, we propose two distributed, heuristic solutions namely **Heuristic-Minimize-Energy consumption (Hr-Min-E)** and **Heuristic-Maximize-Energy-Introduced (Hr-Max-E-I)**. These solutions differ in how a low power node selects a willing mobile node for replacement. Hr-Min-E tries to select a mobile node that is located closest to the low energy node (to minimize the energy consumption associated with movement) while Hr-Max-E-I selects a mobile node for replacement that has the maximum remaining energy after the movement.

A low energy node may receive multiple Avail messages from different mobile nodes. The low energy static node calculates the distance, d_j, from the requested deployment point to each of the willing mobile nodes to check whether the replacement can be performed within the remaining lifetime of the low energy node. The node uses either Hr-Min-E or Hr-Max-E-I to select a willing mobile node. For Hr-Min-E, the node compares the distances (d_j) and selects the nearest located mobile node. For Hr-Max-E-I, it uses the Er values in the received Avail messages to calculate the remaining energies after the movement (energy consumed to travel a certain distance can be easily calculated) and selects the mobile node with highest remaining energy after the movement. It then unicasts a **Move** message to the selected mobile node.

The mobile node on receiving the Move message broadcasts an **Update** message and starts moving towards the deployment point. The Update message contains the senders list of neighboring mobile nodes. Mobile nodes that receive this Update message removes the sender mobile node from their neighbor list. Static nodes update their MNN list by removing the sender and adding any unknown mobile node (and marking these as multi-hop neighbors) from the Update message.

Mobile nodes that do not receive a Move message in response to their Avail messages time out and select the next candidate static node from the Help message list to send an Avail message. This is repeated for each entry in the list until a response in the form of a Move message is received. If no Move message is received, the mobile node waits for the arrival of new Help messages. This check covers static nodes that may not have received any Avail messages due to mobile nodes initially selecting other preferable choices.

A low energy static node that has initiated a Help message should receive either one or more Avail messages, or Update messages from all of its one-hop mobile node neighbors. For each Update message that is received, the low energy

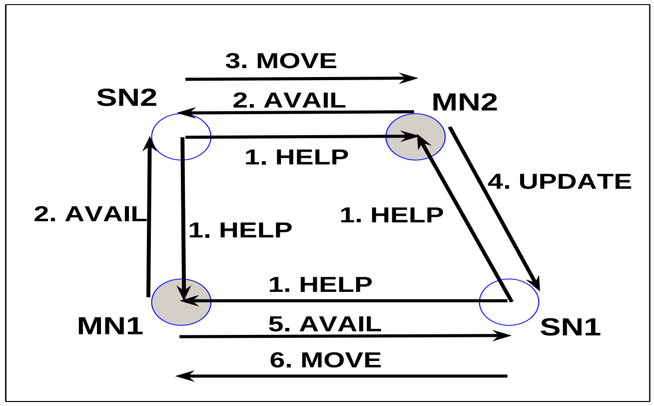

Figure 6.2: Mobile node discovery process.

node continues to build its MNN list with information about multi-hop mobile nodes. If Update messages have been received from all one-hop mobile node neighbors, the low energy node now probes the multi-hop mobile node neighbors in the MNN list. This processing is discussed as case No 2. Figure 6.2 illustrates the mobile node discovery phase of the coverage maintenance process.

6.2.2.2 Case No 2

A low power node has only multi-hop mobile nodes in the MNN list.

The low energy node unicasts the Help message to each known multi-hop mobile neighbors. This message is delivered, over multi-hop, to the mobile nodes. The low power node also starts a wait timer. If an Avail message is received from one of these mobile nodes, the wait timer is reset. On expiry of the wait timer, if Avail messages have been received, the low energy node applies the heuristic solutions (Hr-Min-E or Hr-Max-E-I) to invite a mobile node for replacement. In case no Avail message has been received from known multi-hop mobile nodes, the node follows case No 3.

Note that Update messages are only single-hop broadcasts and thus a low energy node probing a multi-hop mobile node will not receive the Update messages. The wait timer is necessary to safeguard against existence of stale information in the MNN list, whereby the multi-hop mobile nodes have already relocated.

6.2.2.3 Case No 3

A low power node has an empty MNN list.

The low energy node broadcasts a **Locate** message (with scope = 1, for one-hop neighbors only). Neighbors with populated MNN list reply to the Locate message by sending their MNN list to the low power node. The low power node builds its own MNN list based on the replies and then unicasts the Help message to the discovered mobile nodes. If no replies are received, the Locate message is sent again with an incremented scope value. Maximum value for the scope counter can be set based on the required reaction time/extent of the area.

The mobile node discovery mechanism described is similar to the expanding ring search [92] [93] but is more efficient in terms of number of broadcasts. Specifically, the update messages enables the discovery of mobile nodes located up to two hops away from the static nodes that are being probed. Let M denotes the number of mobile nodes and N represents total number of static nodes in the topology ($M << N$). Also suppose that l represents the maximum scope value that will cover all the static nodes N in the topology. The additional overhead of M node broadcasting Update messages on movement, results in saving of n_l node broadcasts, where n_l is the number of static nodes between scope value $l-1$ and l.

