A FRAMEWORK FOR ENABLING ENERGY EFFICIENT SEMANTIC VIEWS IN WIRELESS SENSOR NETWORKS FOR DATA INTENSIVE APPLICATIONS

by

Hui Ling

M.S , University of Pittsburgh, 2005

Submitted to the Graduate Faculty of

the Department of Computer Science in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTER SCIENCE

This dissertation was presented

by

Hui Ling

It was defended on

December 7th 2009

and approved by

Dr. Taieb Znati, Department of Computer Science, and Graduate Program Telecommunications and Networking

Dr. Daniel Mossè, Department of Computer Science

Dr. Youtao Zhang, Department of Computer Science

Dr. Louise Comfort, Graduate School of Public and International Affairs (GSPIA)

Dissertation Director: Dr. Taieb Znati, Department of Computer Science, and Graduate Program Telecommunications and Networking

ABSTRACT

A FRAMEWORK FOR ENABLING ENERGY EFFICIENT SEMANTIC VIEWS IN WIRELESS SENSOR NETWORKS FOR DATA INTENSIVE APPLICATIONS

Hui Ling, PhD

University of Pittsburgh, 2010

Sensor networks have been envisioned to be a promising technique for data intensive applications such as disaster management and emergency response and are being designed and deployed for these applications [1]. The effectiveness of sensor networks in providing information is determined by human's capacity to recognize and comprehend information from the raw data collected, and act accordingly. Finding relevant information from the large amount of data, however, becomes a challenging problem because user interests continues to grow as the number and variety of sensors increase and users expect to receive only the data they select to view. Transmitting users irrelevant data during data processing not only overloads users with unneeded data but also incurs unnecessary communication overhead. Furthermore, the user interests may be correlated when a large number of users seek information from sensor networks. As a result, a lot of redundant data transmission can be incurred during processing in resource-constrained sensor networks. Data aggregation, though effective in reducing data transmission for aggregated queries, doesn't take the correlation among user interests into consideration during processing. Therefore, additional techniques need to be proposed to provide efficient information delivery for correlated user interests in resource-constrained sensor networks.

To bridge the gap between data collected by sensors and the information interests of users, the concept of "semantic view" is proposed in this thesis. The semantic view is a powerful

abstraction which allows the fusion of multi-sensor and multi-source data into a virtual data gathering and analysis infrastructure commensurate with the interest of an end user. The main challenge is to enable semantic views in an energy efficient manner in resource constrained sensor networks. To that end, a framework which consists of five protocols and algorithms, "Query Aware Sensing", "Probabilistic Query Dissemination", "Correlated Multi-query Processing", "Location Discovery using Out-of-Range information with multi-lateration", and "End-to-end pairwise key establishment" is presented. The ultimate goal is to develop an energy efficient and secure framework towards enabling semantic views in sensor networks for data intensive applications.

TABLE OF CONTENTS

ACKNOWLEDGEMENT xvi

1.0 INTRODUCTION 1

1.1 Background and Motivation 1

1.2 Problem Statement 6

1.3 Thesis Approach 7

1.4 Summary of Contributions 11

1.4.1 Query Aware Sensing 12

1.4.2 Probabilistic Query Dissemination 13

1.4.3 Correlated Multi-Query Processing 13

1.4.4 Location Discovery for Semantic View Processing 14

1.4.5 Secure Message Exchange for Semantic View Processing 14

1.5 Thesis Organization 15

2.0 LITERATURE REVIEW 16

2.1 Energy Efficient Coverage in Sensor Networks 17

2.2 Probabilistic Query Dissemination in Sensor Networks 21

2.3 Query Processing in Sensor Networks 24

2.4 Location Discovery in Sensor Networks 27

2.5 Summary 31

3.0 OVERVIEW OF THE FRAMEWORK 32

3.1 System Model 32

3.1.1 Network Model 32

3.1.2 Energy Model 33

3.1.3 Semantic View Definition . . . 34
3.1.4 Query Definition . . . 34
3.2 Framework Architecture . . . 36
4.0 QUERY AWARE SENSING . . . 40
4.1 Problem Statement . . . 40
4.2 GRASS: A GReedy Algorithm for Sensing Scheduling . . . 42
4.2.1 *COV* Derivation . . . 42
4.2.2 Weight Computation . . . 43
4.2.2.1 Faces Over Areas . . . 46
4.2.3 *COV* Update . . . 47
4.2.4 Incremental Scheduling . . . 47
4.3 Analysis . . . 48
4.4 Simulation Results . . . 49
4.4.1 Methodology . . . 49
4.4.2 Performance Comparison . . . 50
4.4.3 Communication Overhead . . . 54
4.4.4 Ratio of Communicating over Sensing Energy Consumption . . . 55
4.5 Summary . . . 58
5.0 PROBABILISTIC QUERY DISSEMINATION . . . 59
5.1 Problem Statement . . . 59
5.1.1 Probabilistic Forwarding . . . 60
5.2 Area Coverage-based Probabilistic Forwarding . . . 61
5.3 Copies Coverage-based Probabilistic Forwarding . . . 63
5.4 Area and Copies Coverage-based Probabilistic Forwarding . . . 65
5.5 Neighbor Coverage-based Probabilistic Forwarding . . . 67
5.6 Uncovered Sensors After Probabilistic Forwarding . . . 68
5.7 Simulation Results . . . 69
5.7.1 Methodology . . . 69
5.7.2 Performance Comparison . . . 70
5.7.2.1 ACPF, CCPF, ACCPF and NCPF . . . 70

5.7.2.2 ACPFM, CCPFM, ACCPFM and NCPFM 71
5.8 Summary 76
6.0 CORRELATED MULTI-QUERY PROCESSING 80
6.1 Problem Statement 81
6.2 Overview 83
6.3 Correlated Multi-query Processing at the Base Station 86
6.3.1 Shared Intermediate Views 86
6.3.2 Range-Based SIVS Construction 88
6.3.3 Query Processing using SIVS 92
6.3.4 SIVS Update and Query Rewriting for New Queries 95
6.4 Correlated Data Collection at Sensor Nodes 96
6.5 Analysis 99
6.6 Simulation Results 101
6.6.1 Methodology 101
6.6.2 Performance Comparison 102
6.6.3 Communication Overhead 107
6.7 Summary 112
7.0 LOCATION DISCOVERY FOR SEMANTIC VIEW PROCESSING 114
7.1 Problem Statement 114
7.2 Out-of-Range Information 116
7.2.1 Case 1: N is a reference node and U has two neighboring reference nodes 117
7.2.2 Case 2: N is an unknown node with two neighboring reference nodes and U has two neighboring reference nodes 118
7.2.3 Case 3: N is an unknown node with one neighboring reference node and U has two neighboring reference nodes 119
7.3 Localization Scheme 120
7.4 Simulation Results 123
7.4.1 Methodology 123
7.4.2 Effect of h 124

7.4.3 Performance Comparison . . . 124
7.5 Summary . . . 128
8.0 SECURE MESSAGE EXCHANGE FOR SEMANTIC VIEW PROCESSING . . . 129
8.1 Attack Model . . . 129
8.2 End-to-End Pairwise Key Establishment . . . 130
8.2.1 Key Pre-distribution Scheme and Path Key Exposure . . . 130
8.2.2 End-to-End Pairwise Key Establishment using Multiple Secure Paths 131
8.3 Security Analysis . . . 133
8.4 Overhead Analysis . . . 137
8.5 Summary . . . 138
9.0 CONCLUSION AND FUTURE WORK . . . 139
9.1 Future Work . . . 143
APPENDIX. ALGORITHMS AND THEOREMS . . . 145
A.1 Minimum Set k Covering . . . 145
A.2 Gabow's Algorithm for Maximum Matching on Graphs . . . 147
A.3 Security Analysis . . . 147
A.3.1 Proof of Lemma 2 . . . 149
A.3.2 Proof of Lemma 3 . . . 152
BIBLIOGRAPHY . . . 153

LIST OF TABLES

1 Two types of field in the simulation 50
2 Average node degree *deg* in a network of n nodes for probabilistic forwarding 70
3 Parameters for schemes simulated 71
4 Low density network 73
5 Medium density network 74
6 High density network 75
7 Low density network with m-technique 77
8 Medium density network with m-technique 77
9 High density network with m-technique 79
10 Average node degree *deg* in a network of n nodes for location discovery . . . 124
11 Notation 133

LIST OF FIGURES

1 Information flow in disaster management 4
2 Semantic views in disaster management system 9
3 Semantic view abstraction 9
4 Query example in sensor networks 33
5 The framework for enabling energy efficient semantic views in sensor networks 37
6 Planar graph from area coverage 45
7 Average sensing energy consumption in a moderately covered field 51
8 Average sensing energy consumption in a densely covered field 52
9 Average sensing lifetime in a moderately covered field 53
10 Average sensing lifetime in a densely covered field 53
11 Average sensing lifetime with communication overhead in a moderately covered field 54
12 Average sensing lifetime with communication overhead in a densely covered field 55
13 Average sensing lifetime for two ratios of communication over sensing energy consumption in a moderately covered field 56
14 Average sensing lifetime for two ratios of communication over sensing energy consumption in a densely covered field 56
15 Average sensing lifetime with communication overhead for two ratios of communication over sensing energy consumption in a moderately covered field . . 57
16 Average sensing lifetime with communication overhead for two ratios of communication over sensing energy consumption in a densely covered field 57
17 Extra area that node N2 can cover by rebroadcast 62

18 Effect of node passivity, k, on ACPF forwarding probability 64
19 Redundancy case 2 64
20 Effect of node passivity, k, on copies coverage-based forwarding probability . 66
21 Example of path directed forwarding 68
22 Number of reachable nodes in a low density network 72
23 Number of messages forwarded in a low density network 72
24 Number of messages received in a low density network 72
25 Number of reachable nodes in a medium density network 72
26 Number of messages forwarded in a medium density network 73
27 Number of messages received in a medium density network 73
28 Number of reachable nodes in a high density network 74
29 Number of messages forwarded in a high density network 74
30 Number of messages received a high density network 75
31 Number of reachable nodes with m-technique in a low density network 75
32 Number of messages forwarded with m-technique in a low density network . . 76
33 Number of messages received with m-technique in a low density network . . . 76
34 Number of reachable nodes with m-technique in a medium density network . 77
35 Number of messages forwarded with m-technique in a medium density network 77
36 Number of messages received with m-technique in a medium density network 78
37 Number of reachable nodes with m-technique in a high density network . . . 78
38 Number of messages forwarded with m-technique in a high density network . 78
39 Number of messages received with m-technique in a high density network . . 78
40 Query routing trees for data collection 82
41 Overview of correlated multi-query processing 84
42 Example of data aggregation for correlated queries 85
43 SIV of two queries 87
44 An example of a correlation graph 92
45 Example of data aggregation 98
46 Example of correlated data aggregation 100

47 Number of data transmission saved by correlated multi-query processing at the base station for 20 queries 103
48 Number of data transmissions saved by correlated multi-query processing at the base station when 1 attribute is sensed 104
49 Number of data transmissions saved by correlated multi-query processing at the base station when 2 attributes are sensed 104
50 Number of data transmissions saved by correlated multi-query processing at the base station when 3 attributes are sensed 104
51 Number of data transmissions saved by correlated multi-query processing at the base station when 4 attributes are sensed 104
52 Number of data transmissions saved by correlated multi-query processing at the base station when 5 attributes are sensed 105
53 Percentage of data transmissions saved by correlated multi-query processing at the base station when 1 attribute is sensed 105
54 Percentage of data transmissions saved by correlated multi-query processing at the base station when 2 attributes are sensed 106
55 Percentage of data transmissions saved by correlated multi-query processing at the base station when 3 attributes are sensed 106
56 Percentage of data transmissions saved by correlated multi-query processing at the base station when 4 attributes are sensed 106
57 Percentage of data transmissions saved by correlated multi-query processing at the base station when 5 attributes are sensed 106
58 Number of data transmissions saved by correlated data aggregation at sensor nodes when 1 attribute is sensed 107
59 Number of data transmissions saved by correlated data aggregation at sensor nodes when 2 attributes are sensed 107
60 Number of data transmissions saved by correlated data aggregation at sensor nodes when 3 attributes are sensed 108
61 Number of data transmissions saved by correlated data aggregation at sensor nodes when 4 attributes are sensed 108

62 Number of data transmissions saved by correlated data aggregation at sensor nodes when 5 attributes are sensed 108
63 Number of data transmissions saved using CORUp1 when 1 attribute is sensed 108
64 Number of data transmissions saved using CORUp1 when 2 attributes are sensed 109
65 Number of data transmissions saved using CORUp1 when 3 attributes are sensed 109
66 Number of data transmissions saved using CORUp1 when 4 attributes are sensed 109
67 Number of data transmissions saved using CORUp1 when 5 attributes are sensed 109
68 Number of data transmissions saved using CORUp2 when 1 attribute is sensed 110
69 Number of data transmissions saved using CORUp2 when 2 attributes are sensed 110
70 Number of data transmissions saved using CORUp2 when 3 attributes are sensed 110
71 Number of data transmissions saved using CORUp2 when 4 attributes are sensed 110
72 Number of data transmissions saved using CORUp2 when 5 attributes are sensed 111
73 Number of data transmissions saved using CORLow when 1 attribute is sensed 111
74 Number of data transmissions saved using CORLow when 2 attributes are sensed 111
75 Number of data transmissions saved using CORLow when 3 attributes are sensed 111
76 Number of data transmissions saved using CORLow when 4 attributes are sensed 112
77 Number of data transmissions saved using CORLow when 5 attributes are sensed 112
78 An example of trilateration 114
79 Location discovery through multi-lateration 115
80 Using Out-of-Range information to resolve an unknown node's position 117
81 Localization algorithm at reference nodes 121
82 Localization algorithm at unknown nodes 122
83 The effect of h in networks with four reference nodes 125
84 Number of resolved sensors after location discovery 125
85 Percentage of resolved sensors after location discovery 127
86 Number of anchor sensors required to resolve locations of all sensors in the network 127
87 An example sensor network after shared key discovery. 131
88 Security analysis of equal path hop count 136
89 Security analysis of a real path set 137

LIST OF ALGORITHMS

1 GRASS: A GReedy Algorithm for Sensing Scheduling 43
2 *COV* derivation 44
3 Planar Graph Construction 45
4 *COV* update 47
5 ACPF 63
6 CCPF 65
7 ACCPF 66
8 NCPF 67
9 SIVS Construction 93
10 Query Rewriting 94
11 SIVS Update and Query Rewriting for New Queries 97
12 Greedy algorithm for set k covering 146
13 E 148
14 L 148
15 R 149

ACKNOWLEDGEMENT

It has been a long journey at University of Pittsburgh to reach this finish line. There are many people I am deeply grateful to for their care and support. Without them, I would not have gone this far.

First, I would like to thank my advisor, Dr. Taieb Znati, for his constant guidance during my study. This thesis would not even be possible without his support and guidance. Dr. Znati has great visions on my research problems. He kept encouraging me to explore new directions and despite being busy, he is always ready for discussion when I need. Through these years, I have learned quite a lot from him on how to become an independent researcher.

My officemates, Chatree Sangpachatanaruk, Anandha Gopalan, Guanfeng Li, Octavio Herrera, Hammad Iqbal, Ihsan Qazi, Paul Dillion and Mehmud Abliz must be thanked for willing to spend time discussing my research and helping me out when I need. They have made my life at University of Pittsburgh less intimidating and more enjoyable.

I also want to thank my committee members: Dr. Daniel Mosse, Dr. Youtao Zhang, and Dr. Louise Comfort for their time and effort through my thesis development. Their comments and suggestions have been invaluable to me and help me to improve the quality of my dissertation. They have became more than just committee members to me, I feel very comfortable to go to any of them for advice and suggestions on everything if I need to.

Finally and most importantly, I want to thank my family for their never-ending support and love. My father, Deihong Ling, and mother Lixian Yu have always believed in me, even when I lose confidence in myself. It is their encouragement and love that keep me moving forward when challenges arise in my life. My wife, Lan Fang, has been extremely patient with my study and always stands by me when I have difficulties in my life. My sister, Ping Ling, has been greatly helpful in taking care of my parents while I am away from them

during my study at Pittsburgh. Their support has been the key to my graduation and with love, I dedicate my thesis to them.

1.0 INTRODUCTION

The development of wireless sensor networks was initially motivated by military applications such as battlefield surveillance, and then used to support industrial and civilian applications including environmental monitoring, protection of critical infrastructures and disaster management and emergency response. As the number and varieties of sensor network applications continue to grow, more and more information about various aspects of the physical world are provided by sensor networks. As a result, sensor networks are becoming a very important part of the networking infrastructure in the future.

1.1 BACKGROUND AND MOTIVATION

Advance in Micro-Electro-Mechanical-Systems has enabled the development of small-size, low-cost, low-power and multi-functional sensor nodes. A sensor node typically consists of sensing boards, a limited capacity processor and communication devices. The sensing boards sample data such as temperature, pressure and light from where the sensor is deployed. The processor allows a sensor to perform some simple processing on the sampled data if needed. The communication devices, such as radio transceivers, allow sensors to talk to each other and collaborate to accomplish complex tasks such as mobile target tracking, which are otherwise impossible for individual sensors to finish.

A Wireless Sensor Network (WSN) is composed of a large number of interconnected sensor nodes and often supports several unique features. First, the positions of sensor nodes need not be engineered or pre-determined during deployment, which allows random deployment in inaccessible terrains or challenging environment in disaster management. Second,

sensor nodes, equipped with limited power processors, can carry out simple computations locally and transmit only the required and partially processed data to a base station, instead of sending all raw data for fusion. As a result, decisions can be made at sensor nodes to respond to certain events more quickly than being made by the base station where all raw data from sensors are received and processed. Furthermore, they can self organize after deployment and function without human intervention. These unique characteristics make sensor networks a promising technique for a wide range of applications including field surveillance, environment monitoring, structural health monitoring and disaster management.

In the past few years, extensive research has been conducted to implement these features in sensor networks and significant progress has been made in various aspects of sensor networks. Now with the needed technology such as medium access control protocols and routing algorithms in place, sensor network has evolved from a mere concept into practical implementations and deployments. Several sensor network systems have been deployed or are being deployed for various types of applications. To name a few, a sensor network has been deployed for real-world habitat monitoring in Great Duck Island, Maine in 2002 [2]. ZebraNet, a wireless sensor network designed and developed at Princeton University, is deployed at Mpala research centre to perform novel studies of animal migrations and inter-species interactions [3]. Researchers at CMU are developing sensor-data driven vertical decision support systems for specific critical infrastructure systems to provide a "nervous system" in these systems such that proactive, intelligent decision support and control can be achieved over their lifetime [4]. Another sensor network has been deployed in Alpine, Switzerland to gather data for permafrost monitoring [5]. An underwater sensor network is also being designed and will be deployed in the costal region of Padang, Indonesia, for near-shore tsunami detection and disaster management [1]. As a result of these deployments, sensor networks will become a critical part of the networking infrastructures in the future.

As a promising technique for collecting data about the real world, which would be either expensive or impossible to collect otherwise, sensor networks provide us an effective means to monitor the real world. What is more important, however, is not only to monitor the real world, but also to assess and react to critical events in the world, such as detecting possible tsunamis from underwater sensor network data and then preparing the communities for the

coming tsunami strikes. It is noted that the "absorptive capability" of human decision makers is limited in complex environments [6]. The human ability to acknowledge and understand new information is limited, and at a certain threshold of exposure, human cognitive process simply shut out information that is too complex or different from previous experiences. The limited capability, in turn, reduces human ability to respond to events such as disasters in a timely manner, particularly in the context of damaging or destructive events. Even though, the human ability to extract information from sensor networks and recognize events such as risk conditions in disaster management and emergency response, can be increased by focusing on data that are directly related to each individual person [7]. In order to be more effective and suited for decision making, sensor networks must be able to provide information specific relevant to each human being in a timely and accurately manner so that each decision maker can absorb and comprehend these information provided by sensor networks.

Towards that end, we start from understanding the limitations of current sensor networks and then present new methods in sensor networks to provide information for human beings. The main approach is to understand the characteristics of the information requirement in these applications and then design new protocols and algorithms in sensor networks to process these requests efficiently.

In these applications, different users or groups of users are often interested at different information from the networks. For one example, in ZebraNet, some zoologists may seek the position and body movement data of wild animals from the network to study their migration patterns. Other zoologists may seek the integration between these data and other data such as weather change and plant life change to understand how the migration patterns of wild animals may be affected by changes in weather patterns and introduction of non-native species. Ecologists may seek the integration of biometric data such as heart rate, body temperature, and frequency of feeding, migration patterns of wild animals, and human activities in surrounding areas to study how human development into wilderness areas affects indigenous species.

For another example, disaster relief usually involves multiple autonomous organizations

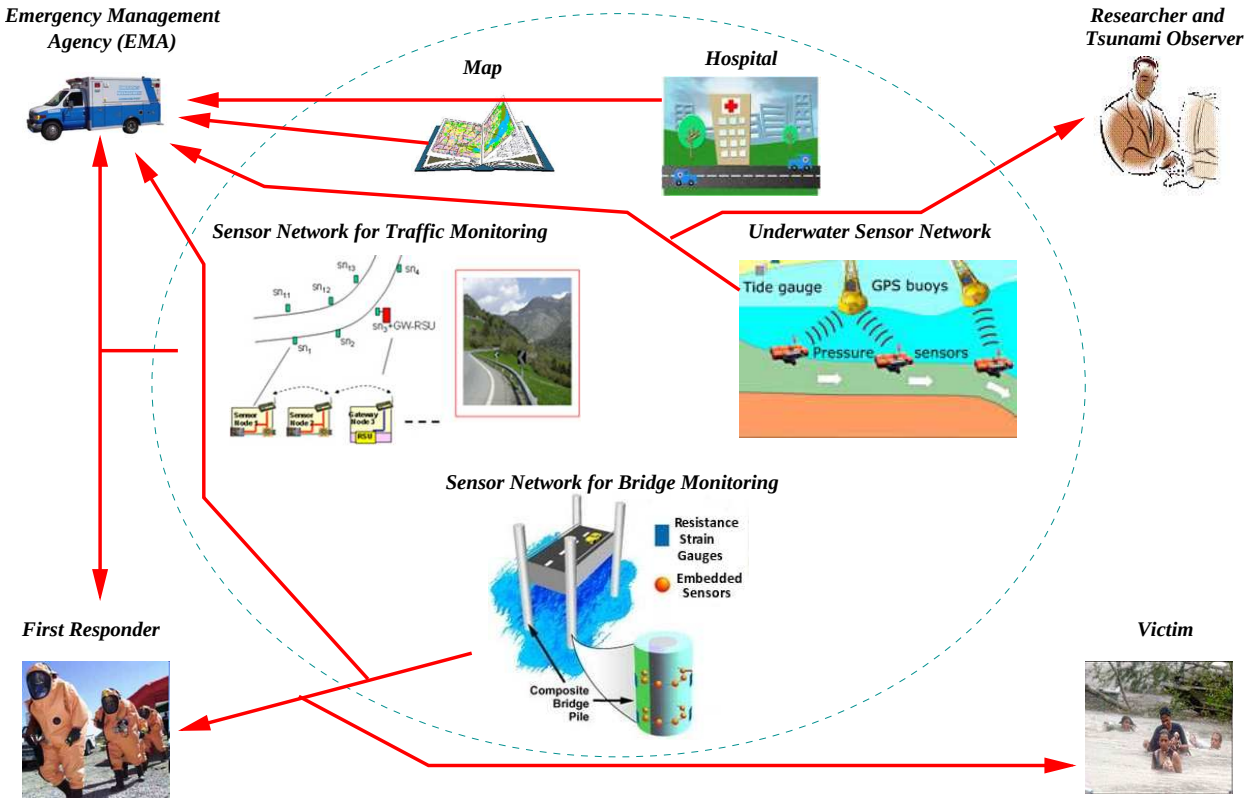

Figure 1: Information flow in disaster management

(governmental, individuals, communities and industry). The difference among organizations leads to a diversity of user interests in information provided by the network. Figure 1 highlights the information flow in a disaster management and emergency response system. In this system, the group of emergency managers seeks the integration of geospatial data about the location of victims with on-line data about the location of medical facilities to provide information needed by first responders in their rescue operations. The first responders, on the other hand, seek traffic information to avoid traffic jams while relocating victims. Yet another group, tsunami observers, is interested in gathering seismic and sea level gauge data in order to estimate the magnitude and speed of the inundation. This information can then be used to issue tsunami warnings of an appropriate level. Other researchers seek to explore the feasibility of correlating the seismic data with sensed data on animal behaviors to explore new ways of identifying disasters.

Even though the information needed by different users may vary, the underlying data sets, from which the information is derived, might overlap. In other words, from the data's point of view, the same data may be needed by many users. Take disaster management and emergency response as an example, in the scenario illustrated in Figure 1, a traffic sensor network provides traffic condition for roads in the disaster area. An under water sensor network collects information on seismic activity, tidal wavelength, etc. An infrastructure sensor network monitors displacement, strain, acceleration of critical buildings, bridges and roads, etc. Each network is of interest to multiple groups of people. The traffic condition of a specific road is needed by first responders in order to plan a clear path for victim transportation. The same information is also needed by emergency management agencies to schedule future logistic transportation. The infrastructure monitoring information is of interest to disaster victims, emergency management agencies (EMA) and first responders to avoid potential dangerous places. Redundant data transmissions can be incurred during data collection for the correlated user interests if they are processed separately by the sensor networks, which can lead to the waste a significant amount of energy in sensor networks.