6.3 Reactive Maintenance: Replacing Dead / Faulty Nodes

The replacement mechanism described in Section 6.2 does not work for coverage loss due to dead sensors because it requires active participation from the node being replaced. Failure of a sensor node can be detected by its neighbors (loss of protocol messages/link layer Acks etc). These neighbors initiate the coverage calculation phase (PCA, Chapter 4, Section 4.3) to check whether the dead neighbor sensor has resulted in any decreased coverage. In case the node failure does not result in any coverage loss, no further action is required.

Neighbors that find that loss of the node has resulted in decreased coverage start a random delay timer. Upon expiry of this random timer, a node broadcasts a **Proxy** message to announce that the sender will act as proxy to manage the coverage maintenance process. Other nodes still in their random wait period, cancel their timers. The node acting as the proxy, now follows the processing discussed in Section 6.2 to manage the mobile node relocation. The Help messages in this case contain the location coordinates of the failed node as the requested deployment point.

For reactive maintenance, we have so far made two simplifying assumptions. First, the death of every node is detected by at least one neighbor of the dead node. Second that the Proxy message broadcast is received by all neighbors of the dead node. We now show how we can relax these assumptions. There could be scenarios in which a group of nodes are destroyed en-mass causing some nodes to die along with all of their one-hop neighbors. In such a case, the assumption that the loss of node is always detected by a neighbor does not hold true. Lets look at an example shown in Figure 6.3. All nodes inside the circle have been destroyed. As a result, node marked X has lost all of its one hop neighbors and

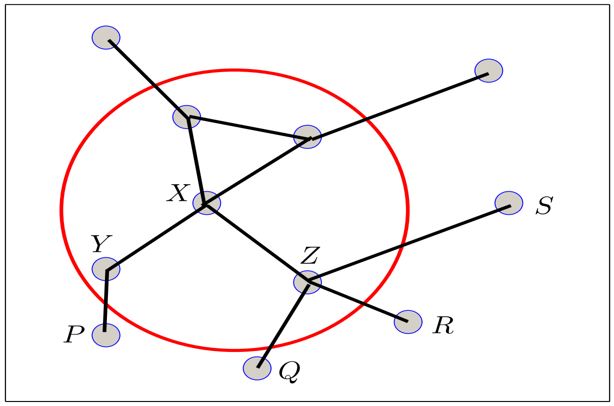

Figure 6.3: Special case of reactive maintenance.

thus there is no neighbor currently available that could act as a proxy to arrange the node replacement. We note that for an initial connected network topology, at least one neighboring node is able to detect the loss of one of nodes in the group that has died. We can thus use recursive node replacement to detect and recover the loss of coverage due to the multiple nodes failure. A mobile node, on reaching its new location, should first establish neighborhood and apply the PCA (phase-I of MAPC) to confirm that its coverage perimeter is fully covered e.g., a mobile node replacing node Y in Figure 6.3, would detect an uncovered perimeter and can request additional help to fully cover the area. Note that a mobile node replacing a B-node does not perform this additional coverage check.

A Proxy message is broadcast by a willing neighbor of the dead node to prevent other neighbors from attempting the node maintenance for the same dead node. This Proxy message may not be received by all of the one-hop neighbors of the dead node. Figure 6.3 shows such a scenario. Nodes Q, R, and S are one hop neighbors of node Z that have died. All three nodes detect the loss of node Z. Suppose Q broadcast a Proxy message. This proxy message is only received by R that cancels its Proxy message. In the meanwhile node S timeouts and also broadcast a Proxy message. Since Q and S cannot communicate with each other (hidden terminals), both would end up performing the coverage maintenance to replace dead node Z. To avoid such a situation, neighbor nodes overhearing the Proxy announcement can resolve the contention. Node R, for example, can stop S from assuming the proxy role as it has already received a Proxy message from node Q. Note that despite this duplicate proxy message check, in extreme cases, two or more nodes can end up performing the coverage maintenance for the same dead node. This can happen due to loss of a node that is the only link between two or more otherwise disconnected regions.

As the deployed nodes have already gone through the coverage enhancement phase during deployment (phase I and II of MAPC protocol), occurrence of this special topology is extremely rare. On the other hand, if sufficient redundant mobile nodes are present in the topology, moving multiple mobile nodes to replace such a bottleneck node can in fact increase the network performance in terms of throughput and reliability.

6.4 Simulation Results

We perform complete simulation of all the three phases of the MAPC protocol with 100 static-30 mobile nodes topology. Mobile nodes use ISM3 from an initial island deployment to enhance the coverage and after the deployment stage, 16 redundant mobile nodes are available at different locations in the topology (Figure 5.9). For proactive maintenance, we have ten low energy static nodes (shown as small filled circles in Figure 6.4 and 6.5) lying along high activity data paths to sink. These low energy nodes go through coverage loss check and then initiate the Help messages.