On the other hand, as the number and variety of sensors increase, so does the volume of data generated by these sensor networks. The effectiveness of sensor networks in providing information is determined by human's capacity to recognize and comprehend information

from the raw data collected, and act accordingly. Sensor networks should not only collect data from the physical world, but also facilitate users to extract and absorb information specific to their needs. Finding relevant information from the continuously increasing amount of data for various kinds of user interests hence becomes a challenging problem because users expect to receive only the relevant information they select to view. Overloading users with irrelevant data is almost as bad as providing users no data, because by the time users identify the needed data from the large amount of data they receive, it might be already too late to act correspondingly. Furthermore, transmitting irrelevant data to users also incurs unnecessary data communications in the sensor networks, which can be a major issue in a long run in energy-constrained sensor networks.

In-network processing and data aggregation are presented in [8][9] to answer aggregated queries such as "average" or "min". In these schemes, sensor data is aggregated at intermediate sensor nodes to reduce the amount of data being transferred in the network during processing. This approach, referred to as "in-network aggregation" can reduce redundant data transmissions in sensor networks, in comparison to the approaches in which data is collected and aggregated centrally. However, data aggregation doesn't reflect the correlation among user interests. A large number of data may be processed and transmitted multiple times to users with similar interests, which can cause significant communication overhead in data intensive sensor networks. Therefore, additional techniques are needed to understand the correlations among user interests and provide efficient data collection for these user interests in data intensive sensor networks.

1.2 PROBLEM STATEMENT

Given a set of diversified and yet correlated user interests in data intensive sensor networks, a naive approach for satisfying these user interests is to deliver all sensor data in the network to users and let the users to retrieve what they need. This approach, apparently causes users be deluged with unneeded data and hence significantly reduces the usability and effectiveness of the system. Another alternative method is to use in-network processing such as data

aggregations to reduce data communications in the network by local computations in sensor nodes. This method, however, as we have pointed out above, does not take user interests into consideration and therefore may still deliver redundant data to users. The overall research question to be answered in this thesis then is

- **Is it possible to design a general framework in data intensive sensor networks to provide only relevant information for a large set of correlated user interests? If so, can the user interests be further processed energy efficiently in the resource-constrained sensor networks?**

Specifically, the main challenges of designing such a framework in data intensive sensor networks to achieve energy efficient information delivery for multiple users/user groups are

- How to identify and select the relevant data for each user from the large volume of data in sensor networks?
- How to efficiently collect the relevant data for a set of diverse and yet correlated user interests in sensor networks?
- How to capture the correlations among user interests and avoid redundant data transmissions during processing of these interests?

1.3 THESIS APPROACH

To close the gap between sensor data and user interests in data intensive applications, we propose the concept of semantic view, an abstraction to support mission-aware information delivery commensurate with and relevant to the goals and needs of users. More specifically, a semantic view is s set of queries, which specifies a set of predicates on specific data types and timing and location constraints. An instantiation of a semantic view is a dynamically created logical grouping of collaborative sensors and monitoring devices whose task is to process, filter and fuse a flood of data into accurate and actionable information for decision makers, as specified by the semantic view.

Figure 2 depicts a multi-layered architecture of a disaster management system and illustrates a set of different semantic views, V_i, $i = 1, 2, \cdots, n$, each of which reflects the mission and interests of a specific organization, O_i, $i = 1, 2, \cdots, n$. The semantic view of the Emergency Operation Center (EOC) captures a global, yet aggregated, view of the system, while the semantic views of other organizations may express interest in gathering specific data types under specific constraints. The semantic view is a powerful abstraction which allows the fusion of multi-sensor and multi-source data into a virtual data gathering and analysis infrastructure commensurate with the interest of the underlying organization.

Conceived to be independent of a specific application, the concept of semantic views is well suited to address the gathering and aggregation of multiple types of sensor data across multiple operational domains. As such, it provides users the ability to maintain absolute time sequencing of data from various sensors within the system, enforce timing and location constraints as specified by the underlying semantic view, and analyze temporal and spatially collected data. Users will no longer be confined to receiving data that only marginally reflects the current situation. Instead, a semantic view maps the query predicates and constraints onto a dedicated set of sensors and monitoring devices for the resolution of the expressed interests. A semantic view allows users to access directly the information they seek, rather than having to depend on information being pushed to them.

In summary, a semantic view of a user/user group is a tuple $< D, \Pi >$, where Π is a set of interests or constraints a user has and D is the set of data in the network which meets these interests. In order for sensors to understand and process a semantic view, a set of queries Q is constructed which returns exactly the same data, D, as requested by a user. As illustrated in Figure 3, a semantic view is an abstraction bridging the gap between sensors which can process queries and users who are interested at data.

In order to support semantic views in sensor networks, the relevant data of semantic views must be identified, selected and collected from sensors. To this end, a framework for enabling semantic views is proposed. The main components of the framework are sensing

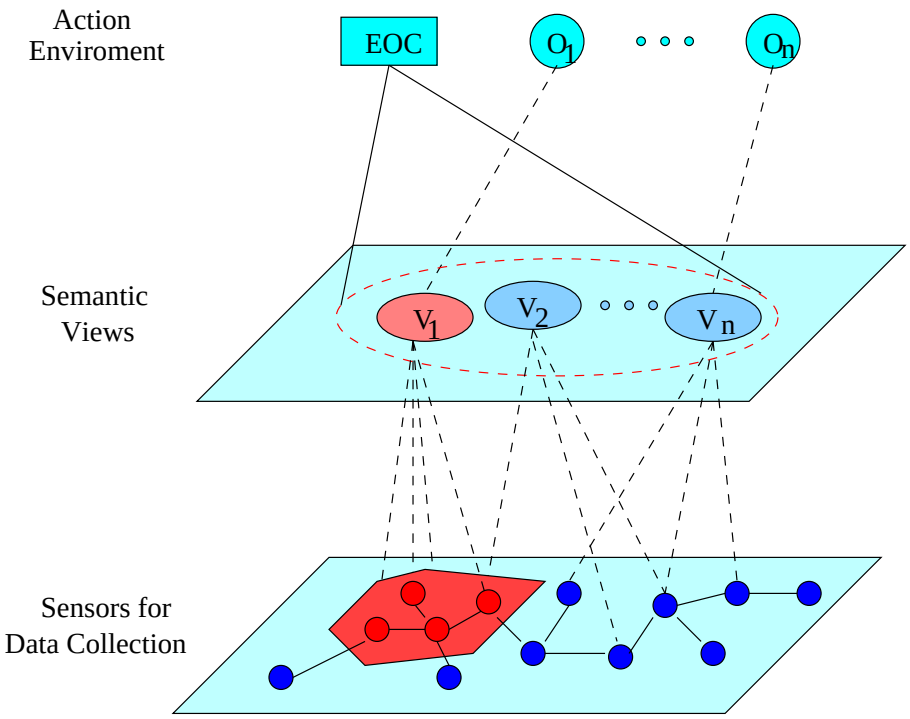

Figure 2: Semantic views in disaster management system

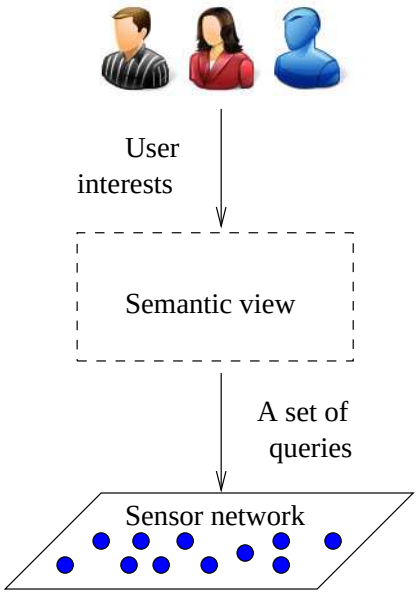

Figure 3: Semantic view abstraction

scheduling, query dissemination, query processing, location discovery and end-to-end pairwise key establishment for secure communications. The first three components address the data identification, selection and collection, respectively, and location discovery provides location information to sensors so that the location constraints of queries can be understood by sensors. The design principle of all these components is that they must be implemented in an energy efficient way to suit for resource-constrained sensor networks. The design is validated through simulations of each component, respectively.

The first component deals with data sensing. It aims to figure out what sensors should sample data in order to provide the required data for current users' semantic views. Consider a multiple purpose sensor network, where each sensor can sense multiple types of data with different amount of energy consumption. Based on user interests, the data to be sensed in the field may vary from area to area, as well as from time to time. As a result, the set of working sensors might also change frequently. Furthermore, many different combinations of sensors may exist to sample data for a set of user queries. To preserve energy consumption, the set of sensors, which spends the minimum amount of energy, should be chosen to sample the data requested by the current set of queries.

After a set of sensors are scheduled to sense data, the queries in users' semantic views must be delivered to relevant sensors for data collection. A naive way is to broadcast the queries to the network so all sensors can receive them. The broadcast approach, although guarantees that each sensor receives at least one copy of the queries, also sends queries to irrelevant sensors and deliver multiple copies to relevant sensors. Since essentially only one copy of the queries is needed by relevant sensors, the redundant transmissions in the broadcast approach will cause a large amount of energy being wasted. The challenge is how to reduce the amount of energy spent at sensor nodes in delivering queries to relevant sensors in the network.

The final step of semantic view processing is data collection. As pointed out above, the same data at a sensor node may be fused into several users' semantic views. These data are unnecessarily transmitted and aggregated multiple times if the queries of user semantic views are processed separately. In a long run, these redundant transmissions may lead to a significant waste of energy. Our approach is to look at the multiple queries together and

find ways to reuse the shared data among them during processing. The challenge is how to identify and reuse shared data among queries to reduce data communication cost in multiple query processing, while preserving the semantic correctness of query processing result.

In addition, locations of sensor nodes must be known in the proposed framework for several reasons. First, sensors locations must be known in order to compute the coverage level of the deployment field. With this information, the desired level of coverage can be ensured when sensors are turned off to preserve energy consumption at sensor nodes. The second reason is that when the queries in user semantic views specify geographical constraints, e.g. the data from a particular area is needed, a sensor needs to know its location in order to determine if its data is required for these queries. Location discovery, therefore, is an indispensable component in the framework and must be addressed as well. A naive approach is to equip each sensor with an external device like a GPS receiver so that it knows its location all the time. However, using such external devices not only makes a sensor much more expensive, but also increases the energy consumption of a sensor by several orders of magnitudes. Multi-lateration schemes, in contrast, rely on a small set of anchor sensors to discovery other sensors' locations through message exchanges. The multi-lateration scheme can significantly reduce the number of anchor sensors needed to discover other sensors' locations. The challenge is how to further minimize the number of the initial anchor sensors to reduce the overall cost and energy consumption of sensor networks.

Furthermore, since semantic view processing relies on information exchange among sensors for collecting data from sensors, attackers can also gain these data by capturing and analyzing all the messages exchanged among sensors. To secure the sensor communications against eavesdropping and traffic analysis by attackers, a key management scheme is needed to establish keys among sensors to encrypt and decrypt their message exchanges.

1.4 SUMMARY OF CONTRIBUTIONS

This thesis studies the information delivery problem in sensor networks for data intensive applications such as disaster management and emergency response. To satisfy the diverse, yet

correlated information needs of users in these systems, the concept of "semantic view" is proposed. The semantic view is a powerful abstraction, which allows the fusion of multi-sensor, multi-source data into a virtual data gathering and analysis infrastructure commensurate with the interest of the underlying organization. It further allows users not to be overloaded by the huge amount of data generated in sensor networks and retrieve information only from relevant sensor nodes. The main contributions of this thesis are the concept of semantic views and a set of protocols and algorithms in the framework towards enabling semantic views in resource constrained sensor networks.

The framework consists of a set of efficiently designed protocols and algorithms, which not only enable semantic views in sensor networks, but also address energy efficiency in their designs so that they can be well suited to resource-constrained sensor networks. The specific components of the framework are:

- Query aware sensing adapts the sensor sensing scheduling to dynamic sensing requirement from semantic views so that a minimum amount of energy is spent on sampling data for user semantic views.
- Probabilistic query dissemination aims to reduce the number of messages for delivering semantic views to relevant sensors through probabilistic forwarding.
- Correlated multi-query processing reduces redundant transmissions for semantic view data collections by identifying and reusing shared sensor data among user semantic views.
- Location discovery using out-of-range information with multi-lateration reduces the number of anchor sensors to discover the locations of other sensors in the network.
- End-to-End pairwise key establishment scheme allows sensors to set up and use symmetric keys to secure their data communications against eavesdropping attack and compromised sensors.

1.4.1 Query Aware Sensing

Query aware sensing derives the coverage requirements from user semantic views and computes a minimum set of active sensors which should sample data in order to answer the user semantic views. The set of active sensors are dynamically updated as the level of cover-

age of user semantic views changes from time to time. This way, only the sensors which must be active are required to sense data at any time. Therefore, a minimum amount of energy is spent to sense relevant data which is needed for answering semantic views. By adapting sensors' sensing scheduling to different levels of coverage over different areas in the deployment field and different time periods, query aware sensing can further reduce sensing energy consumption in comparison to sensing with a static level of coverage, while ensuring all relevant data to user semantic views is sampled.

1.4.2 Probabilistic Query Dissemination

To reduce propagation cost for semantic view dissemination, probabilistic forwarding techniques are proposed to deliver semantic views to sensor nodes in the network. Several schemes, which adapt the probability of forwarding a semantic view at a sensor node to various types of local topology information, i.e. transmission range and neighborhood information, are presented and studied. The design principle is to differentiate forwarding probabilities among neighboring sensor nodes such that all relevant sensors of user semantic views receive a copy with a minimum number of messages being forwarded in the network. These schemes can further reduce the number of messages needed to semantic view dissemination in sensor networks, in comparison to other gossip based broadcast schemes.

1.4.3 Correlated Multi-Query Processing

In correlated multi-query processing, a numerical model is developed to estimate how much data is shared among queries in a semantic view and between semantic views based on the query constraints. From the estimated size of shared data among queries, shared intermediate views are then constructed to maximize reusing of shared data during processing. In principle, a shared intermediate view is only processed once and its result is reused for the queries from which the shared intermediate view is constructed. This way, the scheme reduces the number of data transmission by processing the shared intermediate views only once and reusing their results for other queries. The queries are also transformed in such a way that the results of shared intermediate views can be reused and the semantic correct-

ness of final processing results can be ensured. In addition, correlated data collection is also implemented at sensor nodes. In correlated data collection, each sensor node stores its data to a proxy sensor node, which is closer to the base station. After a proxy sensor has been established for a sensor node, any semantic view requesting its data retrieves the data from its proxy node. By delegating data to a closer proxy node, a sensor node saves the message transmissions from itself to its proxy node for future data requests.

1.4.4 Location Discovery for Semantic View Processing

The basic tenet of this location discovery scheme is the concept of "Out-of-range" information, which is based on the observation that if two sensors cannot hear from each other, then the distance between them must be larger than the transmission range of both sensors. This information can be easily obtained by maintaining a neighbor list at each sensor node. Any non-neighboring sensor of a sensor node can be inferred as out of its range. The out-of-range information, when combined with multi-lateration scheme, can be very useful to resolve location ambiguities of unknown sensors. The conditions that these out-of-range information can be used to resolve location ambiguities are developed for reference nodes and unknown nodes in different scenarios. An unknown sensor with location ambiguity, asks helps from these out-of-range nodes through multi-hop paths and these out-of-range nodes determine if they can help to resolve its location by checking the condition which may apply. It is shown that, with out-of-range information, fewer reference nodes are needed to locate sensors in the network, which in turn reduce cost and energy consumption of the whole network since reference nodes are usually much more expensive and consumes more energy.

1.4.5 Secure Message Exchange for Semantic View Processing

To secure message exchanges for semantic view processing, an end-to-end pairwise key establishment scheme based on key pre-distribution is presented. This scheme allows any two sensors to set up a common symmetric key after key pre-distribution and path key establishment. These keys are then used to protect data communication links between sensors against packet eavesdropping and traffic analysis by attackers. The scheme also protects

data communications among normal sensors from being exposed to compromised sensors.

1.5 THESIS ORGANIZATION

This chapter introduces the main challenges of information processing for correlated user interests in data intensive sensor networks and gives a brief overview of the semantic view approach and its main contributes to address the information processing problem. The rest of the thesis is organized as follows. In Chapter 2, the related work to the thesis, i.e. location discovery, sensing scheduling, query dissemination and query processing in sensor networks are discussed in detail. Chapter 3 gives an overview of the proposed framework. The detailed schemes of the framework, i.e. query aware sensing, probabilistic query dissemination, correlated multi-query processing and location discovery using Out-of-Range information with multi-lateration for semantic view processing are presented at Chapter 4 to 7, respectively. Chapter 8 elaborates the design and analysis of the end-to-end pairwise key establishment scheme for semantic view processing. Chapter 9 summarizes the major work and contributions in the thesis and discusses the directions of future works.

2.0 LITERATURE REVIEW

The purpose of this chapter is to present a detailed review of the literature and the background that are related to this thesis. Over the past ten years, a large amount of effort has been dedicated to various aspects of research of sensor networks, to address the energy constraints of sensors and special traffic patterns in sensor networks. New protocols have been proposed for medium access control (MAC), routing, security, sensor query processing and dissemination protocols etc. Specifically, self configured MAC protocols have been proposed to coordinate the sensor schedule of data transmissions to preserve the energy consumption of individual sensor nodes [10][11][12][13][14]. Many of these schemes are based on or have been motivated by MAC protocols for ad hoc networks [15][16][17]. In addition to the existing protocols for end to end routing in ad hoc networks [18][19][20][21][22], data centric routing protocols are specially designed for sensor networks to focus on data delivery and energy efficiency [23][24][25][26][27]. Because of the ad hoc manner of deployment of sensor networks and the limited amount of energy available at sensor nodes, key pre-distribution based schemes have been proposed to provide cryptographic protection of communications in sensor networks [28][29][30][31][32] [33][34][35][36][37][38] [39][40][41][42][43]. Key management schemes for secure group communications are also being studied for ad hoc and sensor networks [44][45][46] [47][48][49][50] [51][52][53][54][55]. To reduce the communication cost of query processing and data collection in sensor networks, extensions of structured query language like systems to sensor networks have been proposed [56][57], as well as new techniques such as in network processing and data aggregation [58][8][9][59].

With the development of these new techniques and advances in Micro-Electro-Mechanical Systems, it has become feasible to implement a large scale sensor network at a low cost. Nowadays, many sensor networks have been deployed for environment monitoring, traffic

monitoring, field surveillance and disaster management [2][60][61][62][63]. These systems are functioning as very important sources of information for users and have become an essential part of the network infrastructure of the future. The information provided by sensor networks, can lead to a better decision making for users when integrated with information from other sources [64][65][66][67][68][69][70][71]. For example, the traffic information collected from sensor networks, integrated with geographic maps, can help users to plan trips without traffic congestion. The data integration between sensor networks and other data sources has been studied recently and is still being investigated.

This thesis is focused on energy efficient mission-aware information delivery for data intensive applications in sensor networks. It mainly address sensing coverage, query dissemination, query processing and location discovery in sensor networks. In the following, related work to these four problems is reviewed in detail.

2.1 ENERGY EFFICIENT COVERAGE IN SENSOR NETWORKS

The idea of putting redundant sensors into sleep mode has been explored as a method to preserve the limited energy at sensor nodes. It is widely used in designing energy efficient medium access control protocols in sensor networks [10][11][12][13][14]. In these schemes, sensors follow a periodic listen/sleep schedule. If no data needs to be transmitted or forwarded for other sensors at one sensor node during the listen period, it turns off its radio and goes into sleep state. It wakes up and listens again when its sleep period is over. Neighboring sensors form virtual clusters to auto synchronize sleep schedules such that when a sensor transmits data, all neighboring sensors go to sleep except the receiving sensor. By turning off its radio and going to sleep, a sensor can save energy because it cannot transmit anyway when one of its neighboring sensors is transmitting. This method is also utilized to design coordinated routing schemes for energy efficient data transmission in sensor networks [72][73]. These schemes take a cross-layer approach, where routing information is integrated with wake-up schedules for various sensors to increase sensor network longevity. In these schemes, sensors along a routing path are well coordinated such that when one

intermediate sensor in a path finishes its transmission and goes to sleep, the next sensor in the route is ready to start a transmission. As a result, not only is energy preserved at sensor nodes, but the end to end communication latency is ensured to be small in the network as well.

The same idea has been explored as a way to achieve energy efficient coverage in sensor networks [74][75][76] [77]. One scheme, PEAS, presented in [74], extends network lifetime by maintaining a necessary set of working nodes and turning off redundant ones. Sensors alternate among three states: sleeping, probing and working. Sleeping sensors wake up once in a while to probe their neighborhood and replace any failed working sensors as needed. PEAS has two components: probing environment and adaptive sleeping. Probing environment allows a newly waken up node to probe its local neighborhood to discover whether a working node exists within a certain probing range. If no working node exists in that range, it starts working. Otherwise, it sleeps again. Adaptive sleeping decides when a sleeping sensor should wake up again. The scheme is distributed and localized and has low complexity, but it does not preserve the original coverage area. The scheme described at [75] aims to reduce energy consumption by scheduling nodes to sleep and adjusting sensing range. In the proposed approach, each sensor in the network autonomously and periodically makes decisions on whether to turn itself on or off using only local neighbor information. To preserve sensing coverage, a node decides to turn itself off only when it discovers that its neighbors can help it to monitor its whole working area. To avoid a blind point, which may appear when two neighboring sensors expect each other to work, a backoff based scheme is introduced to let each node delay its decision for a random period of time. A set of conditions are then developed to check whether a sensor's working area is covered by other neighboring sensors for different scenarios where sensors may have different sensing ranges and different information. The work focuses on uniform one coverage in the deployed field and must be modified in order to support differentiated coverage requirements over the entire field.

Coverage Configuration Protocol (CCP) and Optimal Geographical Density Control (OGDC) consider both coverage and connectivity in sensor networks [76][77]. The work in [76] proves that when the communication range of sensor nodes is bigger than or equal to twice their sensing range, then a set of nodes achieving a $k - covered$ network also ensures

a $k-connected$ network. The relationship between the level of coverage and connectivity is also quantified, based on which the coverage configuration protocol is designed to achieve different levels of coverage requested by applications. When the sensing range is higher than half of the communication range, the CCP doesn't ensure connectivity. In this case, the CCP is integrated with an existing connectivity maintenance protocol, SPAN [78], to provide both sensing coverage and communication connectivity. SPAN is a decentralized coordination protocol that conserves energy by turning off unnecessary sensors while maintaining a communication backbone composed of active sensors. The communication backbone maintains the topology of the network such that all active sensors are connected through the backbone and all inactive sensors are directly connected to at least one active sensor. A sensor becomes active if it needs to be active at SPAN or CCP and goes to sleep only if it is not eligible according to SPAN or CCP.

OGDC [77], on the other hand, shows how to optimally choose the subset of working nodes under the assumption that node density is sufficiently high. A set of optimal conditions under which a subset of working sensor nodes can be chosen for complete coverage is derived under the ideal case that node density is sufficiently high. Based on the optimal conditions, a decentralized algorithm, OGDC, can be devised for density control in large scale sensor networks. The OGDC algorithm is fully localized and can maintain coverage as well as connectivity, regardless of the relationship between the radio range and the sensing range.

A probabilistic coverage protocol was presented recently in [79]. This protocol aims to investigate the effect of a sensing model on the design of coverage protocols in sensor networks. The disk model, exponential model, staircase model and probabilistic model are considered and a new probabilistic coverage protocol (PCP) is proposed to adapt coverage protocols to probabilistic sensing models. This scheme, however, may not be suitable for applications that require a coverage level of more than one or depend on dynamic characteristics of the event, as the authors mentioned in their paper.

In addition to the area coverage discussed above, in which the whole area must be covered by sensor nodes, point coverage in sensor networks has also been studied [80][81][82][83][84][85]. Point coverage is useful for applications such as target tracking, where only points at which the target might appear should be covered instead of the whole area. In point

coverage, one method to extend sensor network lifetime is to divide sensors into disjoint sets such that every set completely covers all target points and let these sets activate successively. By decreasing the fraction of time a sensor is active, the overall time until power runs out for all sensors is increased. The disjoint set coverage problem is NP-complete, and any polynomial time approximation algorithm has a lower bound of 2 [80]. In [81], the k-coverage and network connectivity problem for point coverage is discussed. Given k, the coverage and connectivity problem requires each target being covered by at least k sensors, while those active sensors being connected as well. To solve this problem, a linear programming based centralized algorithm and two distributed algorithms have been proposed. In [82], two algorithms for efficient placement of sensors are presented to optimize the number of sensors and determine their placement to support distributed sensor networks. These algorithms address coverage optimization under the constraints of imprecise detections and terrain properties. Furthermore, they are targeted at maintaining average coverage as well as at maximizing the coverage of the most vulnerable grid points. The same coverage and placement problem in a three dimensional field is studied in [83]. The problem is shown to be NP-hard, and polynomial time approximation algorithms with proven approximation ratios are presented. In [84], a novel sensor network coverage maintenance protocol called Coverage Aware Sensor Engagement (CASE) was designed to efficiently maintain the required degree of sensing coverage by activating a small number of sensors while putting the others in sleep mode. CASE schedules active/inactive sensing states of a sensor according to the sensor's contribution to the network sensing coverage, which is quantitatively measured by a metric called "coverage merit". By activating sensors with relatively large coverage merit and deactivating those with small coverage merit, CASE effectively achieves energy conservation while maintaining sufficient sensor network coverage.

2.2 PROBABILISTIC QUERY DISSEMINATION IN SENSOR NETWORKS

A naive approach to deliver query q to its relevant sensor nodes is to flood q into the network. In flooding, each sensor broadcasts query q to its neighbors when q is received. Apparently, in this approach, a node may and often does receive multiple copies of q. However, as a matter of fact, only one copy of q is needed at each sensor and all other copies are not needed. As a result, a large number of redundant transmissions are incurred during flooding [86]. Probabilistic approaches such as Gossip were initially proposed to resolve inconsistencies among database servers [87]. When a database is replicated at many sites, maintaining mutual consistency among the sites in the face of updates is a significant problem. It has been shown that deterministic algorithms for replicated database consistency can be replaced with simple randomized algorithms. In randomized approaches, a site randomly updates other sites during maintenance. The probability of inconsistency can be made arbitrarily small by carefully configuring the random updating process. This problem shares a lot of similarity with the routing request transmission in ad hoc networks and sensor networks.

In [88], Gossip is integrated with an ad hoc routing protocol to reduce the overhead of sending routing requests into nodes in the network. Several probabilistic schemes are presented to send routing requests to nodes in the network with high probability. In the basic approach, a node, upon receiving a routing request message, m, forwards m to its neighboring nodes with probability p. A very high probability that all nodes receive a copy of the broadcast message, m, can be achieved if p is sufficiently high. To prevent the early death of m, it is also suggested that the first several hops should always forward m to their neighboring nodes. Other more complex schemes adapting p to local information, such as the number of neighbors at sensor nodes, are also presented and discussed.

In the work described at [89], a Gossip-based broadcast scheme is investigated for heterogenous and dynamic networks. In these networks, it is impossible to adjust the parameters of the Gossip algorithm off line. Instead, it must be dynamically adjusted to current network conditions. A node's gossip rate is adjusted according to the resources available within other nodes in the network. This information, required to perform adaptation, is embedded in the

normal gossip of data messages and exchanged among nodes through these data messages. Global congestion information is used to control the message emission rate at nodes which want to transmit data.

The probability can be further adapted to each node's coverage information in the network [90]. In such schemes, the contribution of each node to the broadcast of routing requests is quantified as coverage in terms of area, copies or number of neighbors. The forwarding probability is adapted to each node's coverage contribution. The different values of forwarding probabilities at neighboring sensor nodes ideally lead to a small set of nodes forwarding routing requests at any time while ensuring that each routing request can reach its destination node with high probability. The effect of probabilistic forwarding on the route established is studied through simulations. Results show that transmissions of routing requests can be further reduced with only a slight increase in routing delay.

In [91], Gossip-based approaches are utilized for group-based reliable multicast in large scale distributed applications. A reliable probabilistic multicast scheme, rpbcast, is presented. Rpbcast is a hybrid of centralized and gossip based approaches. It uses gossip as the primary retransmission mechanism and only contacts loggers if gossips fail. Rpbcast adds packet reliability guarantees to Gossip-based multicast using loggers, and in the meantime preserves the performance advantages of Gossip-based multicast. Large groups of active senders are supported using negative gossip that specifies those messages a receiver is missing instead of those messages it has received. The negative gossip allows pull-based recovery, which converges faster than push-based recovery. Rpbcast also applies hashing techniques to reduce message overhead and approximate group membership for garbage collection.

The underlying assumptions of gossip are discussed in [92], as well as how sensitive the robustness of gossip is to these assumptions. A list of five hidden assumptions are stated explicitly. Among them are "In a gossip protocol, participants gossip with one or more partners at fixed time intervals"; "There is a bound on how many updates are concurrently propagated" and "Every gossip interaction is independent of concurrent gossiping between other processes". The authors also discussed briefly how to ensure the performance advantages of Gossip in different scenarios when these assumptions are not valid.

Probabilistic forwarding is also gaining attention in the area of sensor networks for broad-

casting and routing [27][93][94][95]. In [27], a probabilistic routing algorithm, rumor routing, is presented to reduce the communication cost of delivering events to queries. In rumor routing, when a sensor node observes an event, it probabilistically generates an agent to forward the event to its neighboring sensor nodes. Similarly, when a query is generated or received at a sensor node, the sensor node forwards the query in a random direction if it does not have a route to the event. By disseminating events probabilistically to other sensors in the network, a query may reach sensors along a route to the event with less number of hops. Rumor routing, however, is only useful when the number of queries compared to the number of events is not too large or too small. The parameters in rumor routing can be adjusted to support different query to event ratios, delivery rates and route repairs.

Parametric probabilistic sensor network routing protocols apply a limited flooding strategy during route discovery [93]. The key element is that the retransmission probability for a packet at a sensor node is a function of various parameters rather than a constant. For destination attractor, a sensor closer to the destination of a message forwards with a higher retransmission probability. In contrast, for directed transmission, a sensor in the shortest path towards the destination forwards with a very high probability. The global information needed by these two schemes, the hop distance to the destination and the distance from source to the destination is estimated using a light weight message exchange protocol. It has been shown through simulations that different quality of service levels, measured as a fraction of packets delivered, can be supported by destination attractors and directed transmission, even in the presence of highly noisy network information.

Localized techniques for broadcasting in multi-hop ad hoc sensor networks are discussed in [94]. The authors present three different schemes: the Irrigator protocol, the Irrigator v2.0 scheme and the Fireworks protocol. The first two schemes are based on the idea of flooding over a sparse virtual topology, computed by means of inexpensive and fully decentralized protocols. The Fireworks protocol, instead, belongs to the class of on line probabilistic flooding. It has been shown through simulation that the three approaches can significantly decrease energy consumption and network load and increase the reliability of the broadcasting primitive over the GOSSIP protocol, resulting in promising solutions for energy constrained sensor networks.

It has also been shown that further performance improvement can be achieved for gossip by exploring network wide or local information [95]. For example, neighbor states are utilized to set the gossip probability in [95]. The simulation results show that a superior performance in terms of coverage, energy efficiency, per hop latency and overhead can be achieved. We take this approach a step further and investigate how various kinds of local neighborhood information can be explored to reduce the energy consumption for query dissemination in sensor networks.

2.3 QUERY PROCESSING IN SENSOR NETWORKS

Several sensor database query systems, such as Cougar [56] and TinyDB [57], have been developed by database researchers. These works aim to extend SQL-like systems for sensor networks by focusing on reducing power consumption during query processing. The Cougar approach to tasking sensor networks through declarative queries is introduced in [56]. A set of challenging research problems, including distributed in-network processing, query optimization, query languages, catalog management and multi-query optimization, are described and discussed as well. TinyDB, a query processor for sensor networks that incorporates acquisitional techniques, is presented in [57]. TinyDB is a distributed query processor that runs on each of the nodes in a sensor network. TinyDB has many of the features of a traditional query processor (e.g., the ability to select, join, project and aggregate data), but also incorporates a number of other features designed to minimize power consumption via acquisitional techniques. These techniques, taken in aggregate, lead to significant improvements in power consumption and increased accuracy of query results over non-acquisitional systems.

In addition to these two pioneer systems, a large number of studies have been conducted to address many other aspects of query processing techniques for sensor networks [58][8][9][59]. An energy efficient routing scheme for data collection from all nodes in a sensor network is proposed in [58]. The scheme explores suppression, both spatial and temporal, to reduce the energy cost of sensor data collection. The suppression of spatial and temporal redundancy is modeled by monitoring node and edge constraints. A monitored node triggers a report

if its value changes. A monitored edge triggers a report if the difference in values between its nodes changes. The set of reports collected at the base station is used to derive all node values. The routing scheme, constraint chaining, builds a network of constraints which are maintained locally but allow a global view of values to be maintained with minimal cost. In-network processing and data aggregation are presented in [8][9]. To answer aggregated queries such as "average" or "min", sensor data can be aggregated at intermediate sensor nodes to reduce the amount of data being transferred in the network during processing. This approach, referred to as "in-network aggregation" can significantly reduce bandwidth consumption over approaches where data is collected and aggregated centrally. The *operator placement problem*, which deals with how to place filter operators in queries at the "best" sensor node in the network based on its selectivity and cost so that the total cost of computation and communication is minimized, is addressed in [9]. It is shown that the problem is tractable; however greedy algorithms can be suboptimal. An optimal algorithm is then presented for uncorrelated filters, correlated filters and multiway stream , respectively. The work in [59] complements sensors with statistical data models to provide more meaningful query results and reduce the number of message transmissions during data collection. Models can help provide more robust interpretations of sensor reading against inaccurate or even faulty sensor readings and also extrapolate the values of missing sensors or sensors readings at geographic locations where sensors are no longer operational. Furthermore, models provide new opportunities for optimizing the acquisition of sensor readings, because sensors are only used to acquire data when the model itself is not sufficiently rich to answer the query with acceptable confidence.

Multiple query processing, in particular the optimization (MQO) problem, has been studied by database researchers [96]. The focus is given to finding common sub-expressions in a single complex query or multiple such queries run as a batch. By identifying and evaluating the common sub-expressions only once, the overall evaluation cost of multiple queries can be reduced. Greedy and heuristic search algorithms have been designed for this purpose. The focus of MQO in sensor networks, however, is different since data in sensor networks is spread over all sensors and aggregated results are usually returned to a base station during query processing. The sensor data, once aggregated, is difficult to reuse at

the base station for multiple query optimization. New schemes, therefore, should be proposed to address these new challenges.

The MQO problem has been recently addressed by several researchers [97][98][99][100]. The scheme presented at [97] explores using spatial query information for multi-query optimization. A notion equivalence classes (EC) is defined as the union of all regions covered by the same set of queries. A query is then expressed as a set of ECs intersecting with its query region. In this approach, EC becomes the unit of processing and the results from these ECs are used to derive the processing results of queries. Experimental results show that large amount of energy can be saved using this optimization technique.

The impact of MQO is analyzed in the work described at [98]. A cost model was developed to study the benefit of exploiting common subexpressions in queries. The authors also propose several optimization algorithms for both data acquisition queries and aggregation queries that intelligently rewrite multiple sensor data queries (at the base station) into "synthetic" queries to eliminate redundancy among them before they are injected into the wireless sensor network. The set of running synthetic queries is dynamically updated by the arrival of new queries as well as the termination of existing queries. A synthetic query, is rewritten from a set of queries and essentially collects data for all these queries. The idea of synthetic query works for queries collecting raw data, but not for aggregated queries. Data aggregation, such as summation, is like a one way function. From the raw data, the aggregated value can be derived, but not vice versa. Similarly, from the aggregated result for a synthetic query, the aggregated result of the queries which the synthetic query is written from cannot be derived, even if the raw sensor data of these queries are contained in the synthetic query.

The scheme is then extended into a Two-Tier Multiple Query Optimization (TTMQO) scheme [99]. The first tier, called base station optimization, adopts a cost-based approach to rewrite a set of queries into an optimized set that shares the commonalities and eliminates the redundancy among the queries in the original set. The optimized queries are then injected into the wireless sensor network. The second tier, called in-network optimization, efficiently delivers query results by taking advantage of the broadcast nature of the radio channel and sharing the sensor readings among similar queries over time and space at a finer granularity.

These proposed schemes for MQO [97][99] have explored spatial or temporal information among queries to reduce the transmission costs of multi-query processing. In this thesis, We propose to investigate a finer granularity, the semantic correlation among multiple queries, to further optimize multiple query processing in sensor networks.

The problem of "Many-to-Many aggregation" in sensor networks is addressed in [100], where destinations require data from multiple sensors while sensor data are also needed by multiple destinations. The ideas of multicast and in-network aggregation are combined to reduce communication costs. It is natural to use multicast to send source reading at sensors to multiple destinations. In the meantime, in-network aggregation is used to reduce the data communication costs for each destination. However, a sensor reading, once aggregated, becomes specific for one destination, and cannot be reused by other destinations. The goal is to determine when the in-network aggregation should be performed during multicast to minimize the overall communication costs for all destinations.

A similar problem of computing multiple aggregations in stream processing is studied in [101]. In stream processing, many users run different, but often similar, queries against the stream. Several techniques have been developed to find commonalities among aggregated queries with same or different predicates and windows. The stream is chopped into slices for aggregated queries with the same predicate but different windows. For queries with the same window but different predicates, the predicates of queries are used to divide the tuples at stream source into fragments. These tuples in fragments can be aggregated to form partial fragment aggregates, which can in turn be processed to produce the results for various queries. These two techniques are put together to process queries with different predicates and windows. The proposed approach is particularly effective in handling query update in a streaming system.

2.4 LOCATION DISCOVERY IN SENSOR NETWORKS

The localization schemes that use reference nodes can be classified into two main categories: range-based schemes and range-free schemes. Range-based schemes mainly consist of two

basic phases: distance (or angle) estimation and distance (or angle) combining. Distance estimation handles how to estimate the distance or angle between two nodes. In the distance combining phase, this information is combined to derive the locations of unresolved nodes.

The most commonly used methods in distance estimation include received signal strength indicator (RSSI), time based methods (ToA, TDoA), and the angle-of-arrival (AoA) technique [102]. RSSI measures the power of a signal at the receiver and derives the distance between the sender and receiver, based on the known transmission power and propagation model. Time based methods record the time of arrival (ToA) or time difference of arrival (TDoA) and translate it directly into the distance based on the known signal propagation speed. ToA and TDoA are used at GPS [103] and Cricket [104] for distance estimation. In the work described in [105], it is shown these location estimation problems can be solved by measuring the received signal strength from just one or two anchors in a two dimensional plane with directional antennas. If the antennas of a target sensor node are aligned, then the power received by multiple receiving antennas of the target from a single transmitting antenna on an anchor can be used to estimate the position of the target sensor node. Otherwise, received power at two different antennas of the target node from two transmitting antennas of one anchor node can be used for location discovery. The power received at antennas from two different anchor nodes using uncorrelated channels can be used to improve the location estimation accuracy. In these schemes, it is assumed that anchor nodes can talk to all other nodes in the network. Therefore, one node can estimate its location by communicating with anchor nodes directly.

Distributed positioning algorithms, in contrast, do not assume anchor nodes can talk to all other nodes in the network. In distributed positioning schemes, a small set of anchors randomly distributed over the network starts the location discovery process by communicating to immediate neighboring nodes. A sensor node, upon receiving messages from its neighboring nodes, computes the distance or angle between them and derives its own location from this information. After a node successfully estimates its own position, it becomes an anchor node and continues the location discovery process by sending messages to its neighboring nodes. The process continues until all nodes resolve their locations or no more sensor nodes can resolve their locations. In the distance combining phase, multi-lateration

techniques, such as atomic, collaborative and iterative multi-lateration, can be used to estimate sensor nodes' location [106]. The N-Hop Multi-lateration scheme [107] discusses the conditions under which one-hop, two-hop and n-hop multi-lateration can uniquely determine nodes' locations. Obviously, for successful one-hop multi-lateration, an unknown node must be neighbor to at least three nodes whose positions are known. In addition, it is necessary for an unknown node to use at least one reference point that is not collinear with the rest of its reference points in order to uniquely determine its location. Similar conditions are also defined for two-hop and n-hop multi-lateration. The Ad-hoc Positioning System (APS) [108] uses four different distance metrics, ranging from minimum hop count and sum of hop lengths to local geometric constructions to locate nodes in the network. A variant of APS utilizes angle-of-arrival of signals received from anchor nodes for location estimation [109]. These schemes rely on a high level of network connectivity so that each node can gain sufficient information in the distance combining phase for location estimation. Furthermore, a high percentage of anchor nodes must exist to achieve a small location error at each sensor node.

The schemes presented in [110][111][112] use mobile beacons whose locations are always known to help location discovery in terrestrial sensor networks. The mobile beacon nodes traverse the sensor network and disseminate their locations to other nodes in the network. A sensor node keeps track of location of and distance to the mobile beacon when it is moving. This information can then be used to derive the unknown sensors' location. Apparently, the trajectory of the mobile beacon and when the mobile beacon sends packets to other nodes are critical to the location discovery scheme. It is shown that all sensors can estimate their location as long as the trajectory of the beacon covers the entire deployment area in such a way that each point receives at least three non-collinear beacon messages [110]. In [111], a mobile user is used to collect inter-node distances between sensors and the mobile user. A movement strategy is carefully designed so that the collected distances can produce a globally rigid structure of known distances among the sensors in the network. In the perpendicular intersection scheme presented in [112], a sensor node keeps measuring received signal strength from a mobile beacon which moves in a specially designed trajectory. It is known the received signal strength depends on the distance between the sending sensor and receiving sensor, and reaches the highest value when the sender and receiver are closest to each other. When the

mobile beacon moves along a straight line, a sensor node receives the strongest signal when the line between the sensor node and mobile beacon is perpendicular with the trajectory line of the mobile beacon. By not directly mapping distance from signal strength, which usually introduces inaccuracy in distance estimation due to noise, the scheme can achieve high accuracy for location estimation in sensor networks.

Range free localization schemes, such as [113][114][115][116] do not use range or bearing information for location estimation purposes. As a result, these schemes are generally simpler than range-based schemes. However, on the other hand, these schemes only provide a coarse estimation of a sensor node's location.

DV-Hop [113] employs a classical distance vector exchange protocol to maintain a node's distance in terms of hops to all anchor nodes. The hop distance is translated into physical distance after an average distance per hop is estimated based on the hop distance and geographical distance among anchor nodes. The estimated distance to anchor nodes are then used to perform triangulation at each node for location estimation. The DV-Hop algorithm performs well only in networks that have uniform and dense node distribution.

A variant of DV-Hop, Density-aware Hop-count Localization (DHL) have been proposed to improve the accuracy of location estimation when the node distribution is not uniform [114]. DHL incorporates density of a node's neighborhood into the average hop distance estimation. Consequently, it can estimate a more accurate location of sensor nodes than DV-Hop in real deployment scenarios of sensor networks.

Area Localization Scheme (ALS) [115] locates sensor nodes into a certain area instead of an exact coordinate. Each anchor node sends out beacon signals at a set of predefined power levels. The sensors measure the lowest power level that they can receive from each anchor node. The information is then synthesized into an n-dimensional coordinate, where i^{th} coordinate represents the lowest power level from the i^{th} anchor node. The granularity of the scheme depends on the interval of power levels each anchor is configured at to broadcast its beacon signals.

Approximate Point In Triangle (APIT) [116] uses RSSI of beacon signals received from anchor nodes to determine if a sensor node is inside a given triangle. The APIT tests are carried out with all different combinations of audible anchor nodes. The test results are then

aggregated together and the location is estimated as the center of gravity of the intersections of all these triangles. This scheme requires a large level of node density to achieve a good level of accuracy of location estimation.

Several other schemes have been proposed to ensure robust location estimation against range estimation errors or erroneous reference information [117][118][119], to reduce the number of beacon signals needing to be used for location discovery [120], or to reduce location measurement error cumulation [121]. The localization problem in sparse networks has also drawn interest from researchers recently. In [122][123], the conditions for unique localization in networks are studied and used to identify all the localizable nodes in partially localizable networks to prevent flawed location estimations. Furthermore, a special class of sparse network, *bilateration network*, is investigated in [124]. The finite possible location sets of nodes are derived sequentially and some particular edges in the network are then used to sweep location possibilities. It is shown that nodes in a bilateration network can be finitely localized using the proposed scheme.

2.5 SUMMARY

This chapter presented an overview of the research related to the framework for enabling semantic views in the areas of: sensing coverage, query dissemination, multi-query processing and location discovery in sensor networks. Section 2.1 presents the current research on how to ensure a required level of coverage by sensors using a minimal amount of energy. In section 2.2, related work on disseminating queries to relevant sensor nodes in the network is discussed. The state of arts for query processing and data collection in sensor networks is detailed in section 2.3. Section 2.4 describes the currently available approaches for location discovery in sensor networks.

3.0 OVERVIEW OF THE FRAMEWORK

This chapter presents an overview of the proposed framework. The definition of semantic view is given first, followed by a detailed description of the architecture and components of the proposed framework.

The framework aims to enable energy efficient semantic views for data intensive applications in sensor networks. The inputs are user information needs, which are captured and defined as semantic views. The framework, consisting of algorithms and protocols at the base station and the sensors in the network, builds an infrastructure to efficiently identify, select and collect the relevant data from sensor nodes for processing these semantic views.

3.1 SYSTEM MODEL

3.1.1 Network Model

A sensor network consists of a base station and a set of sensor nodes. Each sensor has multiple sensing capabilities and can sense k types of data. For simplicity, sensors are assumed to have the same sensing range and to consume the same amount of energy, e_s, per sensing for the same type of data. Each sensor divides its time into epochs. During each epoch, a sensor only sample data once if needed. It is assumed that time synchronization among sensors can be achieved using schemes like [125]. The coverage requirement specifies a level of coverage required over an area in the field. It is defined per epoch and keeps changing from epoch to epoch. If a coverage requirement lasts more than one epoch, it is repeated through all the epochs in its lifetime. The coverage requirement may arrive at different times within one

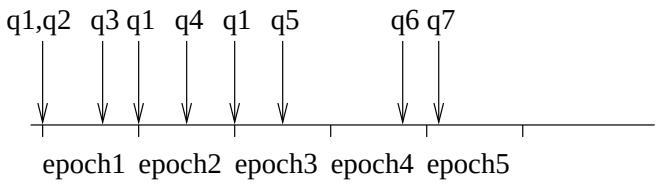

Figure 4: Query example in sensor networks

epoch. Figure 4 presents an example scenario of query arrivals.

For each type of data t, the sensing board is either "on" or "off". Initially, the sensing capabilities are turned off for all sensor nodes when they are deployed in the field in order to save energy consumption. The relevant sensing capabilities at sensor nodes are then turned on or off according to schedules made by the base station for given user queries.

3.1.2 Energy Model

A sensor consumes energy for both sensing and communication. Each sensor, once scheduled to sample the field, must consume energy for sensing and delivering the sensed data back to the base station. In addition, it may also consume energy in order to relay data from other sensors to the base station. Furthermore, to avoid collision, sensors may have to exchange messages for successful data transmissions. These issues, however, are not the focus of this research. Therefore, the energy consumption for message retransmission due to collisions is not considered in our model.

As mentioned above, each sensor consumes e_s per sensing. A sensor may consume different amounts of energy for message sending and receiving. For simplicity, in the energy model it is assumed that they are the same. Each sensor, once scheduled to sample the field, must also send the sensed data back to the base station. Therefore, it also consumes e_c per sensing. A sensor, consumes no energy if it is not scheduled to sense.

3.1.3 Semantic View Definition

Let $S = \{S_i, 1 \leq i \leq n\}$ be a set of sensors, $\forall S_i \in S$, let $D(S_i) = \{d_j^{S_i}, 1 \leq j \leq t\}$, where $d_j^{S_i}$ is the type j data sensed by sensor S_i. Furthermore, let $D = \{\bigcup_{S_i \in S} D(S_i)\}$ represent the set of data sensed by the network. Given a group of users, g, a semantic view V_g is now defined as:

Definition 1. *A semantic view* $V_g =< D_g, \Pi_g >$, *where* D_g *is a set of data and* Π_g *is a set of boolean functions.* $\Pi_g = \{P_g^i(), 1 \leq i \leq n\}$. $P_g^i : D \longrightarrow \{true, false\}$ *and* $D_g = \bigcup_{P_g \in \Pi_g} \{d | d \in D \text{ and } P_g(d) = true\}$.

Different semantic views may share common interests. The notion "Correlated" is used to define such semantic views.

Definition 2. *Given two semantic views,* V_{g_i}, *and* V_{g_j}, V_{g_i} *and* V_{g_j} *are correlated if and only if* $D_{g_i} \bigcap D_{g_j} \neq \emptyset$.

In order for sensors to understand and process semantic views, a set of queries are constructed from a semantic view. Let Q be a set of m queries, $Q = \{q_i, 1 \leq i \leq m\}$, and the data collected by a query q_i, $D(q_i) = \{d | d \in D \text{ and } d \; satisfies \; q_i\}$, the results for Q, $D(Q)$ then equals to $\bigcup_{q \in Q} D(q)$. Given a semantic view $V_g =< D_g, \Pi_g >$, a set of queries, Q_g is constructed such that $Q_g = \{q \mid \exists d \in D_g, d \in D(q)\}$ and $D_g == D(Q_g)$.

The definition of a query is given in the following section.