Figures 6.4 and 6.5 show the movement schedules for the heuristic and the centralized optimized solutions while Figure 6.6 shows the quantitative perfor-

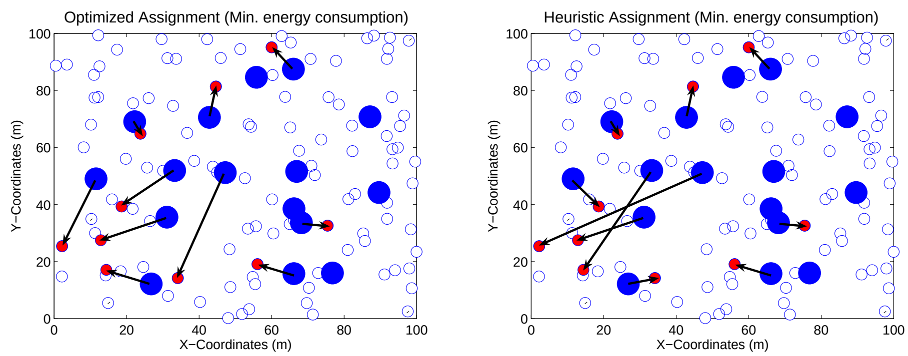

Figure 6.4: Proactive maintenance: Minimize energy consumption.

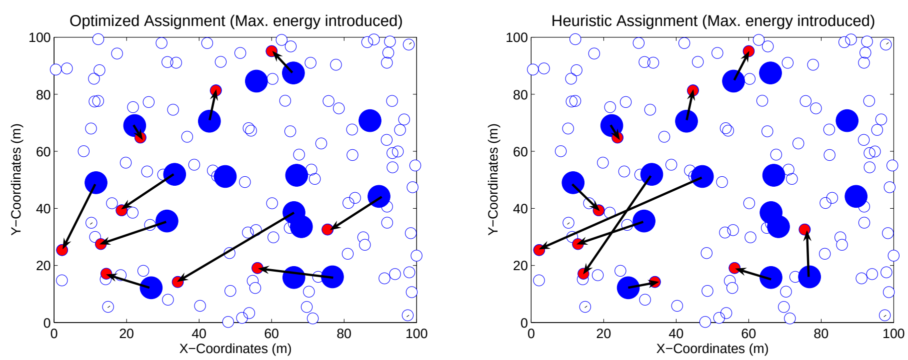

Figure 6.5: Proactive maintenance: Maximize energy introduced.

mance comparison. The optimized assignment with minimize energy consumption (referred as Opt-Min-E) resulted in the least consumption of energy (1322J) due to movements while Hr-Min-E consumed 1433J of energy for the replacement schedule. Optimized assignment with maximize energy introduced (referred as Opt-Max-E-I) resulted in introduction of highest total energy (41270J) with distributed Hr-Max-E-I ranked second best by giving a replacement schedule that introduces 41160J of energy in the network. The energy consumed for movement in Opt-Max-E-I and Hr-Max-E-I are almost similar.

For reactive maintenance, we assume that all nodes in a circle of radius 12.5 m are destroyed resulting in the creation of a void shown in Figure 6.8 and 6.9. Note that any redundant mobile node located inside the circle is also assumed to

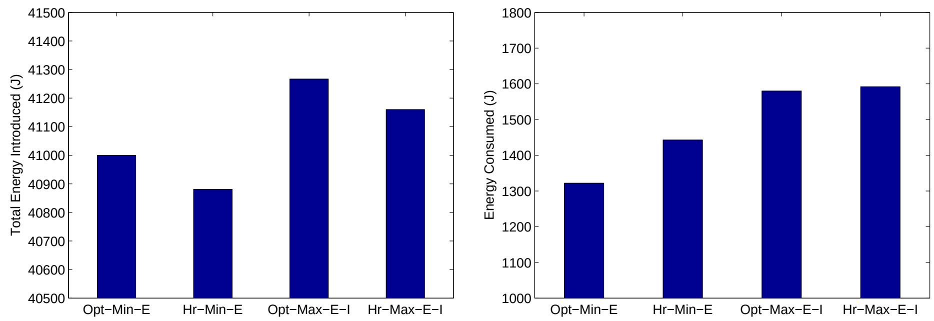

Figure 6.6: Proactive maintenance: Performance of assignment algorithms.