3.1.4 Query Definition

The following simple declarative language is used to define user queries. The language defines *variable*, *predicate* and *rule*, by which a query is defined.

Definition 3. *A variable,* V, *can be the name of a data attribute sensed by nodes in the network, location of sensors, temporal specification of data sampling or level of coverage requirement.*

Definition 4. *A predicate,* P, *is in the format of* $< V \; op \; constant >$. *"op" is the arithmetical operator,* $<, >, \leq, \geq, =,$ *or* $\neq$. *Each* P, *specifies a filter on the data to be collected.*

Definition 5. *A rule, $R = (R \wedge P) \mid\mid P$, is either a conjunction of predicates or a simple predicate.*

Definition 6. *A query, q, is in the format of $AF(V)?R_1 \vee R_2 \vee \cdots \vee R_m$. AF specifies an aggregate function on variable V, such as Max, Min, Avg. AF can be null if no aggregation is needed.*

The location variable X, Y, Z specifies a location constraint of a query. In other words, a query q is only interested at sensor data in a certain area if a location constraint is given. Otherwise, by default, a query seeks data from all sensors in the network. Another special variable in our query language is Level of Coverage, LoC. LoC enables a user to specify the desired quality of sensing. It enforces that each point in the query q's target area must be covered by at least LoC different sensors. The default level of coverage is 1 if not specified explicitly.

The temporal variable specifies the interval of query processing. Based on the value of the temporal variable, a user query can be classified into a snapshot query, which is only executed once, or a long-lived query which collects data from the sensor network repeatedly during a specified time period, T, in a specified interval.

A query, q, essentially specifies a set of filters on the sensor data to be collected, in addition to the spatial-temporal constraints and level of coverage requirement. q is eventually mapped to a set of sensor nodes, $Sensors(q)$, whose data meets all the rules in q and the other constraints.

A simple query q in a traffic sensor network can be given as :

$$\begin{aligned} WithinArea(X,Y) &= X \leq 200 \wedge X \geq 100 \wedge Y \leq 200 \wedge Y \geq 100 \\ Filter(X,Y,T) &= WithinArea(X,Y) \wedge T = 0 \\ q &= Avg(Speed)?Filter(X,Y,T) \end{aligned}$$

The rule "WithinArea" expresses that the potential interesting data are those sensors within a square area from (100,100) to (200,200). $T = 0$ indicates that this query is a snapshot query. q then shows that the data to be collected should be aggregated as the average speed.

3.2 FRAMEWORK ARCHITECTURE

The proposed framework for enabling energy efficient semantic views is presented in Figure 5. It mainly consists of four components: query aware sensing, probabilistic query dissemination, correlated multi-query processing and location discovery. The correlated multi-query processing is further divided into two parts: the correlated multi-query processing at the base station and correlated data collection at the sensor nodes. Upon receiving a user semantic view, the base station uses query aware sensing scheduling to determine what sensors should sense data for the user semantic view. It then uses correlated multi-query processing to derive a new set of queries and SIVS and delivers these queries and SIVS to sensor nodes using probabilistic dissemination. After the relevant set of sensors are decided for the semantic view, sensors use correlated data collection to send their data back to the base station. Through the whole lifetime of the sensor network, the location discovery protocol is executed at sensor nodes periodically to determine sensors locations, and the end-to-end pairwise key establishment scheme is used to provide symmetric keys between sensors for secure message exchange.

In query aware sensing, the set of active sensors is dynamically adjusted to achieve the required level of coverage for the current set of semantic views. When new queries arrive at the base station, the base station first derives the level of coverage requirement, COV, from these queries. Then based on the current sensing scheduling, it computes a minimum set of sensors which must be additionally activated in order to provide the desired level of coverage using a greedy algorithm. These sensors are then added to the current set of active sensors. When the current sensing period ends, the current set of active sensors is updated from the set of queries which still need to be processed. The sensors do not sample data and turn off their sensing boards unless they are instructed by the base station to sense, rather than they are constantly sensing.

The queries in semantic views are also used at the same time by "correlated multi-query processing" at the base station. An estimation model is used to measure the size of shared data between two queries. Based on the estimation value, pairs of queries are selected in

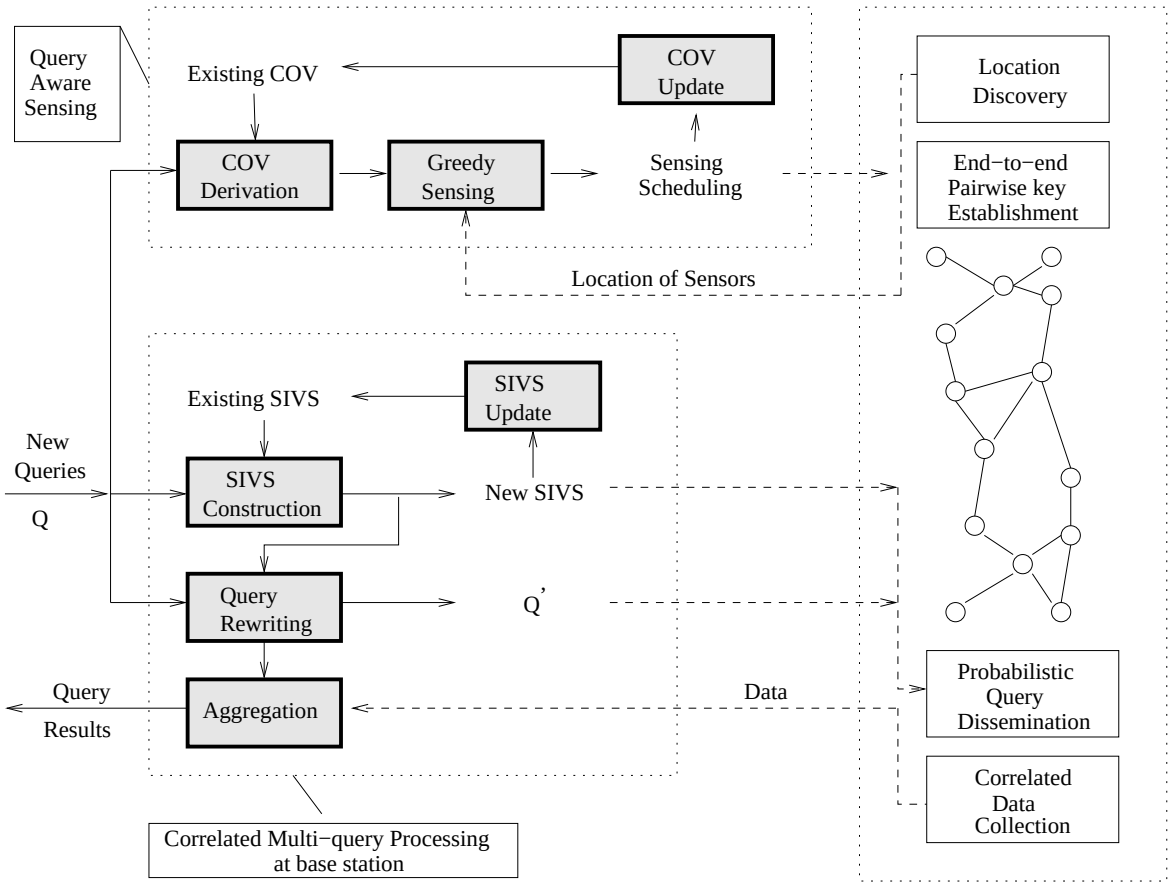

Figure 5: The framework for enabling energy efficient semantic views in sensor networks

such a way that the estimated size of shared data among all these pairs is maximal. A shared intermediate view is constructed for each pair of queries, which captures the actual set of sensor data shared by these two queries. To ensure semantic correctness, the original queries are rewritten into a different set of queries such that the data for an original query is now divided into the set of sensors for the shared intermediate view and the rewritten query. The set of shared intermediate views is also dynamically updated when new queries arrive at the base station. As in query aware sensing, the set of shared intermediate views is cleared at the end of a sensing period and rebuilt at the beginning of the next sensing period.

These shared intermediate views, along with the rewritten queries, are delivered to sensor nodes in the network using probabilistic query dissemination. In probabilistic query dissemination, each sensor forwards a query with a certain probability. This probability is adapted to each sensor node's local information, such as the additional area its forwarding can cover, or the additional number of sensor nodes its transmission can reach, or the number of messages with the same query it has already overheard.

After sensors receive the queries, they use correlated data collection to reduce the number of data transmissions for correlated queries. In correlated data collection, each sensor node stores its data to a proxy sensor node which is closer to the base station. A proxy node is established when the data at the sensor node is first acquired by a query. The node in the routing tree which first aggregates the value of a sensor node is the proxy sensor for the sensor node. After a proxy sensor has been established by a sensor node, any later query requesting its data shall retrieve the data from its proxy node. If the proxy node of a sensor fails, a new proxy node is established when the next query requests data from the sensor.

Sensors run the location discovery using Out-of-Range information with multi-lateration to compute their locations, which are used by query aware sensing to determine how to use the sensors to achieve a desired level of coverage and by sensors to check if its data is needed for queries with geographical constraints. In this scheme, some sensors are initially configured as reference nodes or anchor nodes. The scheme starts with the anchor nodes disseminating their positions to neighboring unknown sensor nodes. An unknown sensor node then measures its distance to each of the neighboring reference/anchor nodes respectively, assuming that the distance between two sensors can be estimated using methods such as

RSSI or ToA. If more than three neighbor nodes are reference nodes, an unknown node then estimates its own location using trilateration. In addition, the least square method is used to refine a sensor node's location in an over determined system. Otherwise, the unknown sensor node sends messages to non-neighboring nodes to check if they can help to resolve its location using Out-of-Range information. It is shown that the out-of-range information, i.e. when the distance between two non-neighboring sensors is larger than a certain threshold value, can be used to resolve location ambiguities in many scenarios. Once the unknown sensor's location is resolved, an unknown node becomes a reference node and disseminates its position to other unknown nodes in the network to enable the continuation of the location discovery process.

Furthermore, an end-to-end pairwise key establishment scheme based on key pre-distribution is used to set up symmetric keys between sensors. The scheme enhance the security of path keys by using multiple secure paths during key establishment. These symmetric keys are then used to protect message exchanges in semantic view processing against packet eavesdropping and traffic analysis by attackers.

4.0 QUERY AWARE SENSING

This chapter discusses the query aware sensing component in the proposed framework. An energy model and network model are presented at the beginning to form the basis of discussion. The level of coverage requirement is derived from user semantic views and the sensing problem is formulated as an integer programming problem. A heuristic based greedy algorithm, referred to as "GRASS", is then presented to compute a minimum set of active sensors at a specific time period. The dynamic update of the active set of sensors is also discussed as new queries may be added to the network and old queries may leave.

4.1 PROBLEM STATEMENT

The main idea of sensor scheduling is to reduce energy consumption of sensor nodes by turning off some sensor nodes' sensing capability under the condition that the remaining sensor nodes still provide the desired level of coverage. In previous studies of sensor scheduling, it has been assumed that a fixed level of coverage is required over the sensor deployment field. However, given a group of diversified user interests, the level of coverage may vary from one area to another in the field. Furthermore, the user interests may also change from time to time. The level of coverage, therefore, may also vary over time. A fixed high level of coverage can provide a high quality of sensed data. However, it might be over-provisioning when no interesting event is occurring in the network. On the other hand, if some event does occur and the fixed level of coverage in the area is low, the quality of sensed data in the area of focus will be low. Therefore, the level of coverage should be adapted to user queries and the sensor nodes should be scheduled thereafter to achieve the desired level of coverage.

Let n be the number of sensors in the network, t be the number of types of data a sensor can sense, c_j be the energy consumption per sample of data of type j, e_i be the residual energy of sensor i and $SA(i)$ be the sensing area of sensor i. A set of m queries, $Q = \{q_1, q_2, \ldots, q_m\}$, defines a sensing coverage requirement over the deployed field based on the constraints specified in the queries. Let COV_q^j be the coverage requirement of query q over data type j. Each query, q, specifies a targeted area, A, and a Level of Coverage, LoC. A is the area to be sensed for the current query, and LoC enforces that each point in A must be covered by at least LoC different sensors in order to provide the desired quality of sensed data. COV_q^j is, then expressed as a tuple (A_q^j, LoC_q^j). The problem is to determine a sensing scheduling to minimize:

$$\sum_{i=1}^{n}\sum_{j=1}^{t}(X_{ij} \times c_j) \tag{4.1}$$

Where $X_{ij} = 1$ if sensor i is selected to sense data type j. Otherwise, $X_{ij} = 0$. The schedule must satisfy the following constraints:

$$\forall q \in Q, \forall j, 1 \leq j \leq t, \forall P \in A_q^j, \quad \sum_{i=1}^{n} X_{ij} \times (P \in SA(i)) \geq LoC_q^j \tag{4.2}$$

and

$$\forall i, 1 \leq i \leq n, \quad e_i - \sum_{j=1}^{k} X_{ij} \times c_j \geq e_{threshold} \tag{4.3}$$

Constraint 4.2 requires that the desired coverage requirement of each query must be satisfied by the current scheduling of sensing. Constraint 4.3 enforces that each sensor configured to sense data for the current set of queries has at least the amount of $e_{threshold}$ energy left after data sampling. The $e_{threshold}$ must be large enough to allow the current sampled data to be transmitted back to the base station. Otherwise, the sampled data would be lost and the level of coverage requirement would not be met.

Constraint 4.2 can be simplified into the following:

$$\forall j, 1 \leq j \leq t, \forall P, \quad \sum_{i=1}^{n} X_{ij} \times (P \in SA(i)) \geq LoC_P^j \tag{4.4}$$

Where LoC_P^j, the level of coverage at a point P, is defined as follows:

$$LoC_P^j = \max_{\forall q \in Q, P \in A_q^j} LoC_q^j \tag{4.5}$$

Constraint 4.4 can be further transformed into a list of constraints if an area is approximated into a set of points. The maximum number of constraints from 4.4 is $t \times NumberOfPoints$, in which the $NumberOfPoints$ depends on the granularity of approximation. This simplification transforms the sensing scheduling problem into an integer programming problem, which is known to be NP-Hard. In the following section, a greedy algorithm for query aware sensing scheduling is presented.

4.2 GRASS: A GREEDY ALGORITHM FOR SENSING SCHEDULING

In GRASS, the level of coverage requirement for each type of data j is transformed into $COV(j) = \{(A_1, LoC_1), (A_2, LoC_2), \cdots\}$, where $\forall (A_{i1}, LoC_{i1}), (A_{i2}, LoC_{i2}) \in COV(j)$, $A_{i1} \cap A_{i2} = \emptyset$. The rest of GRASS is an iteration-based algorithm. During each iteration, a sensor node computes a weight, w, from $COV(j)$ and its own sensing area. The weight w measures how much coverage each sensor node contributes if it is scheduled to sense data. The sensor with the highest weight is then selected during the current iteration. The coverage requirement $COV(j)$ is updated when a sensor i is scheduled to sense data j. One sensor node is selected during each iteration and the whole process continues until $COV(j)$ becomes empty or all nodes have been selected. The overall steps of the greedy algorithm are presented in Algorithm 1. The main steps COV derivation, weight computation and COV update are further explained in the following sections.

4.2.1 COV Derivation

Each query q in Q may specify a coverage requirement, (A_q^j, LoC_q^j), for each type of data j to be sensed. A_q^j may overlap with each other in Q. To better explain the level of coverage

Algorithm 1 GRASS: A GReedy Algorithm for Sensing Scheduling

INPUT:

a set of queries, $Q = q_1, q_2, \cdots, q_m$

INITIALIZATION

$\forall i, j,\ X_i^j = 0$

Derive the level of coverage requirement, $COV(j)$, from Q

while ($COV(j)$ is not empty) **do**

Compute weight, w of each sensor, S_i, based on its sensing area, SA, and current $COV(j)$

Pick the sensor, i, with the largest weight, w, set $X_i^j = 1$

Update $COV(j)$ according to sensor i's sensing area, SA_i

end while

OUTPUT:

X_i^j for each sensor i

requirement, the whole deployed area, A, is divided into disjoint subareas and the coverage requirement of each subarea in defined a way similar to Equation 4.5.

$$LoC_A^j = \max_{\forall q \in Q,\ A \subseteq A_q^j} LoC_q^j \tag{4.6}$$

Given Equation 4.6, it is straightforward to derive the level of coverage from Q. The algorithm is presented in Algorithm 2.

4.2.2 Weight Computation

The definition of weight is critical to the sensing scheduling algorithm. It determines which nodes are selected during each iteration. Intuitively, a simple heuristic towards selecting a minimal number of sensor nodes is to choose nodes which cover the most area left to be covered. However, coverage of points in an area is shown to be correlated to each other in [126]. In [126], the area to be covered is transformed into a planar graph $G = (V, E)$, with vertices V corresponding to the intersection points of the boundaries of all sensors' coverage

Algorithm 2 COV derivation

Initialization
  $COV = \emptyset$
for all $(q \in Q)$ **do**
  $A_{LEFT} = A_q^j$
  for all $(A, LoC) \in COV$ **do**
    if $LoC_q^j \geq LoC$ and $A_{LEFT} \cap A \neq \emptyset$ **then**
      $COV = COV - \{(A, LoC)\} + \{(A_{LEFT} \cap A, LoC_q^j), (A - A_{LEFT}, LoC)\}$
      $A_{LEFT} = A_{LEFT} - A$
    end if
  end for
  if $A_{LEFT} \neq \emptyset$ **then**
    $COV = COV + \{(A_{LEFT}, LoC_q^j)\}$
  end if
end for

regions and edges E connect pairs of adjacent intersection points along the boundaries of sensor coverage circles. Figure 6(b) presents an example planar graph constructed from the coverage requirement described in Figure 6(a). The main steps for constructing a planar graph from the sensor locations and sensing ranges are presented in Algorithm 3.

Once the planar graph, G is constructed, it is easy to find all the faces by walking through the edges of G. In a planar graph, a face can be identified by a directed edge and its orientation. Furthermore, an edge also belongs to two faces if the outer face is included. Giving these properties, a face can be found by starting from one edge and continuing at the next node with the edge which is not included in two faces yet and has the largest clockwise degree. The continual traversal of edges in this way can yield to a new face found. From Euler's formula, the number of faces in a planar graph G, $f = e - v + 2$, in which v is the number of vertices and e is the number of edges in G [127]. Furthermore, it is known that if $v \geq 3$, then $e \leq 3v - 6$ [127]. In the planar graph constructed from a network with n sensors,

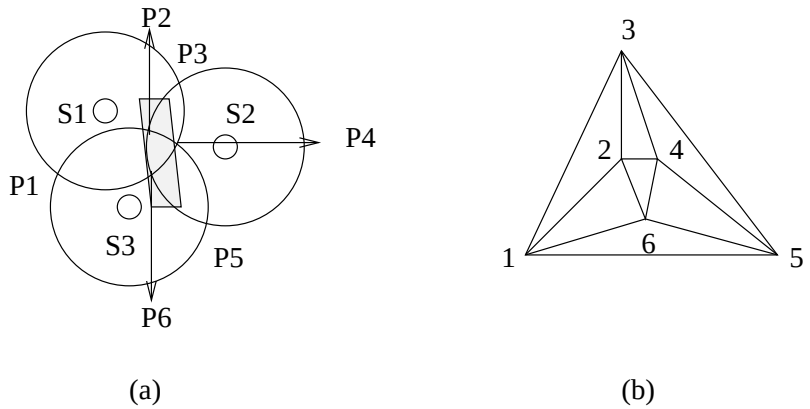

Figure 6: Planar graph from area coverage

Algorithm 3 Planar Graph Construction

INPUT:
  a set of sensors, S,
for all sensor $s_i, s_j \in S$ **do**
  if the sensing circle of s_i, s_j intersects **then**
    add two vertices, V_{ij}, V_{ji} to V
  end if
end for
for all $V_{ij} \in V$ **do**
  find all $V_{kl} \in V$, such that $k == i || l == i$
  from these nodes, find the node(s) in the sensing circle of s_i, which is closest to where s_i, s_j intersects
  add edges between V_{ij} and these closest nodes to E
end for
OUTPUT:
  planar graph $G = (V, E)$

there are at most $n(n-1)$ vertices. As a result, there are at most $2n(n-1)-4$ faces in the graph. Once the faces are identified, each sensor's coverage area is now represented as a set of faces that the intersection point of each vertex of these faces are within the sensor's sensing area.

It is easy to see that for each face of G, i.e. the parts of the plane bounded by edges, if at least one internal point is covered by a sensor, then the entire face is also covered by the same sensor. As a result, a face shall be treated as the unit of coverage. Based on this observation, a sensor which covers the most faces shall be chosen in order to minimize the number of nodes for a coverage requirement. This heuristic leads to the definition of weight as Equation 4.7.

$$w_i = \sum_{\forall(A,LoC)} Number\ of\ faces\ in\ SA(i) \cap A \tag{4.7}$$

4.2.2.1 Faces Over Areas

It seems to be intuitive, simple and may also effective to select sensors with larger sensing coverage area during sensing scheduling. However, as we pointed out above, a face is the unit of coverage and it is more accurate to use faces to measure the weight of coverage than using areas because a sensor covers a larger area doesn't necessarily provide more coverage in terms of sensing.

Take the coverage requirement described in Figure 6(a) for an example, if coverage area is used to select sensors for scheduling in the greedy algorithm, sensor S2 would be selected first and then the other two sensors must both be selected in order to cover the area. In contrast, if faces are used, S1 covers {f123, f234, f246, f126}, S2 covers {f234, f246, f456}, and S3 covers {f126, f246, f456, f156} for the given area. At first, either S1 or S3 would be selected since they both cover four faces, one more face than how many faces S2 can cover even though S2 covers more area. As a result, only two sensor are needed to cover the area.

4.2.3 COV Update

When a node i is selected to sense data j, it provides single coverage for all the queries interested in data in its sensing area, $SA(j)$. The level of coverage requirement set, COV, therefore must be updated to reflect such coverage. The main steps of the update are presented in Algorithm 4.

Algorithm 4 COV update

1: $A_{LEFT} = SA(i)$
2: **for all** $(A, LoC) \in COV$ **do**
3: **if** $A_{LEFT} \cap A \neq \emptyset$ **then**
4: $COV = COV - \{(A, LoC)\} + \{(A_{LEFT} \cap A, LoC - 1), (A - A_{LEFT}, LoC)\}$
5: $A_{LEFT} = A_{LEFT} - A$
6: **end if**
7: **end for**
8: **for all** $(A, LoC) \in COV$ **do**
9: **if** $A == \emptyset$ or $LoC \leq 0$ **then**
10: $COV = COV - \{(A, LoC)\}$
11: **end if**
12: **end for**

4.2.4 Incremental Scheduling

At the beginning of each epoch, the base station uses the greedy algorithm to determine the set of sensors to sense in this epoch based on the currently known coverage requirements. When a new coverage requirement arrives later during this epoch, the areas covered by current set of active sensor nodes are first deducted from the coverage requirement using Algorithm 4 and additional nodes are scheduled to sense using the greedy algorithm.

At sensor nodes, the sensing boards are turned off at the beginning of an epoch to conserve energy. Upon receiving a sensing schedule from the base station, a sensor turns on its sensor board and senses data. After the sensing operation is finished, a sensor turns off its sensor board for the rest of time during the epoch.

A sensor may also turn off its radio to save energy. In order to receive the sensing schedules from base station when the radio is off, a sensor must periodically wake up within an epoch to check messages. These periodic checks, however, can be eliminated if a sensor knows that it must wake up and sense in next epoch. This is possible since some queries are periodic and repeat themselves over several epochs. In these cases, the base station knows what queries will be executed in future epochs and can determine a scheduling that may last for several consecutive epochs.

4.3 ANALYSIS

In this section, we develop an analytic bound on the approximation ratio of the proposed algorithm. Using Algorithm 2, the coverage requirement can be rewritten into an equivalent set of coverage of disjoint areas $SC = \{< A_1, LoC_1 > \cdots < A_m, LoC_m >\}$. For each area A_i, a planar graph is defined, and a set of faces $F_i = \{f_1, f_2, \cdots, f_{m_i}\}$ is constructed. The monitored area, A_i, is then represented as a set of faces $A_i = \{f | f \cap A_i \neq \emptyset\}$. Each sensor i is represented as a set of faces $S_i = \{f | f \subseteq SA(i)\}$, in which each S_i is a subset of A_i. The coverage problem then is equivalent to select a minimum number of subsets from $\{S_1, S_2, \cdots, S_n\}$, such that:

$$\forall f \in A_i, \ (\sum_{1 \leq i \leq n} f \in S_i) \geq LoC_i \tag{4.8}$$

This problem is a variant of the set covering problem and is referred to as the set k covering problem since each element must now be covered k times. In comparison, each element only needs to be covered once in the original set covering problem. The set k covering problem is formulated as follows. Given a universe $U = \{a_1, a_2, \cdots, a_m\}$, a set $S = \{S_1, S_2, \cdots, S_l\}$ where $\forall 1 \leq i \leq l, \ S_i \subseteq U$, and an integer number, k, find a minimum cardinality $J \subseteq \{1, 2, \cdots, l\}$ such that

$$\forall x \in U, (\sum_{i \in J} x \in S_i) \geq k \tag{4.9}$$

The set k covering problem can be approximated within a factor of H_{mk} in equation 4.10 using the proposed greedy algorithm. The details of the proof can be found in Appendix A.1.

$$H_{mk} = \sum_{1 \leq i \leq mk} \frac{1}{i} \approx \ln(mk) \tag{4.10}$$

With respect to the number of faces, m, in the planar graph for an area A_i, $m \leq n_i \times (n_i - 1) + 2$, where n_i is the number of sensors which covers a subarea in A [126]. Therefore, for a single coverage $< A_i, LoC_i >$, the approximation ratio is $H_{(n_i \times (n_i - 1) + 2) \times LoC_i}$. Similarly, an approximation ratio for a set of coverage $SC = \{< A_1, LoC_1 >, \cdots, < A_m, LoC_m >\}$ can be derived as:

$$\begin{aligned} & H_{\sum_{1 \leq i \leq m} LoC_i \times (n_i \times (n_i - 1) + 2)} \\ & \leq \ln \sum_{1 \leq i \leq m} LoC_i + 2 \ln n \end{aligned} \tag{4.11}$$

4.4 SIMULATION RESULTS

4.4.1 Methodology

In the simulations, two metrics are used to evaluate the performance of the sensing scheduling schemes: the average energy consumption and network lifetime. These metrics depend on the energy consumption for communications and sensing. We started from a model that the ratio of communicating over sensing energy consumption as 10 to 1, which is derived from actual measurement of energy consumption of sensors [2]. To study the performance of our scheme over other energy models, we also collected results when a different ratio 5:1 is used.

We simulated two levels of coverage by varying the number of sensors in a field of 50X50m. In the case of 150 sensors deployed using uniformly random distribution, the average level of coverage, i.e the average number of sensors covering one point in the field, is 10.3. The value increases to 17 when 200 sensors are deployed on the same field. These values may seem to be very high, and indicates that the whole area is very densely covered. However, both cases can only guarantee each point in the area to be covered by at least 2 sensors,

since the edges of the field are much sparsely covered than other areas. In the simulation, the network of 150 sensors is referred as "moderately covered field", while the network of 200 sensors is referred as "densely covered field".

Table 1: Two types of field in the simulation

	moderately covered field	densely covered field
Number of sensors	150	200
Average coverage level	10.3	17

The sensing scheduling scheme adapts the sensors to the dynamic coverage requirement at each epoch. Apparently, the performance of the scheme also depends on the level of coverage required. During the simulation, the *coverage load* is used to measure how much coverage is required for each simulation. It is defined as follows:

$$Coverage\ load = \frac{\sum_{<A,LoC>\in SC} A * LoC}{\Lambda \times \max(LoC)} \tag{4.12}$$

In order to deliver the sensed data back to the base station, a sensor may need to exchange messages with other sensors to maintain a routing tree and a contention free channel for message transmission. In the simulation, this amount of energy consumption is not considered since it must be consumed by any data collection scheme and is not related with sensing scheduling. For simplicity, it is assumed each sensor only needs to send one message for each sensed data. Therefore, one sensing operation consumes energy for one data transmission in addition to the cost of one sensing.

4.4.2 Performance Comparison

The energy consumption of GRASS using faces is compared with GRASS using areas and the optimal sensing scheduling for a static level k coverage. k is the maximum level of coverage provided in the network and it equals to 2 in the simulations. The "Static-2" scheduling selects a minimum set of sensors as follows. The area is approximated into a set of points, SP, and each sensor is represented as a subset of points in SP. The coverage problem is

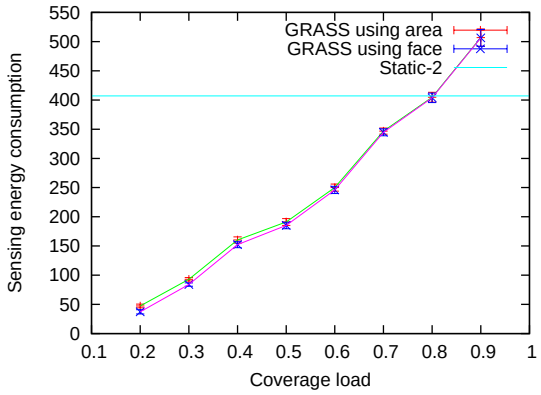

Figure 7: Average sensing energy consumption in a moderately covered field

then equivalent to select a set of sensors such that each point in SP appears in the union of points of selected sensors at least twice. This problem can be formulated as an integer programming problem, which in general cannot be solved in polynomial time. But when the number of points is small, the minimum set of sensors can be computed within a small amount of time using GLPK [128].

It is easy to see that a static schedule can guarantee that no matter how the coverage requirement changes, the sensing schedule can always provide the required level of coverage in the field, under the cost that many unneeded sensors are being turned on and sensing during each epoch. Figure **??** and Figure **??** present the average sensing energy consumption for a single set of coverage requirement in modestly and densely covered field, respectively. In these two figures, each point represents an average value over 30 different sets of coverage requirement with the same coverage load. The confidence interval of each single piece of data with a 95% confidence level is also presented.

As expected, GRASS using faces and GRASS using areas can both save save a significant amount of energy consumption from sensing when the coverage load is small by adapting the sensing schedules to the dynamic level of sensing coverage. The saving decreases as the coverage load increases. When the coverage load is high, the static coverage actually consumes a smaller amount of energy than the proposed scheme since the schedule is optimized.

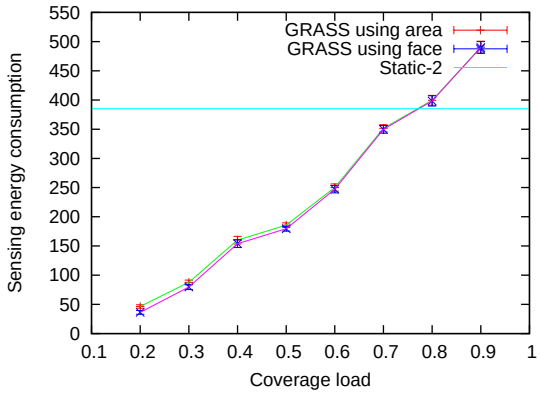

Figure 8: Average sensing energy consumption in a densely covered field

Therefore, when the coverage load is very high throughout the network lifetime, query aware sensing should not be used. Instead, optimum static coverage should be used.

Also as expected, GRASS using faces consumes slightly less energy than using areas, since it is more accurate to measure the sensing coverage contribution using faces than areas. Furthermore, GRASS using faces can provide an approximation ratio over the optimal sensing scheduling, while GRASS using areas cannot. Therefore, it is preferred to use faces in GRASS than using areas.

The second set of simulation studies the sensing lifetime of GRASS using faces, GRASS using areas and "static-2" coverage. The sensing lifetime is defined as the number of epochs until the coverage of the field breaks. During the measurement of a sensing lifetime, a set of coverage requirement with the same coverage load is generated during each epoch. Figure 9 and Figure 10 present the average sensing lifetime in modestly and densely covered field, respectively. In these two figures, each point is an average value of 20 runs of lifetime measurement with the same coverage load.

Both Figure 9 and Figure 10 show that, when the coverage load is low, the GRASS using faces and GRASS using areas can both achieve a much higher lifetime than the static-2 scheduling. On the other hand, when the coverage load is very high, static-2 coverage can provide longer lifetime than GRASS schemes.

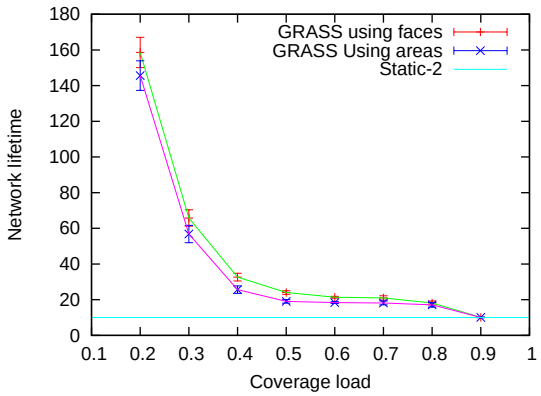

Figure 9: Average sensing lifetime in a moderately covered field

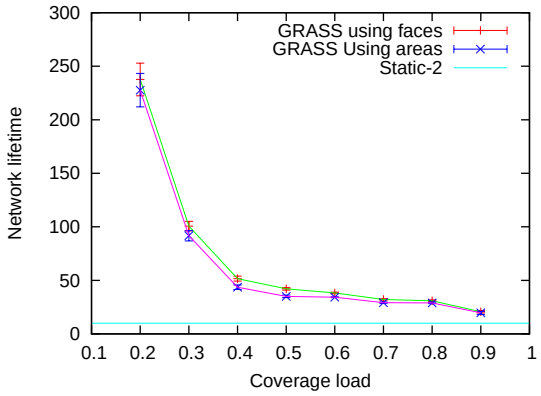

Figure 10: Average sensing lifetime in a densely covered field

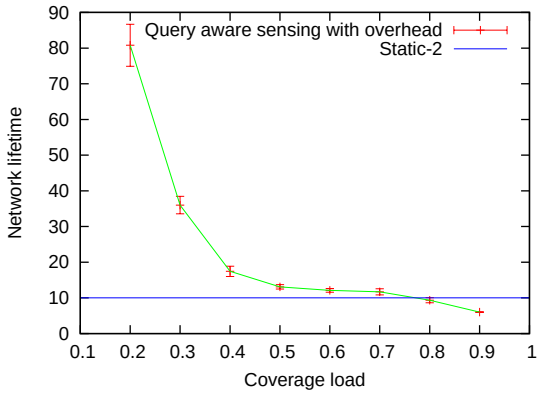

Figure 11: Average sensing lifetime with communication overhead in a moderately covered field

4.4.3 Communication Overhead

In the proposed scheme, the sensing schedule is computed at the base station and must be delivered from the base station to relevant sensors during each epoch. These messages then incur an additional amount of communication overhead. Therefore, the energy saving of sensing by the proposed scheme might be reduced by the amount of energy consumption necessary to send the schedules. However, there are many ways to reduce the communication overhead. For example, the sensing schedule can be combined with any broadcast message from the base station to sensor nodes within an epoch, such as a data request from the base station.

If a separate message must be sent from the base station to relevant sensors, an additional amount of 10 units for receiving the sensing scheduling needs to be consumed at each sensor per sensing in our scheme. Figure 11 and 12 present the results for the average network lifetime when such communication overhead is considered in a network of 150 and 200 sensors, respectively. The results show that the proposed scheme can still extend the network lifetime when coverage load is low. Again, the network lifetime becomes shorter than the static-2 scheduling in the 150 nodes network when coverage load is 0.8 and 0.9.

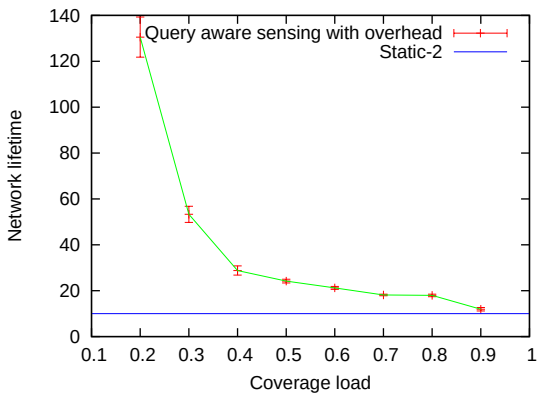

Figure 12: Average sensing lifetime with communication overhead in a densely covered field

4.4.4 Ratio of Communicating over Sensing Energy Consumption

Figure 13 and 14 compares the average sensing lifetime for two different ratios of communicating over sensing energy consumptions, 10 to 1 and 5 to 1, in a moderately and densely covered field, respectively. Our scheme mainly saves energy by eliminating unneeded sensors from sensing while still providing the required level of coverage. A lower ratio of communicating over sensing energy consumption means saving in sensing energy consumption can be used for more data transmissions. As a result, the sensing lifetime is expected to be bigger, as shown from the results in Figure 13 and 14.

The difference in lifetime decreases as the coverage load increases since the energy saving also decreases. The same pattern can be observed from the results presented in Figure 15 and 16, which takes the communication overhead per sensing scheduling into consideration. In short, our scheme is more effective if the ratio of communicating over sensing energy consumption is lower.

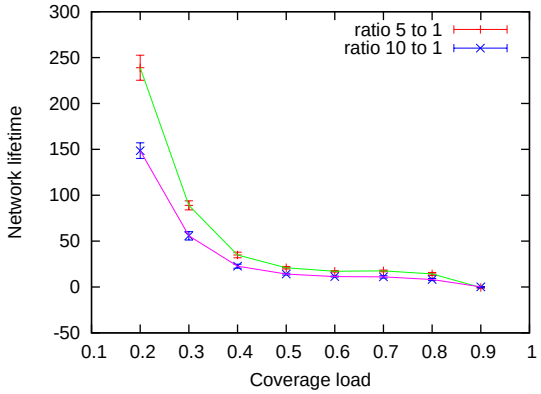

Figure 13: Average sensing lifetime for two ratios of communication over sensing energy consumption in a moderately covered field

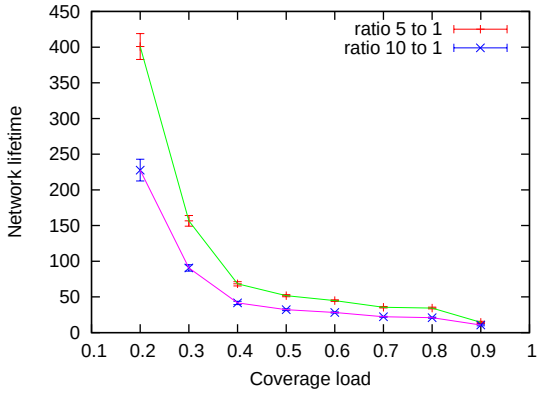

Figure 14: Average sensing lifetime for two ratios of communication over sensing energy consumption in a densely covered field

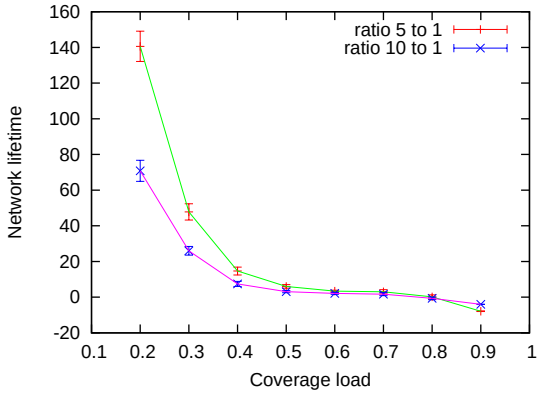

Figure 15: Average sensing lifetime with communication overhead for two ratios of communication over sensing energy consumption in a moderately covered field

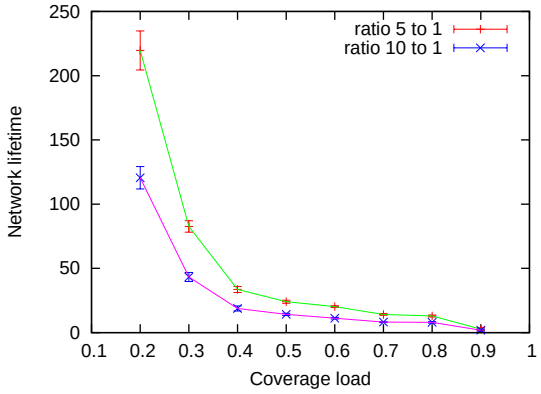

Figure 16: Average sensing lifetime with communication overhead for two ratios of communication over sensing energy consumption in a densely covered field

4.5 SUMMARY

In this chapter, the query aware sensing component in the framework is presented in detail. In query aware sensing, the coverage requirement is derived from semantic views and the sensing coverage problem is transformed into an integer programming problem. A greedy based heuristic algorithm is then developed to select a minimum set of working sensors to sample data requested by semantic views. The simulation results show that various amount of energy can be saved unless the required level of coverage on the field is extremely high.

5.0 PROBABILISTIC QUERY DISSEMINATION

This chapter describes the probabilistic query dissemination of queries and shared intermediate views to relevant sensors in the network. To reduce the query propagation cost, the probability p of nodes forwarding a query q is adapted to various types of local topology information, i.e. transmission area and neighborhood information. Four schemes, namely Area Coverage-based Probabilistic Forwarding (ACPF), Copies Coverage-based Probabilistic Forwarding (CCPF), Area and Copies Coverage-based Probabilistic Forwarding (ACCPF) and Neighbor Coverage-based Probabilistic Forwarding (NCPF) are presented and discussed in the chapter. A remedy process is also presented to send queries to these sensors which do not receive queries after the probabilistic forwarding process.

5.1 PROBLEM STATEMENT

Given a set of queries, $Q = \{q_1, q_2, \ldots, q_m\}$, if query q wants to collect data from area A_q^t, the potential relevant nodes are a subset of nodes $\{N|Loc(N) \in A_q^t\}$. Otherwise, the potential relevant nodes are all the nodes in the network. A node determines if its data should be collected for query q only after the data is sampled. In other words, the query must be delivered to all potential relevant nodes in order for the query to be processed in the network.

The dissemination of query q can be modeled as a directed propagation graph $G(q)$. $G(q)$ is a subgraph of the network. All potential relevant nodes of query q must be connected to the base station node in the propagation graph. The communication cost of a node N in the propagation graph consists of the energy to receive q and the energy of forwarding

q to other nodes in the network. Let n_r^q be the number of copies of q node N receives and n_s^q be the number of copies of q node N forwards. For each edge $e =< N_i, N_j >$ in $G(q)$, $(n_i)_s^q = (n_i)_s^q + 1$, and $(n_j)_r^q = (n_j)_r^q + 1$. Assuming the communication cost of sending a query is e_s and that of receiving a query is e_r, the total communication cost at node N for propagating q is:

$$e_r * n_r^q + e_s * n_s^q \tag{5.1}$$

The communication cost of the propagation graph $G(q)$ of query q, therefore, is:

$$\sum_{N \in G(q)} e_r * n_r^q + e_s * n_s^q \tag{5.2}$$

Given 5.2, it is easy to know that the communication cost of propagating a set of queries Q to their potential relevant nodes is:

$$\sum_{q \in Q} \sum_{N \in G(q)} e_r * n_r^q + e_s * n_s^q \tag{5.3}$$

Any redundant query propagation will increase the cost in Equation 5.3. The challenge, therefore, is to reduce redundant query transmissions as much as possible, while delivering the query request to all relevant sensor nodes.

5.1.1 Probabilistic Forwarding

Probabilistic forwarding was introduced to reduce the communication overhead of broadcast in ad hoc and sensor networks [88][94]. In the basic Gossip scheme, each intermediate sensor rebroadcasts a broadcast message with a probability p. Obviously, when p equals 1, the scheme reverts back to flooding, while when the value of p is less than 1, the average amount of flooding traffic is reduced by a fraction of $(1 - p)$. Apparently, the value of the parameter p is critical to the efficiency of the probabilistic forwarding scheme. The lower p is, the fewer messages are broadcasted but also the smaller fraction of sensors can be reached after probabilistic forwarding. In this chapter, we show how to adapt the probability p to local

topology of each sensor node to reach a larger number of sensors with a smaller number of transmissions.

The first scheme, referred to as Area Coverage-based Probabilistic Forwarding (ACPF), exploits the overlapping of transmission areas between neighboring nodes to determine the value of p a sensor uses to forward a query. Based on this scheme, this value of p depends on the extra coverage achieved by a rebroadcast of the query message— the larger the coverage, the higher the value of p. The second scheme, referred to as Copies Coverage-based Probabilistic Forwarding (CCPF), takes a different approach and uses the number of duplicate query messages overheard during a random time interval to determine its forwarding probability, p. As the number of the overheard duplicate query messages increases, the forwarding probability of the sensor decreases. The third scheme, referred to as Area and Copies Coverage-based Probabilistic Forwarding (ACCPF), takes advantage of both the transmission area coverage and the number of the overheard duplicates of the same query to determine the value of p. The last scheme, referred to as Neighbor Coverage-based Probabilistic Forwarding (NCPF), eliminates unnecessary rebroadcast by maintaining neighboring information at each sensor node. In this scheme, a sensor, which finds that all of its neighbors have been covered by other nodes' rebroadcast, refrains itself from forwarding the query message.

These four schemes, when combined with m-technique, become ACPFM, CCPFM, ACCPFM and NCPFM. The m-technique is that a sensor rebroadcasts a query message if it does not forward the query message at first and then overhead fewer than m copies of the same queries afterwards within a certain period of time. This technique, although simple, can be used to increase the number of sensors to reach for the probabilistic forwarding schemes.

5.2 AREA COVERAGE-BASED PROBABILISTIC FORWARDING

The ACPF exploits the fact that the overlap between a sensor's own coverage area and its neighbors' coverage areas is critical in reducing the number of duplicate query propagation within a neighborhood. In a wireless sensor network, the coverage areas of neighboring sensor

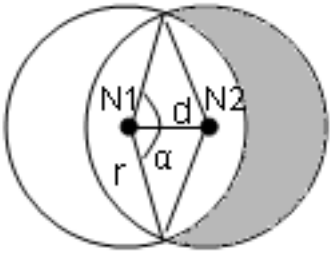

Figure 17: Extra area that node N2 can cover by rebroadcast

nodes typically overlap. Furthermore, it is usually the case that the additional area covered by a sensor's rebroadcast, after receiving a message from its neighbor, is a small fraction of the whole node's coverage area, as depicted by Figure 17. In this figure, the shaded area represents the extra area covered by $N2$'s rebroadcast after $N1$'s broadcast.

Let d denote the distance between sensors $N1$ and $N2$, r denote the transmission range of a sensor node, and $CA(N)$ denote the physical area covered by sensor N's transmission. The shaded area $E(N2, N1) = CA(N2) - CA(N2) \cap CA(N1)$ can be computed as follows. The maximal value of $E(N2, N1)$, denoted as E_{max}, is achieved when $d = r$. Its value is $0.61\pi r^2$.

$$\begin{aligned} E(N2, N1) &= \pi r^2 - 2 \times (\tfrac{\alpha}{360}\pi r^2 - \tfrac{d}{2}\sqrt{r^2 - \tfrac{d^2}{4}}) \\ &= \pi r^2 + d\sqrt{r^2 - \tfrac{d^2}{4}} - \tfrac{\arccos \frac{d}{2r}}{90}\pi r^2 \end{aligned} \tag{5.4}$$

The objective of the ACPF protocol is to reduce the number of unnecessary query propagations, while maintaining the likelihood of query delivery to all relevant sensors. To achieve this objective, the ACPF protocol exhibits the following properties:

- The forwarding probability increases with the extra coverage the node adds to the area covered by the original request

- The forwarding probability reflects the passivity of a node in participating in the process. This property can be expressed, for example, in terms of the residual energy of a node, whereby a node's passivity increases as its energy depletes. Other criteria can be used to define the passivity of a node

The main steps of the ACPF algorithm are depicted in Algorithm 5, where p represents the basic gossiping probability, and $0 \leq k \leq 1$ represents the node's passivity. Upon receiving a query, q, from $N1$, node $N2$ computes the extra coverage area with respect to $N1$'s coverage area and uses it to compute its forwarding probability, p'.

Algorithm 5 ACPF

1: Compute extra coverage area: $E(N2, N1) = \pi r^2 + d\sqrt{r^2 - \frac{d^2}{4}} - \frac{\arccos \frac{d}{2r}}{90}\pi r^2$
2: Derive a new probability p' as follows: $p' = p \times e^{k \times \frac{E(N2,N1) - E_{max}}{\pi r^2}}$
3: Forward q with probability p'

Notice that as k increases, the forwarding probability, p', decreases, further reducing the amount of overhead caused by unnecessary propagation of query messages. A large value of k, however, may increase the likelihood that a query message "dies" before it reaches its destination. Figure 18 shows the impact of k on p' and highlights the need for a careful consideration of the passivity parameter in order to increase the likelihood of query delivery.

5.3 COPIES COVERAGE-BASED PROBABILISTIC FORWARDING

ACPF relates the forwarding probability to the "gain" achieved by a rebroadcast over the additional coverage area. It is clear, however, that this gain may not be significant if the neighboring nodes are within the same distance from the sending node. This can be illustrated by the scenario depicted in Figure 19. In this scenario, $N1$, $N2$ and $N4$ receive a query message from $N3$. Using ACPF, $N1$, $N2$ and $N4$ are likely to produce similar values for p'. This is due to the fact that $d(N1, N3)$, $d(N2, N3)$ and $d(N4, N3)$ are close to r. As a result, all three nodes may end up rebroadcasting query q, thereby making $N1$'s rebroadcast unnecessary, since its area is covered by node $N2$, $N3$ and $N4$. CCPF addresses this

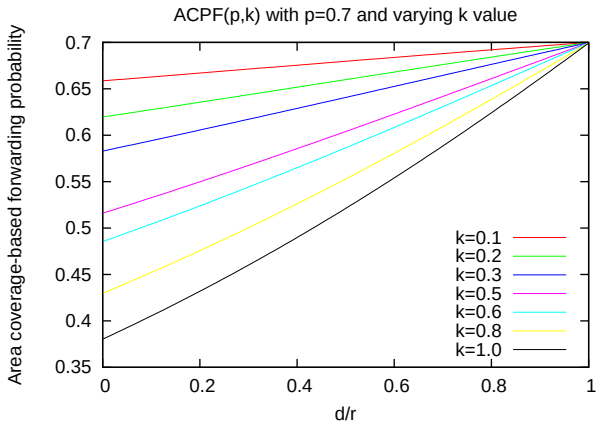

Figure 18: Effect of node passivity, k, on ACPF forwarding probability

shortcoming by taking into consideration the number of rebroadcasts of the query q a node hears within its neighborhood.