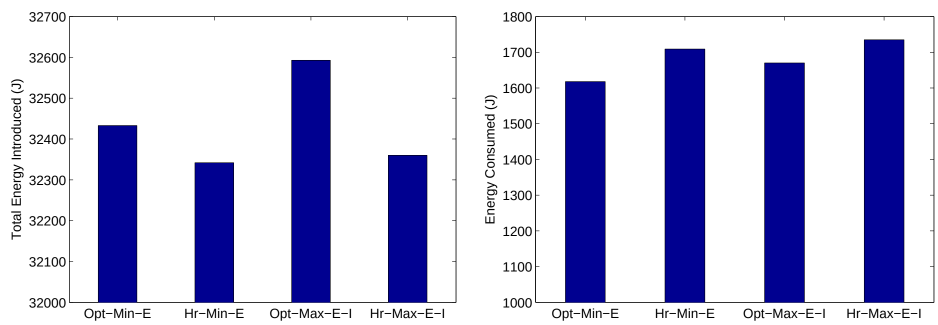

Figure 6.7: Reactive maintenance: Performance of assignment algorithms.

be destroyed. In the shown topology, eight static nodes get destroyed in all and neighbors of these dead nodes detect this loss and initiate the node replacement process.

Figure 6.7 compares the performance of the algorithms in terms of energy consumption and total energy introduced in the network while Figures 6.8 and 6.9 show the movement schedules for the heuristic and the centralized optimized solutions. Similar to the simulation results for proactive maintenance, the Opt-Min-E resulted in the least consumption of energy for the replacement schedule. Opt-Max-E-I resulted in provision of highest total energy (32593J with introduction of eight mobile nodes) while distributed Hr-Max-E-I gave a replacement schedule

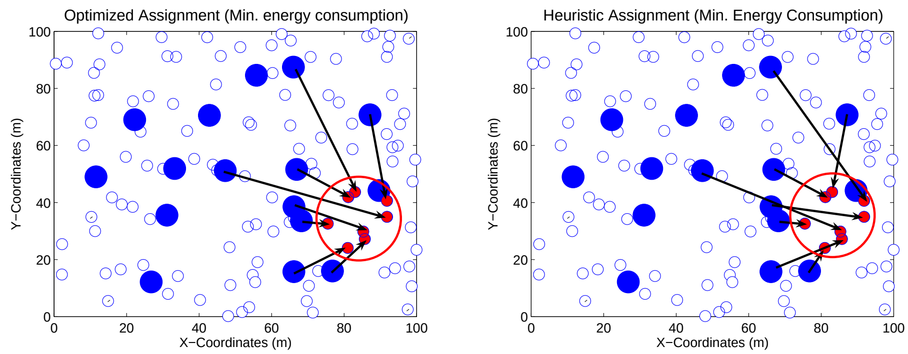

Figure 6.8: Reactive maintenance: Minimize energy consumption.

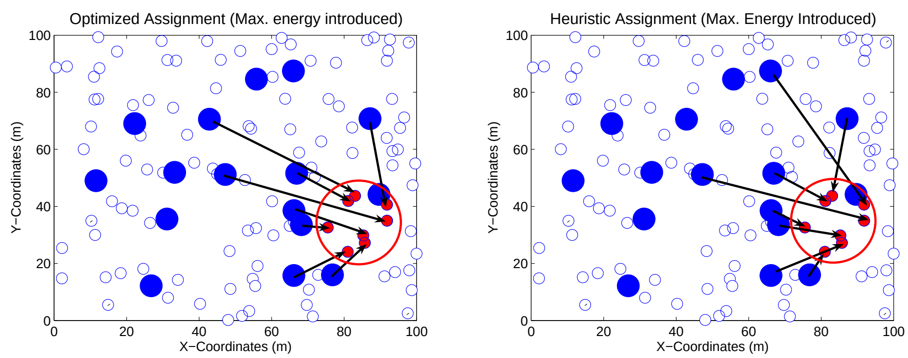

Figure 6.9: Reactive maintenance: Maximize energy introduced.

that introduces 32360J of energy in the network. Energy consumption for node replacement by using Hr-Max-E-I is the highest among all the four replacement algorithms.

The results show that performance of these distributed, heuristic solutions is comparable with the centralized optimal solutions. The small performance degradation is due to the localized and distributed nature of the heuristic solutions.

6.5 Summary

This chapter describes the coverage maintenance procedure. Mobile nodes are utilized to replace the dead/low energy nodes to the coverage. This not only results in maintaining the acceptable level of coverage but also prolongs the network life time. Both proactive and reactive coverage maintenance has been discussed. We demonstrated that our proposed distributed, heuristic solutions for coverage maintenance perform close to the centralized optimal solutions.

Chapter 7

Conclusion and Future Research Directions

7.1 Conclusion

This thesis has proposed a comprehensive three phase protocol, MAPC, for ensuring area coverage employing a hybrid sensor network. We argue that a hybrid sensor network is a cost-effective solution, as compared to an all mobile network, that provides the necessary re-configuration capability that is vital for self-organization, coverage enhancement, and repair of a wireless sensor network. MAPC protocol utilizes mobile sensor nodes to ensure that adequate area coverage is maintained from deployment through the lifetime of a WSN. To our knowledge, no such comprehensive solution has been presented in the literature that addresses the deployment, coverage enhancement, and coverage maintenance issues for a hybrid sensor network in its entirety.