The basic steps of the CCPF algorithm are depicted in Algorithm 6. Based on this algorithm, a sensor, N, which receives a query, q, listens for a random time interval. During this period, the node counts the number of rebroadcasts of q by its neighbors and uses this number to compute its forwarding probability.

As in ACPF, the parameter k represents the passivity of a node in forwarding the query message. Figure 20 shows the variation of p' with respect to k. Notice that nodes in

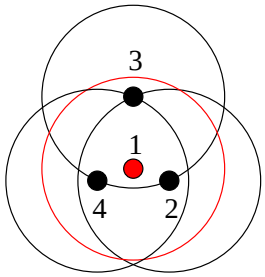

Figure 19: Redundancy case 2

Algorithm 6 CCPF

1: **if** q is a new query **then**
2: create a new counter $c(q) = 0$ for q;
3: set a timer $\tau = rand(0, t)$; buffer query q
4: **else**
5: $c(q) = c(q) + 1$; $exit$
6: **end if**
7: Listen for the same query q during the interval τ
8: After τ expires, compute $p' = p \times e^{-k \cdot c(q)}$
9: Rebroadcast q with probability p' and remove q from its buffer

CCPF randomly generate a listening interval, τ, within $[0, t]$. Consequently, the number of rebroadcasts overheard in a neighborhood is likely to differ from one node to another, leading to a different value of p' for each neighbor. Care must be taken in the choice of t to avoid excessively long listening time periods, while at the same time ensuring that the periods of nodes within the same neighborhood are different.

5.4 AREA AND COPIES COVERAGE-BASED PROBABILISTIC FORWARDING

ACCPF combines the main features of ACPF and CCPF. It uses both the extra coverage area and the number of copies heard over a neighborhood to compute the forwarding probability. In ACCPF, each node maintains a counter, $c(q)$, and the smallest additional coverage area, $E_{min}(q)$, for each query q. When a node $N2$ receives a query q from sensor $N1$, it computes its forwarding probability, p', as follows:

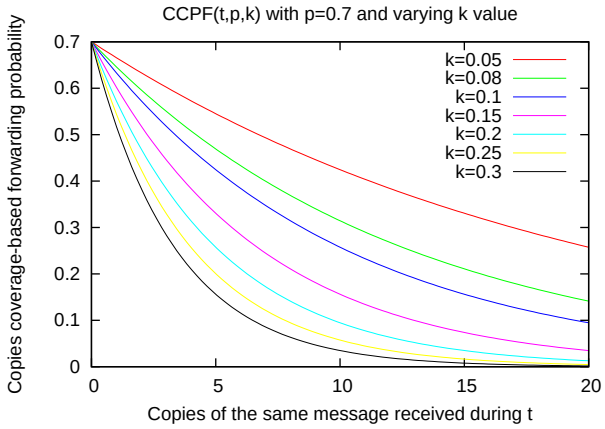

Figure 20: Effect of node passivity, k, on copies coverage-based forwarding probability

Algorithm 7 ACCPF

1: **if** q is a new query **then**
2: $c(q) = 0$; $E_{min}(q) = E(N2, N1)$;
3: set a timer $\tau = rand(0, t)$; buffer query q
4: **else**
5: $c(q) = c(q) + 1$
6: $E_{min}(q) = min(E(N2, N1), E_{min}(q))$ and $exit$
7: **end if**
8: Listen for the same query q during the interval τ
9: After τ expires, compute the new probability $p' = p \times e^{k_1 \times \frac{E_{min}(q) - E_{max}}{\pi r^2}} \times e^{-k_2 \cdot c(q)}$
10: Rebroadcast q with probability p' and remove q from its buffer

5.5 NEIGHBOR COVERAGE-BASED PROBABILISTIC FORWARDING

In ACPF, CCPF and ACCPF, a node's rebroadcast is deemed redundant if its entire transmission area is covered by other nodes. It is possible, however, that some parts of a node's coverage may not be populated and need not be covered. The goal is to cover all nodes, rather than areas in the network. Neighboring information can, therefore, be used to further reduce unnecessary rebroadcasts within a neighborhood. More specifically, if all neighbors of node N are already covered by other nodes' rebroadcasts, node N does not need to forward the query message any further. This observation is then extended to the Neighbor Coverage-based Probabilistic Forwarding (NCPF) algorithm.

In NCPF, prior to forwarding a query message, a node must first collect neighboring information. This information is then included in the query message. Upon receiving a query message q from $N1$, which contains a neighboring list, node $N2$, does not forward the message if its neighbors are already covered. Otherwise, it computes its forwarding probability based on how many copies of q it has received. Algorithm 8 presents the details of this algorithm.

Algorithm 8 NCPF

1: **if** q is a new query **then**
2: $c(q) = 0$; $S = neighbors(N1)$; set a timer $\tau = rand(0, t)$; buffer query q
3: **else**
4: $c(q) = c(q) + 1$; $S = S \cup neighbors(N1)$ and $exit$
5: **end if**
6: Listen for the same query q during the interval τ
7: After τ expires, compute the new probability $p' = p \times e^{-k \cdot c(q)}$
8: **if** $neighbors(N2) \subseteq S$ **then**
9: $p' = 0$
10: **end if**
11: Replace neighbor information in q with the neighbors of $N2$, rebroadcast q with probability p' and remove q from its buffer

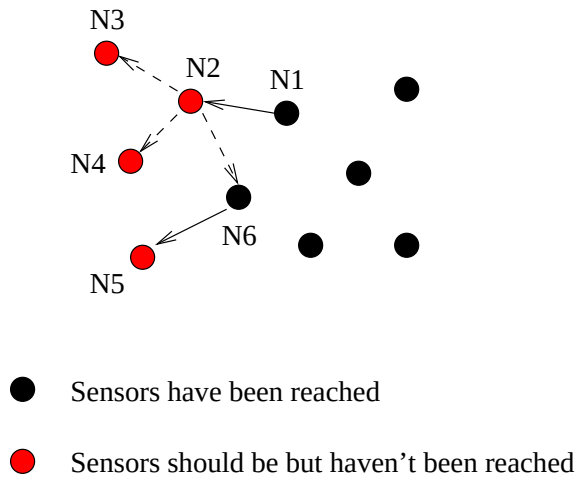

Figure 21: Example of path directed forwarding

5.6 UNCOVERED SENSORS AFTER PROBABILISTIC FORWARDING

The probabilistic forwarding schemes discussed above have the potential to reduce the number of query messages being forwarded in the network. However, they cannot ensure that all sensors are covered during the probabilistic forwarding. Some sensors which should receive a copy of the query message may not have been reached at the end of the probabilistic forwarding. To address this problem, a remedy process is proposed.

Given a query, q, let $R(q)$ be the set of sensors to be reached by q. $R(q)$ can be a subset of sensors if constraints in q such as a geophysical requirement are used to filter out certain sensors. Otherwise, $R(q)$ simply includes all sensors. After the probabilistic forwarding, let $Reached(q)$ be the set of sensors who receive a copy of q. The problem occurs when $R(q) - Reached(q)! = \emptyset$.

A simple method to find out if $R(q) - Reached(q)$ equals $\emptyset$ is to ask every sensor who receives q to return its identity to the base station. This can be achieved by sending a message to parent of each sensor. These messages can be aggregated during transmission. If the base station discovers that some sensors are not reached during the probabilistic forwarding, it identifies paths from sensors in $Reached(q)$ to the sensors in $R(q) - Reached(q)$. q is then

forwarded to these uncovered sensors through these paths. Due to the broadcast nature of wireless signals, all neighbors of the intermediate sensors in these paths receive a copy of q during the forwarding. This process is refereed to as "Path directed forwarding".

To further illustrate "path directed forwarding", an example scenario is presented in Figure 21. In this scenario, two paths, $N1 \rightarrow N2$ and $N6 \rightarrow N5$ are used to reach the uncovered sensors after probabilistic forwarding. The other two sensors, $N3$ and $N4$ can be reached after $N2$ broadcasts q.

The path directed forwarding requires the base station to maintain the topology of sensor networks in order to find a path between the reached sensors of q and the sensors to be reached. This requirement may incur a large amount of overhead if sensor links keep changing. An alternative approach is to utilize the locations of sensors to identify the closest reached sensor for each uncovered sensor and then use location aided limited flooding at these reached sensors to forward q to uncovered sensors.

The remedy process described above can be used to send query q to these uncovered sensors after probabilistic forwarding. The process, however, should be used with caution. If a major number of sensors are left uncovered, the overhead of using the remedy process may overcome the savings in communication gained by probabilistic forwarding and the scheme may end up transmitting more messages than the basic flooding scheme.

5.7 SIMULATION RESULTS

5.7.1 Methodology

The number of reachable sensors after the forwarding process completes and the number of messages forwarded and received are used to measure the performance of the proposed schemes. The reachable nodes information reveals how many sensors receive a copy of the query message after the probabilistic forwarding and shows the coverage of the probabilistic forwarding scheme. The number of messages forwarded and received determines the communication cost of the scheme. In the simulation, message retransmissions due to collision

are not considered since they are related to the medium access control protocol in the sensor network and are not the focus of probabilistic forwarding schemes. In the simulation, it is assumed that no collision or packet errors occur.

During the simulations, three different density of networks are considered by deploying different number of sensors in a 100m× 100m field using uniform random model. The sensor transmission range is 20m. The number of sensors and the average node degree are presented in Table 2.

Table 2: Average node degree deg in a network of n nodes for probabilistic forwarding

$density$	low	medium	high
n	50	100	150
deg	5.28	10.04	14.80

5.7.2 Performance Comparison

In addition to the proposed schemes, the results for PKGOSSIP, P1P2KNGOSSIP and PKMGOSSIP [88], are collected and presented as well for comparison. In PKGOSSIP, the first k hops of sensors from the source always broadcast to prevent the early death of the query message and all other sensors forward queries with probability p. In P1P2KNGOSSIP, if the number of neighbors of a sensor is bigger than N, it uses a small probability, $p1$, to forward a query. Otherwise, it uses $p2$. N is set to be 4 during the simulation. In PKMGOSSIP, a sensor rebroadcasts a query if it does not broadcast the query the first time it receives the message and overhears fewer than m copies of the same query during a small period of time.

During the simulation, each sensor is selected once as the source of a broadcast to minimize the effect of topology on the simulation results.

5.7.2.1 ACPF, CCPF, ACCPF and NCPF

Figure 22,23 and 24 present the result in the low density network. For ease of comparison among different schemes, we list the smallest number of messages forwarded and received to

Table 3: Parameters for schemes simulated

Scheme	p	p1	p2	k	m	t
PKGOSSIP	0.1-0.7			2		
PKMGOSSIP	0.1-0.7			2	2	
P1P2KNGOSSIP		0.1-0.7	0.8	2		
ACPF	0.1-0.7			1		
CCPF	0.1-0.7			0.25		$8 \times MLD$
ACCPF	0.1-0.7			k_1=1		$8 \times MLD$
				k_2=0.25		
NCPF	0.1-0.7			0.25		$8 \times MLD$

reach a comparable number of sensors among these schemes in Table 4. In the low density network, although ACPF, CCPF, ACCPF, and NCPF incurs less communication overhead, they reach much fewer sensors in the network.

The number of reachable sensors increases in the medium density network, as shown in Figure 25 and Table 5. The number, nonetheless, is still smaller than some gossip schemes even though fewer number of query messages are forwarded and received in the network. In the high density network, on the other hand, ACPF, CCPF, ACCPF and NCPF can reach a comparable number of sensors to the gossip schemes, yet still using fewer number of messages for query dissemination. The results are presented in Figure 28 29 and 30, and summarized at Table **??** for comparison.

In summary, ACPF, CCPF, ACCPF and NCPF can reduce the number of messages needed in probabilistic forwarding for query dissemination in all networks, but can only reach a large number of sensors in a high density network. Therefore, they should not be used in low or medium density networks.

5.7.2.2 ACPFM, CCPFM, ACCPFM and NCPFM

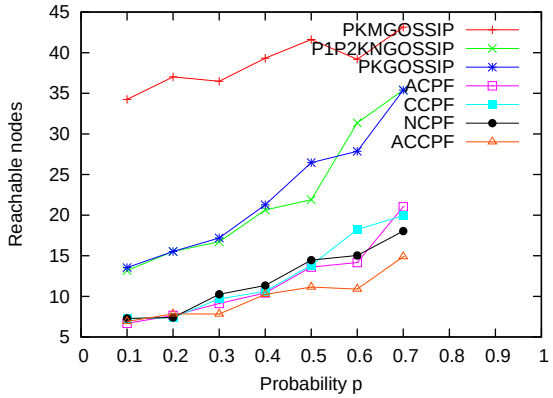

Figure 22: Number of reachable nodes in a low density network

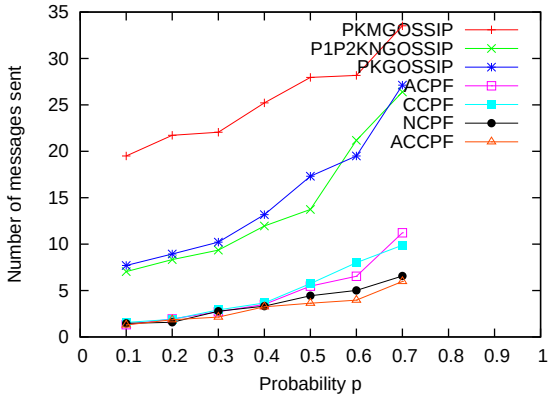

Figure 23: Number of messages forwarded in a low density network

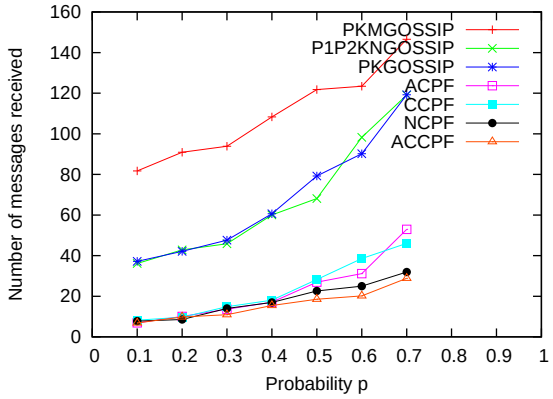

Figure 24: Number of messages received in a low density network

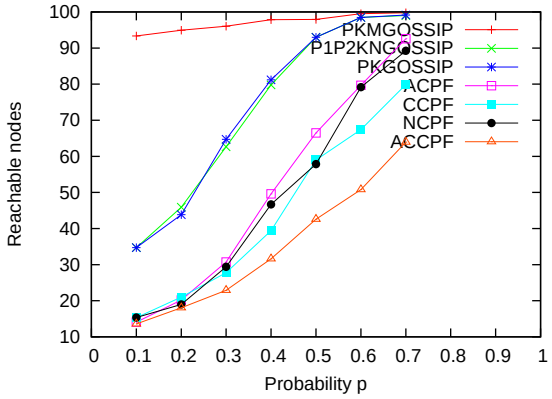

Figure 25: Number of reachable nodes in a medium density network

Table 4: Low density network

	PKGOSSIP	PKM GOSSIP	P1P2KN GOSSIP	ACPF	CCPF	ACCPF	NCPF
Reachable Nodes	35.4	34.24	35.4	21.02	19.94	14.88	18.04
Messages Forwarded	27.12	19.5	26.4	11.22	9.88	5.98	6.56
Messages Received	119.24	81.74	119.1	52.94	46.02	28.86	31.92
Probability P	0.7	0.1	0.7	0.7	0.7	0.7	0.7

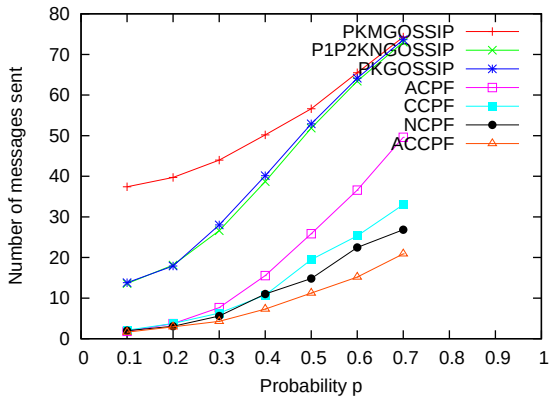

Figure 26: Number of messages forwarded in a medium density network

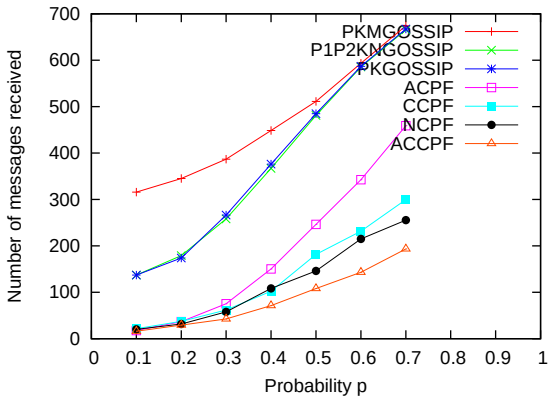

Figure 27: Number of messages received in a medium density network

Table 5: Medium density network

	PKGOSSIP	PKM GOSSIP	P1P2KN GOSSIP	ACPF	CCPF	ACCPF	NCPF
Reachable Nodes	99.14	99.52	99.01	92.59	79.89	63.95	89.25
Messages Forwarded	73.62	65.49	72.99	49.58	33.02	20.88	26.84
Messages Received	667.29	593.47	665.52	459	299.84	193.44	255.43
Probability P	0.7	0.6	0.7	0.7	0.7	0.7	0.7

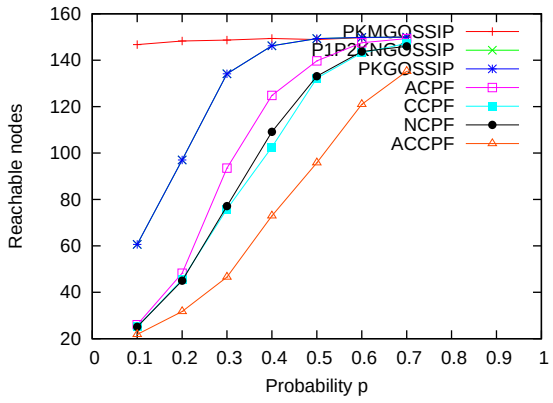

Figure 28: Number of reachable nodes in a high density network

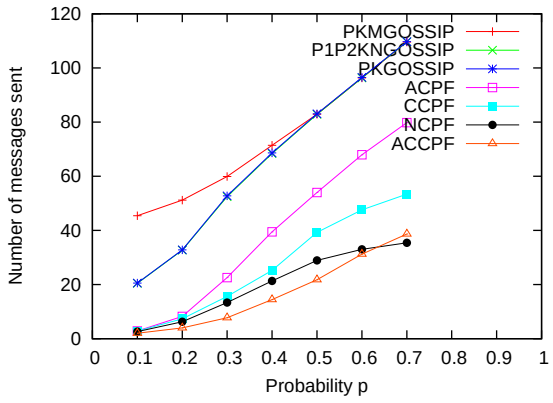

Figure 29: Number of messages forwarded in a high density network

Table 6: High density network

	PKGOSSIP	PKM GOSSIP	P1P2KN GOSSIP	ACPF	CCPF	ACCPF	NCPF
Reachable Nodes	149.287	148.26	149.287	149.333	147.26	135.227	146
Messages Forwarded	83.0733	51.18	68.35	79.78	53.4	38.67	35.4
Messages Received	1156.75	694.5	960.767	1107.73	717.44	528.37	509.4
Probability P	0.5	0.2	0.4	0.7	0.7	0.7	0.7

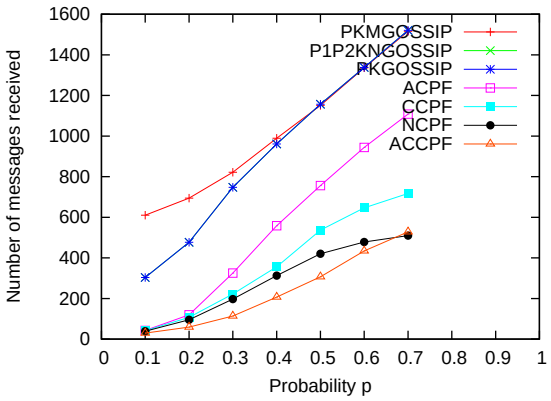

Figure 30: Number of messages received a high density network

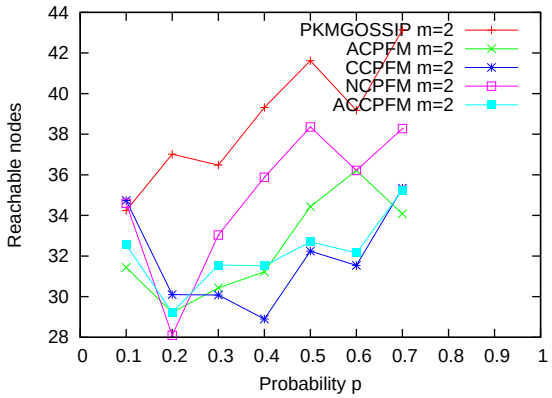

Figure 31: Number of reachable nodes with m-technique in a low density network

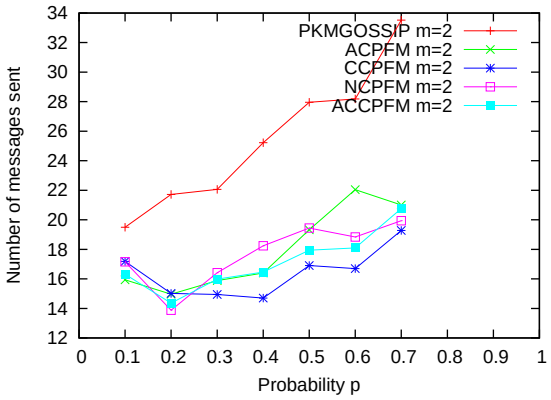

Figure 32: Number of messages forwarded with m-technique in a low density network

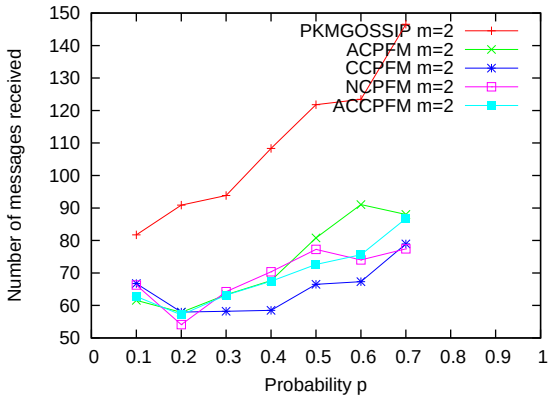

Figure 33: Number of messages received with m-technique in a low density network

The results for ACPFM, CCPFM, ACCPFM and NCPFM are presented at Figure 31-33, 34-36, and 37-39 for the low, medium and high density network, respectively. From the comparison shown in Table 7,8 and 9, it is easy to see that ACPFM, CCPFM, ACCPFM and NCPFM can reach approximately the same number of sensors as gossip schemes with fewer number of messages in all three networks with different levels of density. They can deliver the query to most sensors in the network in medium and high density networks, at the cost of longer transmission time due to the extra waiting time for sensors who don't forward the query message at first, but then forward it using the m-technique.