The primary contribution of this thesis is a pragmatic approach to sensor coverage and maintenance that we hope would lower the technical barriers to its field deployment. A distinguishing feature is that most of the assumptions made in the MAPC protocol are realistic and implementable in real-life applications, e.g., practical boundary estimation, coverage calculations based on realistic sensing model, and use of movement triggering thresholds based on real radio characteristics etc.

First, this thesis proposed a practical boundary estimation scheme with the objective to identify the nodes lying on the outer boundary of a deployed network topology. The boundary estimation results in identification of the virtual boundary of the topology. The scheme is distributed and is executed only by a subset of all deployed nodes. An extension to the basic boundary estimation scheme is also presented that enables the availability of coarse boundary information at several landmarks, called Posts, on the outer boundary of the deployed network. The virtual boundary information is utilized by static nodes during coverage estimation and mobile nodes during the coverage enhancement phase of the MAPC protocol. The performance of the boundary estimation scheme has been studied both through discrete event simulations and conducting experiments on real motes.

Second, this thesis proposed a novel coverage estimation algorithm, PCA, that evaluates the area coverage in WSN. PCA takes into account the realistic variations in sensing behavior associated with range based sensors and evaluates probability of successful sensing at discrete intervals around a sensor location. The algorithm takes advantage of the sensing contributions from neighbors for accurate estimation of a node's area coverage. The performance of the PCA has been investigated by simulation and experiments. Results show that PCA can accurately estimate the area coverage in a region.

Next, a set of integrated deployment and coverage enhancement algorithms have been proposed that are based on virtual force algorithm. The algorithms use movement triggering thresholds that are based on real radio characteristics. Our evaluations show that, for different types of initial deployments, the proposed movement algorithms result in substantial energy savings, 60% to 70% as compared to basic virtual force-based algorithms, a major challenge for the success of a WSN. The deployment problem has been formulated as an Integer Linear Program to arrive at idealistic optimal solutions that form the basis of our performance comparison.

Finally, this thesis addressed the problem of coverage maintenance in WSN by considering coverage loss due to damaged and energy depleted nodes. The coverage maintenance problem has been classified into two distinct categories: Proactive and reactive maintenance. The coverage maintenance problem has also been formulated as an Integer Linear Program and implementable heuristics are developed that perform close to optimal solutions.

7.2 Future Research Directions

In this section, several future research extensions are identified for further improvements in the proposed MAPC protocol.

7.2.1 Relaxing Simplifying Assumptions

Although most of the assumptions made in this thesis are realistic, a few simplifying assumptions do exist e.g., availability of precise localization information, a flat mobility terrain and obstacle free environment etc. Imperfect localization and presence of obstacles will effect the boundary and coverage estimation, movement algorithms, and coverage maintenance. A future work could study the effect of relaxing these assumptions on the performance of the MAPC protocol.

7.2.2 In-situ Calibration

For coverage estimation, we assumed mean values of path loss component, n , and the shadowing deviation, σ, for all the sensors in the region. In real deployment scenarios n and σ varies spatially as well as temporally due to changing environments. The consideration of different n and σ for different sub-regions in the sensor network can capture the spatial variations for an even more realistic sensing behavior. We believe that an in-situ (on site or in a field of similar environmental characteristic if the site is in-accessible) mechanism to calibrate the sensing behavior for each sensor is needed. This can help estimate close-to realistic values of parameters that are critical for the performance of the probabilistic coverage estimation.

7.2.3 Extended Experimental Studies

We believe simulations alone cannot capture the behavior of WSN protocols in real deployments unless validated by field experiments. We have conducted experiments with real sensor hardware for boundary estimation as well as the probabilistic coverage algorithm. Our experimental results show that undesirable radio characteristics have a significant impact on perfectly assumed inter-node communications. The experimental work can be extended by using mobile robots such as Robomote [89] to evaluate the mobile nodes deployment, coverage enhancement, and coverage maintenance phases of the MAPC protocol.

7.2.4 Topology Control

This thesis addresses the area coverage problem with the aim to ensure adequate coverage in a randomly deployed WSN. Areas with low coverage are identified and coverage is enhanced by relocating redundant mobile nodes to strategic locations. On the other hand, a random deployment can also result in areas of high node concentration, for example cluster. MAPC protocol can be supplemented by incorporating a suitable topology control protocol (discussed in Chapter 2 Section 2.3) that can schedule wake/sleep cycles for nodes in areas of high concentration. This can increase the system utility, and extend the overall network lifetime.

7.2.5 Multiple k Coverage

The work presented in this thesis can be extended to incorporate the multiple coverage constraint where every location in the target area is required to be covered by k sensors. This is useful in many sensor networks applications for fault tolerance/redundancy, for accurate target localization using triangulation-based positioning protocols [94], and trilateration based localization [95] etc. The multiple probabilistic coverage constraint can thus be specified as (ρ_{reqd}, k), where ρ_{reqd} is the required detection probability and k is the degree of coverage.

7.3 Summary

This chapter has summarized the work presented in this thesis. Several future research directions have also been identified for improvement in the work presented in this thesis.