5.8 SUMMARY

In this chapter, probabilistic query dissemination is presented to reduce the communication overhead for delivering the queries from semantic views to all relevant sensors in the network. Several approaches which adapt the forwarding probability to different types of local topology

Table 7: Low density network with m-technique

	PKMGOSSIP	ACPFM	CCPFM	ACCPFM	NCPFM
Reachable Nodes	37.02	36.22	35.34	35.24	38.36
Messages Forwarded	21.72	22.04	19.28	20.76	19.44
Messages Received	90.9	91	78.96	86.74	77.26
Probability P	0.2	0.6	0.7	0.7	0.5

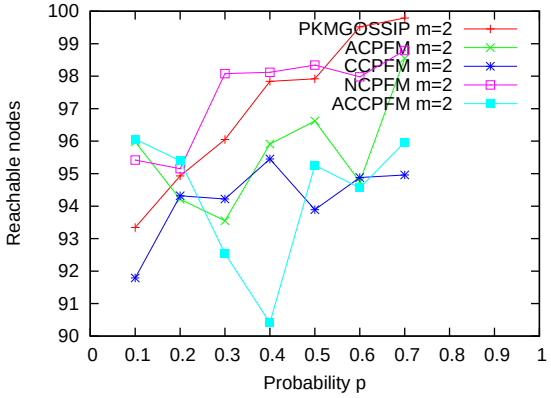

Figure 34: Number of reachable nodes with m-technique in a medium density network

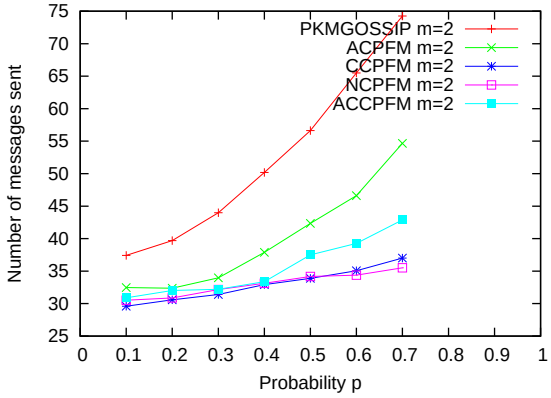

Figure 35: Number of messages forwarded with m-technique in a medium density network

Table 8: Medium density network with m-technique

	PKMGOSSIP	ACPFM	CCPFM	ACCPFM	NCPFM
Reachable Nodes	97.92	96.62	95.45	96.06	98.08
Messages Forwarded	56.63	42.34	32.93	30.89	32.16
Messages Received	511.25	371.1	270.95	250.42	264.64
Probability P	0.5	0.5	0.4	0.1	0.3

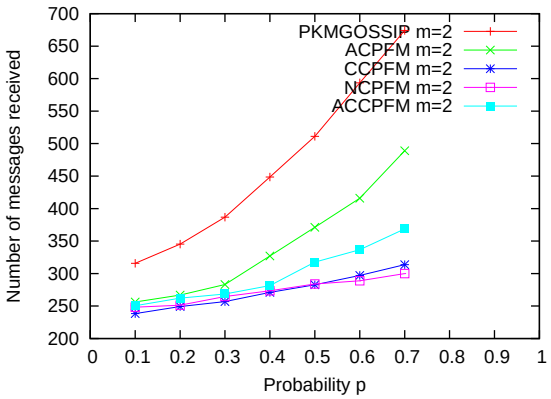

Figure 36: Number of messages received with m-technique in a medium density network

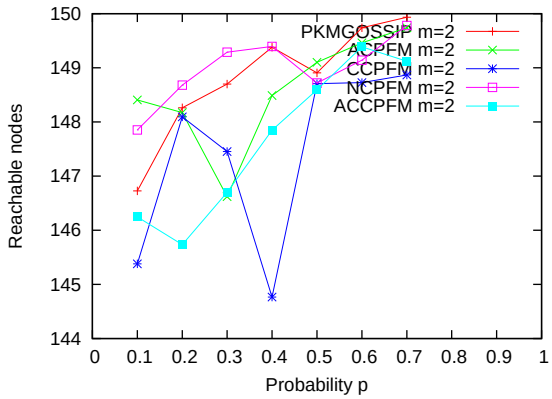

Figure 37: Number of reachable nodes with m-technique in a high density network

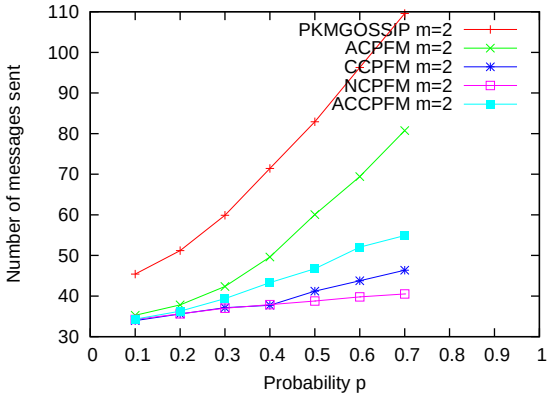

Figure 38: Number of messages forwarded with m-technique in a high density network

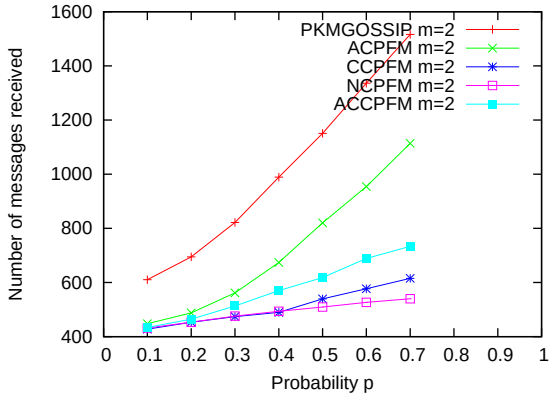

Figure 39: Number of messages received with m-technique in a high density network

Table 9: High density network with m-technique

	PKMGOSSIP	ACPFM	CCPFM	ACCPFM	NCPFM
Reachable Nodes	148.26	149.74	148.867	149.387	149.78
Messages Forwarded	51.18	69.4	46.34	52	40.5
Messages Received	694.5	954.3	615.06	688.4	539.9
Probability P	0.2	0.6	0.7	0.6	0.7

information are investigated. It is also shown how to enhance probabilistic dissemination with additional schemes to cover a very high percentage of sensors after the query dissemination.

6.0 CORRELATED MULTI-QUERY PROCESSING

In SQL, a correlated sub-query is a sub-query (a query nested inside another query) that uses values from the outer query. In current databases, e.g. Oracle, the same correlated sub-query of several outer queries is executed only once and the results are reused for all the evaluations in the outer queries. Similar ideas are applied in correlated multi-query processing for sensor networks.

Two queries are correlated if they need common sensor data for processing. These shared common data are then identified, collected and processed only once and the results are reused to answer these queries. The key question is how to identify and reuse the correlation among queries. It would be trivial if the set of sensor data for a query is already known before a query is processed. However, in sensor networks, upon receiving a query, the base station needs to either acquire all data from sensors or disseminate the query to all sensor nodes in order to find out what sensors provide data for the query. This process involves a lot of data transmissions among sensor nodes. The challenge is to identify the correlation among queries without knowing exactly the set of sensors having data for the queries.

It is observed that the number of sensors having data for a query is determined by the constraints that the query specifies over the data attributes at each sensor node. Given a data attribute, a query having a larger range constraint is mapped to more sensors in the network than another query with a smaller range constraint during processing. Based on this observation, an estimation model is developed to approximate the number of shared sensor data among queries. From the estimation model, the base station can further construct a set of shared intermediate views (SIV) to capture the shared common data among queries. However, without knowing exactly what set of sensors have data for what queries at the base station, some common sensor data shared among queries is left unidentified after the

shared intermediate views are constructed. Sensors, in contrast, know exactly if their data are needed by a query for processing. Hence, it is also investigated how the data collection at sensor nodes can be enhanced to reduce the communications of data collection for multiple correlated queries. Details of the estimation model and other algorithms of correlated multi-query processing at base station and sensor nodes are discussed in the following sections.

In this chapter, shared intermediate view is firstly defined to capture the correlation among queries and it is shown that how the results of these shared intermediate views can be reused to save message transmissions in data collection for multi-query processing. A numerical model is developed to estimate the level of correlation among queries, based on which a set of shared intermediate views with maximum level of correlation is constructed to minimize the data collection cost for multi-query processing. Furthermore, correlated data collection is presented to reduce data transmissions from sensors' perspective.

6.1 PROBLEM STATEMENT

Upon completion of sensing, each sensor node N has relevant data for a subset of queries of Q. The data collection for query q can be modeled as a routing tree, $RT(q)$, rooted at the base station node. Figure 40 shows example routing trees for two queries.

Each edge in $RT(q)$ represents one communication message from the child node to the parent node in the tree. Each node aggregates its own data with the data collected from its children nodes during data collection. As a result, the communication cost of each link is the same in the routing tree. The overall energy cost of data collection for a query q is as follows:

$$e_r + e_s + \sum_{N \in RT(q)} (e_r + e_s) \tag{6.1}$$

The cost of collecting data for a set of queries Q is the summation of 6.1 if each query is

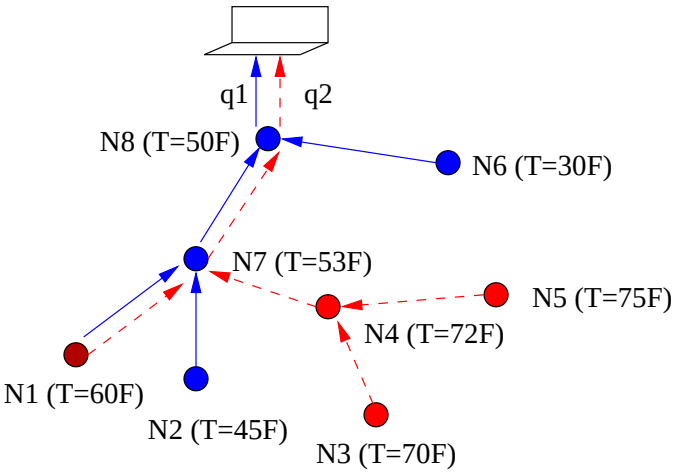

Figure 40: Query routing trees for data collection

processed separately:

$$C(Q) = \sum_{q \in Q} (e_r + e_s + \sum_{N \in RT(q)} (e_r + e_s)) \tag{6.2}$$

However, it is easy to find scenarios where the data set of different queries overlaps. In Figure 40, q_1 collects data from nodes $N1, N2, N6, N7$ and $N8$. q_2 collects data from $N1, N3, N4$ and $N5$. Assume that the routing trees for q_1 and q_2, as shown in Figure 40, are already optimized in terms of communication cost. If q_1 and q_2 are processed separately, then $N1$ has to transmit its data two times to $N7$. However, $N1$ should only need to send its data to $N7$ once during data collection, since the same data can be reused for both q_1 and q_2. A redundant message transmission can be avoided then if the overlapping data of q_1 and q_2 at $N1$ is reused, thereby reducing energy expenditure.

The main challenges of reusing shared data among queries to reduce multiple query processing costs are:

- How to find the set of shared sensor data among a set of queries and if there are multiple possible sets of shared data, which set should be selected
- How to preserve the semantic correctness of query results when the shared data is reused for multiple query processing

6.2 OVERVIEW

The correlated multi-query processing consists of two tiers: one tier at the base station and the other tier at sensor nodes. At the base station, the scheme tries to identify the sharing of sensor data among queries from the query constraints. At sensor nodes, the data collection/aggregation is modified to further reduce the redundant data transmissions of correlated queries.

The query model is described in section **??**. Essentially, queries may arrive at any time within an epoch. At the beginning of an epoch, the base station is given a set of queries, Q. From Q, the base station constructs a set of shared intermediate views. Each intermediate view identifies a set of shared data among queries in Q. These intermediate views are then processed by sensors in the network before any query in Q is processed. Each query, $q \in Q$, is mapped to several intermediate views and an additional set of sensors in the network. The results from these intermediate views, aggregated with data collected from an additional set of sensors, provide the necessary data to answer query q. Figure 41 illustrates the overall process of correlated multi-query processing at the base station.

Later in the same epoch, when new query/queries arrive at the base station, the base station checks the already available results from the existing shared intermediate views as well as existing queries. To reuse any of the existing results, the base station must make sure that the existing SIV/query is contained in the new query. The query containment problem in general is a very difficult problem and NP-hard in relational databases. A simple algorithm which checks the constraints in queries is used to determine if a query is contained in another query. The simplified algorithm finds a subset of contained queries but only

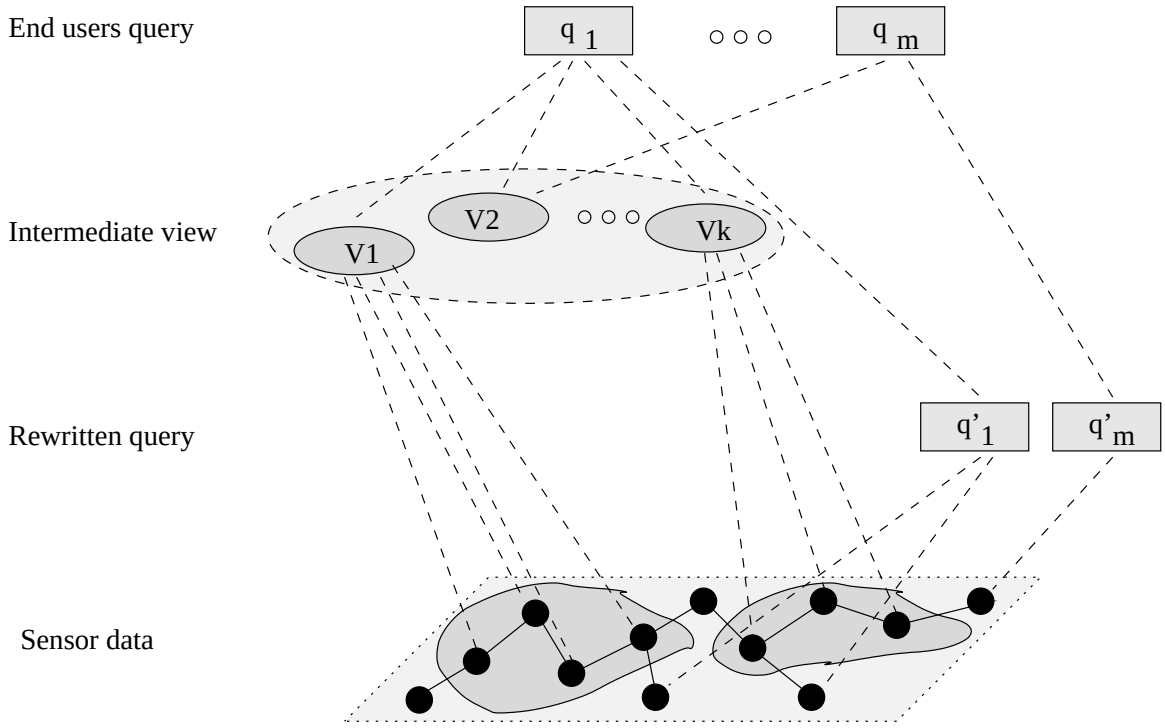

Figure 41: Overview of correlated multi-query processing

requires a polynomial time of computation. A new set of shared intermediate views is then derived from the new query/queries and the existing contained SIV/queries. The new set of SIVs is also added into the existing SIVs for processing future queries. The procedure is repeated for all new queries until the end of an epoch. Because sensor data may change from epoch to epoch, the results of the SIVs in the current epoch cannot be reused by other queries in the next epoch. Therefore, at the end of one epoch, the SIVs are removed and at the beginning of the next epoch a new set of SIVs is constructed by the base station.

The goal of correlated multi-query processing at the base station is to eliminate the overlapping of sensor data among queries so that each sensor only needs to transmit/aggregate its data once. However, without knowing the actual correlation, i.e. the shared set of sensor data among queries, there might still be common data among SIVs and the rewritten queries. In other words, a sensor node may still need to send its data more than once for the SIVs and rewritten queries sent from the base station.

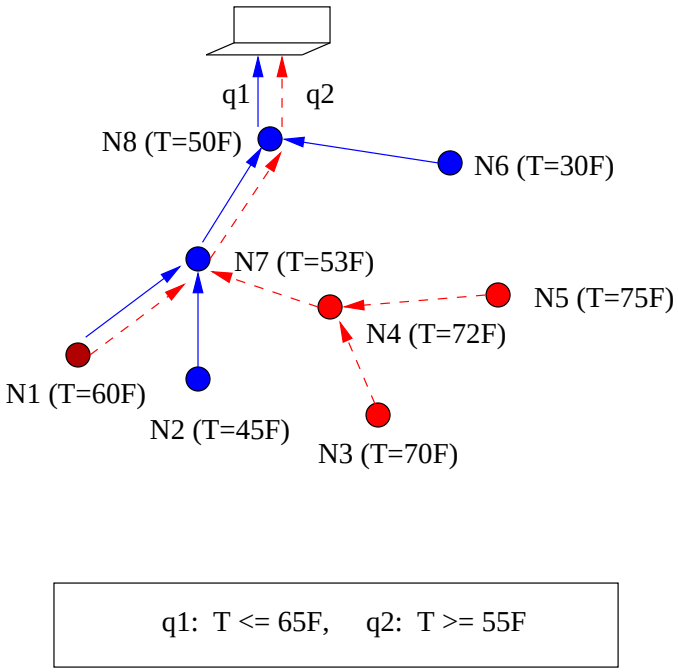

Figure 42: Example of data aggregation for correlated queries

Figure 42 presents an example of data aggregation for correlated queries in sensor nodes.

In this example, query q1 is processed first and then query q2 is disseminated for processing. For q1, the data at node $N1$ is transmitted to $N7$ and $N7$ aggregates its own data with $N1$ and $N2$ during processing. Later when q2 arrives at $N1$, $N1$ needs to transmit its data again to $N7$ for q2. Because the data of $N1$ has been sent to $N7$ during q1's processing, q2 can simply acquire $N1$'s data from $N7$ if $N7$ keeps a copy of $N1$'s data after aggregation. Therefore, the redundant transmission of $N1$'s data from $N1$ to $N7$ can be saved for processing q2. In general, a leaf node in an aggregation tree uses its aggregator as a proxy node for its data. Any later queries requesting data from the leaf node can retrieve the data from its proxy node. Since in an aggregation tree, the aggregator, i.e. proxy node, should be closer to the base station, the communications of the intermediate sensors between the leaf node and its proxy node can also be saved.

6.3 CORRELATED MULTI-QUERY PROCESSING AT THE BASE STATION

6.3.1 Shared Intermediate Views

The key idea of correlated multi-query processing at the base station is to reuse the correlations among queries. Assuming that the correlation among queries, i.e. the shared common set of sensor data, can be identified to the base station, a unified method is needed to reuse these common data.

To this end, "Shared Intermediate Views (SIV)" is defined to capture the correlation among queries. In the query definition language described in section **??**, a rule, R, is a conjunction of predicates. A query, q, uses a disjunction of rules to specify the conditions of data to be collected. Let $Rules(q)$ be the disjunction set of rules q specifies. A shared intermediate view is defined as follows:

Definition 7. *Given two queries, q_1 and q_2, a shared intermediate view, SIV, of (q_1, q_2) is a query AF(V)?R, where AF(V) is the same as q_1 and q_2, and $R = Rules(q_1) \wedge Rules(q_2)$.*

Based on Definition 7, a shared intermediate view set (SIVS) is defined as follows for a

set of queries, Q:

Definition 8. *Given a set of queries, Q, a shared intermediate view set, SIVS, is a set of SIVs of queries in Q. Furthermore, $\forall\ SIV_i, SIV_j \in SIVS$, $queries(SIV_i) \wedge queries(SIV_j) = \emptyset$.*

It is easy to see that an SIV of q_i and q_j is mapped to the common set of sensor data that q_i and q_j share, because each sensor data satisfying SIV must meet all the constraints of both q_i and q_j. Therefore, the concept of SIV provides a method to specify the common set of sensor data between two queries, q_i and q_j, using only the constraints of the queries. In this way, the SIV is independent from the sensor network. No matter which sensors in the network q_i and q_j collect data from, SIV(q_i, q_j) is always mapped to the set of shared sensor data between q_i and q_j. As shown in Figure 43, a SIV of q_i and q_j is always equivalent to the shared set of sensor data between q_i and q_j.

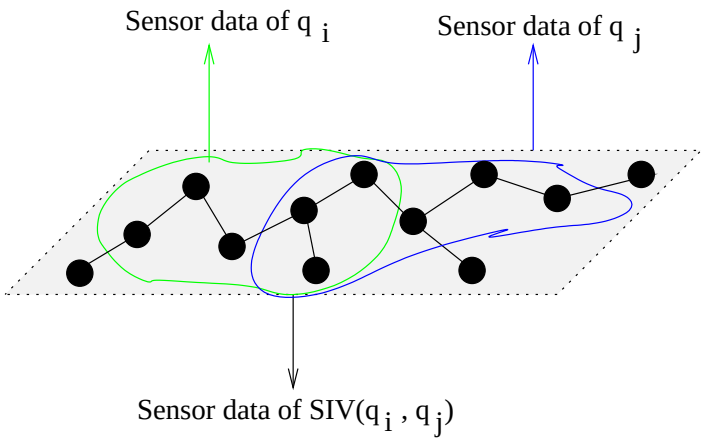

Figure 43: SIV of two queries

Using Definition 8, an SIV can be defined for any pair of queries. However, different SIVs collect data from different number of sensors. Some SIVs may not even acquire any data from the network simply because no sensed data satisfies the constraints in the SIV. It makes no sense to use such SIVs since no common sensor data can be reused for processing.

The correlated multi-query processing relies on the reusing of the processing results of SIVs to reduce the overall processing cost. The construction of SIVs, therefore, is critical to

the performance of correlated multi-query processing. The goal is to derive an SIVS which enables the maximum reusing of shared data among queries. To derive such an SIVS, the size of the shared data between queries must be known. This knowledge, however, cannot be obtained before the relevant sensor nodes are identified for each query. Nonetheless, heuristics can be explored for the maximization of reusing shared data. A simple range-based algorithm for optimum SIVS construction is discussed next.

6.3.2 Range-Based SIVS Construction

A query uses constraints to specify what sensor data to collect for processing. The constraints essentially form a range requirement over data attributes being sensed at sensor nodes. Intuitively, the larger the ranges are, the more data sensors in the network have for a query.

In more detail, if the probability distribution function of a data attribute A, $Pr[A]$, is known, then given a predicate of attribute A, $P(A) : a_l \leq A \leq a_u$, the probability of the sensed value of A at a sensor node is $Pr[a_l \leq A \leq a_u]$. Furthermore, assuming that the sensors sense independently, the expected number of sensors in the network whose data is in the range (a_l, a_u) is $Pr[a_l \leq A \leq a_u] \times n$, where n is the number of sensors in the network. Since n is a fixed value for a sensor network, the amount of sensor data for a predicate $P(A) : a_l \leq A \leq a_u$, can be estimated as $Pr[a_l \leq A \leq a_u]$.

Therefore, for a single predicate, $P(A)$, of attribute A, $P(A) : a_l \leq A \leq a_u$, the estimated number of sensors having data for $P(A)$, denoted as $RR(P(A))$, is defined as follows:

$$RR(P(A)) = Pr[a_l \leq A \leq a_u] \tag{6.3}$$

The probability distribution function can be determined from the historical sensing data. If the sensing attribute value is uniformly distributed, the probability is $\frac{a_u - a_l}{max(A) - min(A)}$, and it is proportional to the range of the predicate.

A rule, R, may specify more than one predicate on a data attribute, A. These predicates, $P^1(A), P^2(A), ..., P^m(A)$ form a composite predicate $Pred(R, A) = P^1(A) \wedge P^2(A) \wedge \cdots \wedge P^m(A)$, which in turn defines a range requirement over A. The estimated relevant data size

of $Pred(R, A)$ is defined as follows:

$$RR(Pred(R, A)) = Pr[\max_{1 \le i \le m} a_l(P^i(A)) \le A \le \min_{1 \le i \le m} a_u(P^i(A))] \tag{6.4}$$

A rule R may also impose range constraints over more than one attribute of T, $ATTR = \{A_1, A_2, \cdots, A_t\}$. In this case, the sensor data must satisfy all the constraints of all the attributes. While the number of sensors can be estimated to satisfy any single range constraint over a single data attribute, it is very difficult to estimate the number of sensors in the range requirements of several data attributes unless the relations among these data attributes are known. As a result, the estimated number of sensors whose data satisfy the constraints in R, $RR(R)$, is roughly approximated as the smallest range requirements over all data attributes:

$$RR(R) = \min_{\forall A \in ATTR(R)} RR(Pred(R, A)) \tag{6.5}$$

With a simple derivation, the conjunction of rules can be transformed into the conjunction of predicates. Therefore, the estimated number of sensors whose data satisfy the conjunction of rules can be defined in a way similar to that for a single rule. The definition is given as follows:

$$RR(R_i \wedge R_j) = \min_{\forall A \in ATTR(R_i) \cup ATTR(R_j)} RR(Pred(R_i, A) \wedge Pred(R_j, A)) \tag{6.6}$$

Determining the correlation of a disjunction of two rules, however, is a little more complex. The sensors having data for the disjunction of two rules, R_i and R_j, is the union of the set of sensors whose data satisfy R_i and R_j. Similar to when computing the union of two sets, the number of sensors whose data satisfy the disjunction of two rules is approximated as follows:

$$RR(R_i \vee R_j) = RR(R_i) + RR(R_j) - RR(R_i \wedge R_j) \tag{6.7}$$

Using the definitions above, the correlation between two queries can be quantified. Given two queries, q_1 and q_2, where $Rules(q_1) = R_1^1 \vee R_1^2 \vee \cdots \vee R_1^{l_1}$, and $Rules(q_2) = R_2^1 \vee R_2^2 \vee$

$\cdots \vee R_2^{l_2}$, the correlation between q_1 and q_2, $COR(q_1, q_2)$, is defined as:

$$\begin{aligned} COR(q_1, q_2) &= RR(Rules(q_1) \wedge Rules(q_2)) \\ &= RR(\bigvee_{1 \le i \le l_1, 1 \le j \le l_2} (R_1^i \wedge R_2^j)) \end{aligned} \tag{6.8}$$

Equation 6.8 provides us with a model to estimate the size of shared data among two queries. The model, however, requires an exponential number of computations to estimate the correlation between two queries. That is, following equation 6.7, the disjunction of $l1 \times l2$ rules needs to compute $2^{l1 \times l2} - 1$ intermediate values before the final value can be obtained. Therefore, to reduce the number of computations required, the correlation definition is approximated to be a lower and upper bound of equation 6.8.