Bibliography

[1] Ian F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, pages 102–114, 2002.

[2] Loren Schwiebert, Sandeep K.S. Gupta, and Jennifer Weinmann. Research challenges in wireless networks of biomedical sensors. In ACM MobiCom'01, pages 151–165, July 2001.

[3] CodeBlue: Wireless Sensor Network for Medical Care. http://www.eecs.harvard.edu/ mdw/proj/codeblue.

[4] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. Computer Networks: The International Journal of Computer and Telecommunications Networking, 45 (5):605–634, 2004.

[5] A. Ledeczi, A. Nadas, P. Vogyesi, G. Balagh, B. Kusy, J. Sallai, G. Pap, S. Dora, K. Molnar, M. Maroti, and G. Simon. Countersniper system for urban warfare. ACM Transactions on Sensor networks (TOSN), 1 (2):153–177, 2004.

[6] Tian He, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh. An energy efficient surveillance system using wireless sensor networks. In IEEE/ACM Mobile Systems, Applications and Services (MobiSys'04), pages 270–283, June 2004.

[7] Alan Mainwaing, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In Proceedings of the Workshop on Wireless Sensor Networks and Applications (WSNA '02), pages 88–97, 2002.

[8] Habitat Monitoring on James Reserve. http://www.jamesreserve.edu.

[9] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. Journal of Internet Computing, 10 (2):18–25, 2006.

[10] Razvan Musaloiu-E, Andreas Terzis, Katalin Szlavecz, Alex Szalay, Joshua Cogan, and Jim Gray. Life under your feet: A wireless soil ecology sensor network. In Workshop on Embedded Networked Sensors (EMNETS' 06), 2006.

[11] Jeongyeup Paek, Krishna Chintalapudi, Ramesh Govindan, John Caffrey, and Sami Masri. A wireless sensor network for structural health monitoring: Performance and experiences. In Workshop on Embedded Networked Sensors (EMNETS' 05), pages 1–10, May 2005.

[12] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steve Glaser, and Martin Turon. Wireless sensor networks for structural health monitoring. In Proceedings of the ACM Embedded Networked Sensor System (SenSys '06), pages 427–428, 2006.

[13] Pei Zhang, Christopher M. Saddler, Stephen A. Lyon, and Margaret Martonosi. Hardware design experiences in zebranet. In Proceedings of the ACM Embedded Networked Sensor System (SenSys '06), pages 227–238, 2006.

[14] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley. Wireless ad-hoc sensor and actuator network on the farm. In Proceedings of the International Conference on Informatipn Processing in Sensor Networks (IPSN'06), pages 492–499, April 2006.

[15] Mani Srivastava, Richard Muntz, and Miodrag Potkonjak. Smart kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments. In ACM MobiCom'01, pages 132–138, July 2001.

[16] Chee-Yee Chong and Srikanta P. Kumar. Sensor networks:Evolution, opportunities, and challenges. In Proceedings of the IEEE, Volume 91 No 8, pages 1247–1256, August 2003.

[17] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In IEEE/ACM MobiCom'99, pages 263–270, August 1999.

[18] Ian F. Akyildiz and Eric P. Stuntebeck. Wireless underground sensor networks: Research challenges. Ad-hoc Networks, 4:669–686, 2006.

[19] Crossbow. http://www.xbow.com/.

[20] Tmotes. http://www.moteiv.com/.

[21] Robert Nowak and Urbashi Mitra. Boundary estimation in sensor networks: Theory and methods. In Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'03), April 2003.

[22] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. Inform. Process. Lett., 2(No 1):18–21, March 1973.

[23] Ivan Stojmenovic and Pedro Eduardo Villaneuva Pena. A scalable quorum based location update scheme for routing in ad hoc wireless networks. Technical Report TR-99-09, SITE, University of Ottawa, 1999.

[24] Dandan Liu, Ivan Stojmenovic, and Xiaohua Jia. A scalable quorum based location service in ad hoc and sensor networks. In 3rd IEEE Mobile Ad Hoc and Sensor Systems (MASS '06), Oct 2006.

[25] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks, 7:609–616, 2001.

[26] Qing Fang, Jie Gao, and Leonidas J. Guibas. Locating and bypassing routing holes in sensor networks. In IEEE INFOCOM 2004, June 2004.

[27] K.R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. Systematic Zoology, 18:259–278, 1969.

[28] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In Proceedings of the ACM/IEEE MobiCom '2000, pages 243–254, 2000.

[29] G. Toussaint. The relative neighborhood graph of a finite planar set. Pattern Recognition 12, 4:261–268, 1980.

[30] S.P. Fekete, A. Kroller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood based topology recognition in sensor networks. In Proceedings of the First workshop on Algorithms for Sensor Networks (Algosensors'04), LNCS 3121, pages 123–136, 2004.

[31] S. P. Fekete, M.Kaufmann, A. Kroller, and K.Lehmann. A new approach to boundary recognition in geometric sensor networks. In Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05), pages 82–85, 2005.

[32] Alexander Kroller, Sander P. Fekete, Dennis Pfisterer, and Stefan Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms(SODA'06), pages 1000–1009, 2006.

[33] S. Funke. Topological hole detection in wireless sensor networks and its application. In Joint Workshop on Foundations of Mobile Computing, pages 44–53, 2005.

[34] Yue Wang, Jie Gao, and Joseph S.B. Mitchell. Boundary recognition in sensor networks by topological methods. In 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06), pages 122–133. ACM, Sep, 2006.

[35] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenkar. Geographic routing made practical. In 2nd Symposium on Networked Systems design and Implementation (NSDI'05), pages 217–230, May 2005.

[36] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In Proceedings of the IEEE INFOCOM, pages 1380–1387, 2001.

[37] Mihaela Cardei and Jie Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. Journal of Computer Communications, 29:413–420, 2006.

[38] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In Proceedings of the 2nd ACM WSNA'03, Sep 2003.

[39] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In Proceedings of the ACM Embedded Networked Sensor System (SenSys '03), pages 28–39, Nov 2003.

[40] B.Chen, K. Jamieson, H. Balakrshnan, and R. Morris. SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In Proceedings of the ACM/IEEE MobiCom '01, July 2001.

[41] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In Proceedings of the ACM SenSys '03, Nov 2003.

[42] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In International Conference on Communications (ICC'01), pages 472–476. IEEE, 2001.

[43] Mihaela Cardei, My T. Thai, Yingshu Li, and Weili Wu. Energy efficient target coverage in wireless sensor networks. In IEEE INFOCOM 2005, pages 1976–1984 Vol. 3, 2005.

[44] Honghai Zhang and Jennifer C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. Technical Report UIUDCS-R-2003-2351, Department of Computer Science, University of Illinois at Urbana Champaign, 2003.

[45] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In Proceedings of the first ACM WSNA'02, Sep 2002.

[46] Jie Jiang and Wenhua Dou. A coverage-preserving density control algorithm for wireless sensor networks. In ADHOC-NOW'04, pages 42–55. Springer-Verlag, July 2004.

[47] Ghrist R. and Muhammad A. Coverage and hole detection in sensor networks via homology. In Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'05), pages 254–260, 2005.

[48] S.P. Kuo, Y.C. Tseng, F.J Wu, and C.Y. Lin. A probabilistic signal-strength based evaluation methodology for sensor network deployment. International Journal of Ad-hoc and Ubiquitous Computing, 1 (1/2):3–12, 2005.

[49] Shansi Ren, Qun Li, Haining Wang, Xin Chen, and Xiaodong Zhung. A study on object tracking quality under probabilistic coverage in sensor networks. Mobile Computing and Communications Review, 9(Number 1), Jan 2005.

[50] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization in distributed sensor networks. ACM Transactions on Embedded Computing Systems, 3(1):61–91, 2004.

[51] Guoliang Xing, Chenyang Lu, Robert Pless, and Joseph A. O'Sullivan. Co-grid: An efficient coverage maintenance protocol for distributed sensor networks. In Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'04), pages 414–423, 2004.

[52] Santosh Kumar, Ten H. Lai, and Jozsef Balogh. On k-coverage in a mostly sleeping sensor network. In IEEE/ACM MobiCom'04, pages 144–158, Sep 2004.

[53] Guiling Wang, Guohong Cao, and Tom La Porta. Movement-assisted sensor deployment. In IEEE INFOCOM 2004, June 2004.

[54] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields:A distributed, scalable solution to the area coverage problem. In 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02), June 2002.

[55] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. Autonomous Robots,Special Issue on Intelligent Embedded Systems, 13(2):113–126, Sep 2002.

[56] S. Poduri and Gaurav S Sukhatme. Constrained coverage for mobile sensor networks. In IEEE International Conference on Robotics and Automation, pages 165–172, May 2004.

[57] Nojeong Heo and Pramod K. Varshney. An intelligent deployment and clustering algorithm for a distributed mobile sensor network. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, volume 5, pages 4576–4581, Oct 2003.

[58] Benyuan Liu, Peter Brass, Oliver Dousse, Philippe Nain, and Don Towsley. Mobility improves coverage of sensor networks. In ACM International

Symposium on Mobile Ad-hoc Networking and Computing MobiHoc '05, pages 300–308, 2005.

[59] Jie Wu and Shuhui Yang. Smart: A scan based deployment protocol. In IEEE INFOCOM 2005, pages 2313–2324 Vol. 4, 2005.

[60] Shuhui Yang, Jie Wu, and Fei Dai. Localized movement-assisted sensor deployment in wireless sensor networks. In IEEE Mobile Ad-hoc and Sensor Systems Conference MASS '06, pages 753–758, Oct 2006.

[61] Peter Corke, Steven Hrabar, Ronald Peterson, Daniela Rus, Srikanth Saripalli, and Gaurav Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In Proceedings of the IEEE 2004 International Conference on Robotics and Automation, Volume 4, pages 3602–3608, May 2004.

[62] Maxim A. Batalin and Gaurav S. Sukhtame. Coverage, exploration and deployment by a mobile robot and communication network. Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks, 26(2):181–196, 2004.

[63] Guiling Wang, Guohong Cao, and Tom La Porta. A bidding protocol for deploying mobile sensors. In 11th IEEE International Conference on Network Protocol ICNP '03, pages 315–324, Nov 2003.

[64] Guiling Wang, Guohong Cao, and Tom La Porta. Proxy based sensor deployment in mobile sensor networks. In Ist IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS-04), pages 493–502, Oct 2004.

[65] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks. In IEEE INFOCOM 2005, pages 2302–2312 Vol. 4, March 2005.

[66] Archana Sekhar, B.S. Manoj, and C. Siva Ram Murthy. Dynamic coverage maintenance algorithms for sensor networks with limited mobility. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), pages 51–60, March 2005.

[67] Saurabh Ganeriwal, Aman Kansal, and Mani B. Srivastava. Self aware actuation for fault repair in sensor networks. In Proceedings of the IEEE International conference on Robotics and Automation (ICRA '04), May 2004.

[68] Yongguo Mei, Changjiu Xian, Saumitra Das, Y. Charlie Hu, and Yung-Hsiang Lu. Replacing failed sensor nodes by mobile robots. In 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06), pages 87–92. IEEE, 2006.

[69] Xu Li and Nicola Santoro. An integrated self-deployment and coverage maintenance scheme for mobile sensor networks. In Conference on Mobile Sensor Networks (MSN'06), LNCS 4325, pages 847–860. Springer-Verlag, 2006.

[70] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. IEEE Personal communications magazine, 7(5):28–34, October 2000.

[71] Network Simulator 2. http://www.isi.edu/nsnam/ns/.

[72] Deepak Ganesan, Deborah Estrin, Alec Woo, and David Culler. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report UCLA/CSD-TR-02-0013, UCLA Computer Science Department, 2002.

[73] RF Monolithics. http://www.rfm.com/.

[74] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In Proceedings of the ACM Embedded Networked Sensor Systems (SenSys '03), pages 14–27, Nov 2003.

[75] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In Proceedings of the ACM Embedded Networked Sensor System (SenSys '03), pages 1–13, Nov 2003.

[76] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In IEEE/ACM Mobile Systems, Applications and Services (MobiSys'04), pages 125–138, June 2004.

[77] ChipCon. http://www.chipcon.com/.

[78] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In Third Workshop on Embedded Networked Sensors (EMNETS '06), May 2006.

[79] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. ACM Transactions on Sensor Networks (TOSN), 2(No 2):221–262, May 2006.

[80] Xueli An, Jing Wang, R. Venkatesha Prasad, and I.G.M.M Niemegeers. Opt-online personal tracking system for context-awareness in wireless personal networks. In Proceedings of the ACM REALMAN '06, pages 47–54, May 2006.

[81] Matthew M. Holland, Ryan G. Aures, and Wendi B. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh '06), pages 140–150, 2006.

[82] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In Proceedings of the 4th Int Conference IPSN/SPOTS '05, 2005.

[83] Nirupama Bulusu, John Heidemann, Deborah Estrin, and Tommy Tran. Self-configuring localization systems: Design and experimental evaluation. ACM Transactions on Embedded Computing Systems, 3(1):24–60, Feb 2004.

[84] T. S. Rappaport. Wireless communications : Principles and Practice. Prentice Hall, New Jersey, 1996.

[85] Abiola Fanimokun and Jeff Frolik. Effects of natural propagation environments on wireless sensor network coverage area. In 35th South-Eastern Symposium, March 2003.

[86] Sachin Adlakha and Mani Srivastava. Critical density thresholds for coverage in wireless sensor networks. In Proceedings of the IEEE WCNC '03, 3, pages 1615–1620, March 2003.

[87] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. Computer Networks, 43:499–518, 2003.

[88] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In Proceedings of the ACM SenSys '04, 2004.

[89] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme. Robomote: Enabling mobility in sensor networks. Technical Report CRES-04-006, University of Southern California, 2004.

[90] Marco Zuniga and Bhaskar Krishnamachari. Analyzing the transitional region in low power wireless links. In 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks SECON '04, Oct 2004.

[91] Mohammad Rahimi, Hardik Shah, Gaurav S. Sukhatme, John Heidemann, and Deborah Estrin. Studying the feasibility of energy harvesting in a mobile sensor network. In Proceedings of the IEEE International conference on Robotics and Automation (ICRA '03), 1, pages 19–24, Sep 2003.

[92] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. Mobile Computing., 353, 1996.

[93] C. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (AODV) routing. IETF RFC 3561, 2003.

[94] D. Nicules and B. Nath. Ad-hoc positioning system (APS) using AoA. In Proceedings of the IEEE INFOCOM, 2003.

[95] Jerry D. Gibson. Mobile Communications Handbook. Springer-Verlag, 1999.