From equation 6.6 and 6.7, it can be derived that:

$$\begin{aligned} RR(R_i \vee R_j) &= RR(R_i) + RR(R_j) - RR(R_i \wedge R_j) \\ &\le RR(R_i) + RR(R_j) \end{aligned}$$

and

$$\begin{aligned} RR(R_i \vee R_j) &= RR(R_i) + RR(R_j) - RR(R_i \wedge R_j) \\ &\ge \max(RR(R_i), RR(R_j)) \end{aligned}$$

Therefore, a lower and upper bound of the conjunction of two rules can be derived as follows:

$$\max(RR(R_i), RR(R_j)) \le RR(R_i \vee R_j) \le RR(R_i) + RR(R_j) \tag{6.9}$$

Inequality 6.9 can be extended to compute the conjunction of m rules:

$$\max_{1 \le i \le m} RR(R_i) \le RR(\bigvee_{i=1}^{m} R_i) \le \sum_{i=1}^{m} RR(R_i) \tag{6.10}$$

Given 6.10, a lower and upper bound for correlation among two queries, q_1 and q_2, can now be derived.

$$\max_{1 \le i \le l_1, 1 \le j \le l_2} RR(R_1^i \wedge R_2^j) \le COR(q_1, q_2) \le \sum_{1 \le i \le l_1, 1 \le j \le l_2} RR(R_1^i \wedge R_2^j) \tag{6.11}$$

Similarly, another upper bound of the $RR(R_i \vee R_j)$ can be defined as follows:

$$\begin{aligned} RR(R_i \vee R_j) &= RR(\bigwedge_{A \in ATTR(R_i)} Pred(R_i, A) \vee \bigwedge_{A \in ATTR(R_j)} Pred(R_j, A)) \\ &= RR(\bigwedge_{A_1, A_2 \in ATTR(R_i) \cup ATTR(R_j)} (Pred(R_i, A_1) \vee Pred(R_j, A_2))) \\ &\leq \min_{A \in ATTR(R_i) \cup ATTR(R_j)} RR(Pred(R_i, A) \vee Pred(R_j, A)) \end{aligned} \tag{6.12}$$

This upper bound of $RR(R_i \vee R_j)$ leads to another upper bound, that of the correlation among two queries, q_1 and q_2.

$$\begin{aligned} COR(q_1, q_2) &= RR(\bigvee_{1 \leq i \leq l_1, 1 \leq j \leq l_2} (R_1^i \wedge R_2^j)) \\ &\leq \min_{A \in ATTR} RR(\bigvee_{1 \leq i \leq l_1, 1 \leq j \leq l_2} Pred(R_1^i \wedge R_2^j, A)\ ) \end{aligned} \tag{6.13}$$

The lower and upper bound in inequality 6.11 and 6.13 give three approximations of $COR(q_1, q_2)$. The upper bounds approximation represents an aggressive approach in estimating the correlations between queries while the lower bound represents a conservative method in estimation. All three approximations give estimated numbers of shared common data between queries. The effectiveness of these approximations depends on how close the estimations are to the actual amount of common sensor data between queries.

Given the model of correlation estimation between queries, it is far from trivial to construct the SIVS with the maximum correlation. The greedy algorithm, which always choose queries with highest value of correlation during each iteration, does not always yield an optimal SIVS. In a simple example of four queries, $Q = \{q_1, q_2, q_3, q_4\}$, assume $COR(q_1, q_2) = 0.4, COR(q_1, q_3) = 0.3, COR(q_2, q_4) = 0.3$, $COR(q_1, q_4) = 0.1$ and $COR(q_3, q_4) = 0.1$, the greedy algorithm constructs the SIVS as $\{SIV(q_1, q_2), SIV(q_3, q_4)\}$. However, $\{SIV(q_1, q_3), SIV(q_2, q_4)\}$ is the SIVS with the maximal correlation.

Given a set of queries, Q and the correlation values between any two queries in Q, a correlation graph, $G = (V, E)$, is constructed to construct the SIVS with the maximum correlation. For each query q in Q, add a node N_q to E. Two nodes, N_{q_i} and N_{q_j}, are connected in G if and only if $COR(q_i, q_j) > 0$. The weight of an edge is the correlation value of the two queries of the corresponding end nodes of the edge. The problem of SIVS

construction with maximal correlation is then equivalent to the maximal weighted match problem in the correlation graph G. For the example queries given above, the correlation graph is shown in Figure 44.

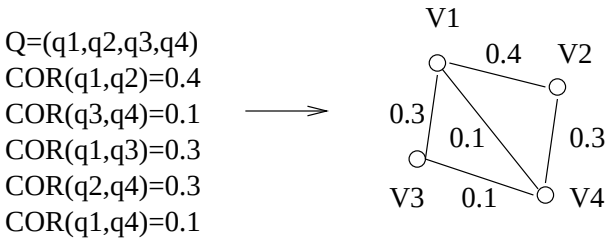

Figure 44: An example of a correlation graph

The maximal weighted match problem is well known in graph theory and has been extensively studied in [129] [130] [131]. Several polynomial algorithms have been proposed to find the optimal solution in polynomial time [132] [133]. The first polynomial algorithm is due to Edmonds [129], based on which improved algorithms were designed later. These algorithms are based on the idea of augmenting paths, which can be used to find new matching with a larger weight from an existing match. In this thesis, Gabow's algorithm is used to find the maximum weighted matching in the correlation graph. The details of Gabow's algorithm are presented in Appendix A.2. For each edge $e =< V_i, V_j >$ in the maximum weighted matching, a SIV is constructed for queries q_i and q_j. The SIVS with maximum correlation is the set of SIVs constructed from all edges in the maximum weighted matching in the correlation graph. The main steps are described in Algorithm 9.

6.3.3 Query Processing using SIVS

After a $SIVS$ is derived for a set of queries, Q, the shared intermediate views in $SIVS$ are firstly sent into the sensor network. Relevant data are collected and processed using an existing communication scheme such as [23]. In order to reuse the results from these shared intermediate views, the queries in Q are modified before they are delivered into the sensor network for processing.

Algorithm 9 SIVS Construction

INPUT:

$Q = \{q_1, q_2, \cdots, q_m\}$

INITIALIZATION:

$SIVS = \emptyset$

for all $q_i \in Q$ **do**

add node v_i to V

end for

for all $q_i, q_j \in Q$ **do**

if $COR(q_i, q_j) > 0$ **then**

add $e = (v_i, v_j)$ to E, $w(e) = COR(q_i, q_j)$

end if

end for

find the maximum weighted matching, M, from $G = (V, E)$

for all $e \in M, e = (v_i, v_j)$ **do**

$SIVS = SIVS \cup SIV(q_i, q_j)$

end for

OUTPUT:

$SIVS$

In principle, for an $SIV \in SIVS$, if SIV is the shared intermediate view for queries in $S \subseteq Q$, then for each query, $q \in S$, q is replaced with $q^{'} = AF(V)?Rules(q) \wedge \neg Rules(SIV)$. The algorithm for query rewriting is presented in Algorithm 10.

Algorithm 10 Query Rewriting

INPUT:
  $Q = \{q_1, q_2, \cdots, q_m\}$, $SIVS$
INITIALIZATION:
  $Q^{'} = Q$
for all $(SIV \in SIVS)$ **do**
  $S = \{q | q \in Q$ *and* SIV *is a shared intermediate view of* $q\}$
  for all $q \in S$ **do**
    $Q^{'} = Q^{'} - q$; $Q^{'} = Q^{'} \cup \{q^{'} = AF(V)?Rules(q) \wedge \neg Rules(SIV)\}$
  end for
end for
OUTPUT:
  $Q^{'}$

These modified queries in $Q^{'}$ are then delivered to sensors in the network. The relevant data are collected and an aggregated result is returned to the base station. To answer an original query q in Q, the aggregated result from $q^{'}$ and the SIV of q are aggregated together, using the aggregation function specified in q.

In the proposed query processing scheme using SIVS, the final result for q is the aggregated result from $q^{'}$ and the SIV. To ensure that the correctness of such an aggregation for query q, the following two conditions, 6.14 and 6.15 must be true.

$$Data(q) == Data(q^{'}) \cup Data(SIV) \tag{6.14}$$

$$Data(q^{'}) \cap Data(SIV) == \emptyset \tag{6.15}$$

Condition 6.14 means that all sensor data collected by q is collected by either $q^{'}$ or SIV. Since $q^{'}$ is constructed as $AF(V)?Rules(q) \wedge \neg Rules(SIV)$, it is easy to see all

data in $Data(q^{'})$ meets $Rules(q) \wedge \neg Rules(SIV)$. Data in $Data(SIV)$, on the other hand, satisfy condition $Rules(SIV)$. Therefore, data in $Data(q^{'}) \cup Data(SIV)$ meets the constraint $(Rules(q) \wedge \neg Rules(SIV)) \vee Rules(SIV)$. The constraint is equivalent to $Rules(q) \vee Rules(SIV)$. It follows then that $Rules(q) \vee Rules(SIV) = Rules(q)$ since $Rules(SIV) = Rules(q) \wedge \cdots \wedge Rules(q_j)$.

Condition 6.15 requires that no duplicate data exist between $Data(q^{'})$ and $Data(SIV)$, because some aggregation functions such as "Average" are duplication sensitive. Collecting a piece of data more than one time results in an inaccurate result. This condition, therefore, is used to ensure that the aggregation of the aggregated results for $q^{'}$ and SIV is always the same as the aggregated result for q for any aggregation function. The condition is obviously true since $Rules(q^{'}) \wedge Rules(SIV) = Rules(q) \wedge \neg Rules(SIV) \wedge Rules(SIV) = \emptyset$.

6.3.4 SIVS Update and Query Rewriting for New Queries

Section 6.3.2 and 6.3.3 explain how to process a set of queries at the beginning of an epoch. Later in the same epoch, new queries arrive at the base station. To explore the correlations among new queries, the existing results are reused if possible. The processing results of existing SIVs and queries can be reused if they are contained in a new query.

The query containment problem, which determines if one query is contained in another query, is very challenging and has been shown to be NP-hard in relational databases [134][135] [136][137][138][139]. Instead of finding all contained queries using non-polynomial algorithms, a simple algorithm is used to find a subset of contained queries with polynomial time. In this algorithm, a query, q_1, is contained at q_2 if and only if the range constraint for each data attribute , A, specified at each rule, R in q_1, $[a_l1, a_u1]$, is within the range constraint of the same data attribute in any single rule in q_2.

Apparently, a query can reuse the results from more than one contained query if these contained queries do not have any sensor data in common. However, without knowing the set of actual sensor data for each query, it is almost impossible to tell if two queries do not share any data in common. Therefore, for simplicity, only one contained query is reused in the proposed scheme. The question is which contained query should be reused? A new

query may also share more sensor data with another new query than existing queries.

To find the optimal solution, the correlation graph of new queries is augmented with additional nodes and edges. For each existing query or SIV, if it is contained in any new query, then a new node is added into the correlation graph. Furthermore, the new node is connected to nodes of new queries which the existing query or SIV is contained in. The weight of such an edge is the correlation between the contained query or SIV and the new query. Gabow's algorithm is used again to find the maximum weighted matching M for the augmented correlation graph. For each edge in M, if it connects to two nodes of new queries, then a new SIV is constructed and added to the existing SIVS. Otherwise, it means a new query should reuse processing results from an existing query or SIV. Since these results are already available at the base station, there is no need to collect these data again since sensors only sense once and the data remains the same within an epoch.

If a new SIV is constructed between two new queries, q_1 and q_2, then both q_1 and q_2 are modified as shown in Section 6.3.3. The rewritten queries and SIV are processed and the results are aggregated to answer q_1 and q_2. If a new query q reuses results from existing query q_e, then q is rewritten as $q^{'} = AF(V)?Rules(q) \cap \neg(q_e)$ and $q^{'}$ is sent to the sensor network for processing. The results of $q^{'}$, aggregated with the existing results of q_e, are used to answer q. The main steps of the SIVS update and query rewriting for new queries are presented in Algorithm 11.

6.4 CORRELATED DATA COLLECTION AT SENSOR NODES

The correlated multi-query processing scheme at the base station aims to reduce the communication cost at sensor nodes for query processing by using shared intermediate views. As effective as the scheme might be, it doesn't completely eliminate the redundancy of data communications at sensor nodes. In other words, a sensor might still need to transmit its data multiple times to the rewritten queries and shared intermediate views. The difficulty of reusing sensor data is due to the fact that the sensor data cannot be recovered once it

Algorithm 11 SIVS Update and Query Rewriting for New Queries

INPUT:

New Queries $Q = \{q_1, q_2, \cdots, q_m\}$, existing queries Q_e and SIVS

construct correlation graph G for Q

augment G with additional nodes and edges for contained queries between Q_e and Q

find the maximum weighted matching, M, from G

for all $e \in M, e = (v_i, v_j)$ **do**

  if q_i, q_j are new queries **then**

    $SIVS = SIVS \cup SIV(q_i, q_j)$

    rewrite both q_i and q_j

  else

    only rewrite the new query

  end if

end for

OUTPUT:

  $SIVS$

is aggregated with other sensor data. For example, from the average value of two pieces of sensor data, it is impossible to compute the original value of each sensor data.

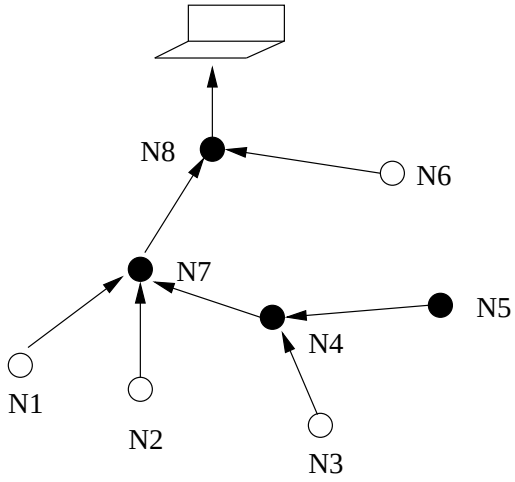

Figure 45: Example of data aggregation

Figure 45 gives an example of data aggregation in a sensor network. Data from nodes $N1, N2, N3$ and $N6$ are needed for a query. Data from $N1, N2$ and $N3$ are aggregated at $N7$. The aggregated value is aggregated again with $N6$'s data at $N8$ before it is returned to the base station. In this particular example, any ancestor node of $N7$ doesn't know what the data of $N1, N2$ and $N3$ are since they have been aggregated. In contrast, $N7$ knows the data of $N1, N2$ and $N3$ because it is the aggregation node for these data. As a result, if $N7$ holds the data of $N1, N2$ and $N3$ after the aggregation, a new query asking for data from $N1, N2$ or $N3$ can collect them from $N7$. In this way, data transmissions between $N1, N2$ or $N3$ can be saved for the new query.

This observation leads to the design of an enhanced data aggregation scheme in sensor networks. The basic idea is that the first aggregation node for a set of individual sensor data becomes a proxy node for these sensors as well. The proxy node stores a copy of the data it aggregates. Furthermore, any future query asking for data from a sensor retrieves it from its proxy node if there is one.

To ensure that the final aggregated result is correct for the new scheme, there are two

issues which must be resolved. First, the copy of data that a proxy holds for a sensor node may become invalid when the sensor node senses a different value in the next epoch. Collecting an expired data from an aggregation node would then lead to an incorrect processing result. A simple method is to attach a time stamp to each copy of data a proxy node stores. The data automatically expires and is deleted after the epoch during which it was sensed is over. In this way, a proxy node can reuse its storage space and a sensor node can also dynamically select different proxy nodes from epoch to epoch.

Second, the sensor data must be collected once and only once during aggregation. A simple solution is to maintain only one proxy for each piece of sensor data. The first time a sensor node receives a query asking for its data, it marks the first aggregation node in its aggregation tree as its proxy. Once a proxy node has been selected for a sensor node, any future queries will receive data from the proxy node. Even if the sensor node receives a query request for data, it simply discards these requests. If a proxy node fails for some reason, it is important that the sensor should select another proxy. Assuming that the node failure can be detected by its neighbors, one of these neighbors must send a message to the sensor node so that it can clear its mark and reselect another proxy node when a new query asks for its data.

Given the example of the aggregation tree in Figure 45, when another query, q_2, needing data from $N1, N2, N3$ and $N4$ is sent to the network, it only needs to retrieve data from $N4$ and the proxy node, $N7$, which has stored the data of $N1, N2$ and $N3$. The aggregation tree for q_2 using the proposed scheme is shown in Figure 46(b).

6.5 ANALYSIS

The range-based SIVS construction mainly consists of two steps. The first step is to build the correlation graph from queries. In this step, the correlations between each pair of queries are computed. Given a pair of queries, $q_1 = AF(V)?R_1 \vee R_2 \vee \cdots \vee R_{l_1}$, $q_2 = AF(V)?R_1 \vee R_2 \vee \cdots \vee R_{l_2}$, it takes $l_1 \times l_2$ computations to derive the correlations between them. Assuming

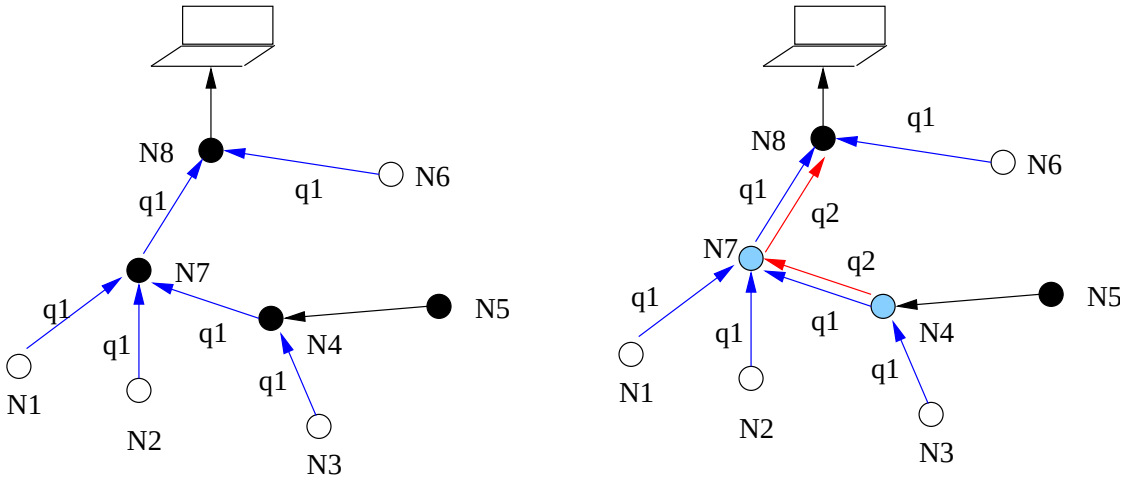

Figure 46: Example of correlated data aggregation

that the number of rules in a query is smaller than a certain threshold value l_{max}, the complexity of computing the correlation between two queries can be considered as a constant value. As a result, it takes $O(k^2)$ to construct a correlation graph from a set of k queries, $Q = \{q_1, q_2, \cdots, q_k\}$.

The complexity of maximum weighted matching is $O(mn \lg n)$, when Gabow's algorithm is used. n denotes the number of nodes in the graph and m is the number of edges in the graph. In our graph, the number of nodes is k and m is at most $k * (k-1)/2$. The SIVS rewrite only iterates at most k times, therefore its complexity is only $O(k)$. So the complexity of range-based SIVS construction is $O(k^3 \lg k)$.

Theorem 1. *Given a set of k queries, the time complexity of the SIVS construction algorithm is* $O(k^3 \lg k)$.

The correlated data collection at sensor nodes, in contrast, does not require complex computations, but need messages to be exchanged between sensors for proxy node selection and maintenance. Since a sensor node selects the first aggregation sensor along its path towards the sink as its proxy node, it takes at most h transmissions for N_1 to inform a

sensor node N_2 that N_1 has became a proxy node for N_2, where h is the length of the longest path in the routing tree towards sink for the proxy node. Similarly, it requires at most h transmissions for a sensor node to update its proxy node after the routing tree for data aggregation is established. In the worst case, h could be the number of sensors in the network, n, therefore, the message complexity of the correlated data collection at sensor nodes is $O(n)$.

Theorem 2. *The message complexity of correlated data collection at sensor nodes in a network of n sensors is* $O(n)$.

6.6 SIMULATION RESULTS

6.6.1 Methodology

To evaluate how effective correlated multi-query processing is in reducing the data communications in sensor networks, a set of simulations are conducted. In the simulations, the correlated multi-query processing scheme at the base station is compared with the basic scheme of processing queries separately. In correlated multi-query processing, every piece of sensor data in the shared intermediate view set is only transmitted/aggregated once. Therefore, the number of data transmissions saved is the number of messages needed to send these data to the base station. This number, however, depends on the structure of the data collection/aggregation tree used for the shared intermediate views. Nevertheless, these data must be transmitted at least once before the result is returned to the base station. To preserve the fairness of comparison, the amount of sensor data in the SIV is measured as the number of data transmissions saved by correlated multi-query processing at base station, so that the saving is network topology independent.

The second tier of the correlated multi-query processing, the correlated data collection at sensor nodes, is compared to basic data collection without using proxies. The number of data transmissions saved using correlated data collection also depends on how many intermediate sensors are between a sensor node and its proxy sensor. Similarly, to make the measurement

network topology independent, when a query acquires sensor data from its proxy node, one data transmission is considered to be saved.

The reduced number of sensor data transmissions can in turn lead to less contention or collision among message transmissions. The focus, however, is on the number of sensor data transmissions for query processing, rather than the number of messages being transmitted in the network. For this purpose, a simple simulator is developed using C++. For simplicity, in the simulation, each data attribute at a sensor node is assigned a value uniformly selected within its possible range. A total number of 100 sensor nodes are given in the network. The simulation time is divided into 10 epochs. Queries are randomly generated at the base station within an epoch. Each simulation is run 500 times. In each run of the simulation, sensor data is randomly generated.

There are two parameters which can affect the amount of shared sensor data among queries: the number of attributes and the number of queries. Queries specify what sensor data to collect by defining constraints over data attributes. The more data attributes sensed in a sensor node, the more types of constraints can be defined by a query, which usually leads to less data being shared among queries. On the other hand, a larger number of queries provides more opportunities for data to be shared among queries. In the simulation, different numbers of attributes a sensor senses and different numbers of queries are used to simulate different amounts of shared data among queries.

6.6.2 Performance Comparison

Figure 47 presents the results for number of data transmissions saved while different number of data attributes are sensed at sensors in the network. In the simulation results, each data point is the average value of 500 runs. Many of these figures also include a confidence interval of 95%. However, the interval is so small that it is invisible in almost all of the figures. The results show that as the number of attributes increases, the amount of shared sensor data identified by correlated multi-query processing at the base station decreases, as does the number of saved data transmissions.

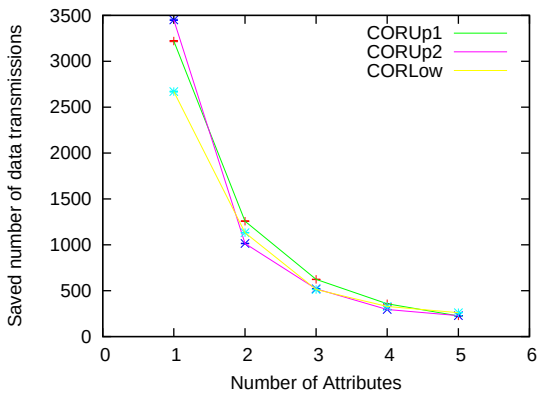

Figure 47: Number of data transmission saved by correlated multi-query processing at the base station for 20 queries

Figures 48 to 52 present the number of data transmissions saved in the simulations, in which sensors sense 1 to 5 attributes, respectively. In these simulations, the number of queries varies from 10 to 50 in intervals of 10. The results show that the saved number of data transmissions does not always increase as the number of queries increases. This is due to two reasons. First, the queries are randomly generated, so a larger number of queries does not necessarily mean more sensor data are shared between queries. Second, increasing the number of queries by 10 might not be enough to generate more shared sensor data between queries. A trend of increasing in the number of saved data transmissions can be observed at intervals of 20 in terms of number of queries.

It is also interesting to know, in these simulations, how much is saved in terms of percentages. Figures 53 to 57 show the percentage of data transmissions saved to the overall number of data transmissions needed for the same set of queries. As shown from these results, the percentage varies in different cases. It can be as significant as 48%, or as low as 3%. The results suggest that significant saving can be achieved using correlated query processing at the base station when a large amount of data is shared among queries.

From the results in Figures 48 to 57, none of the three correlation approximations, namely "CORLow", "CORUp1" or "CORUp2" outperforms the others in every scenario. When one

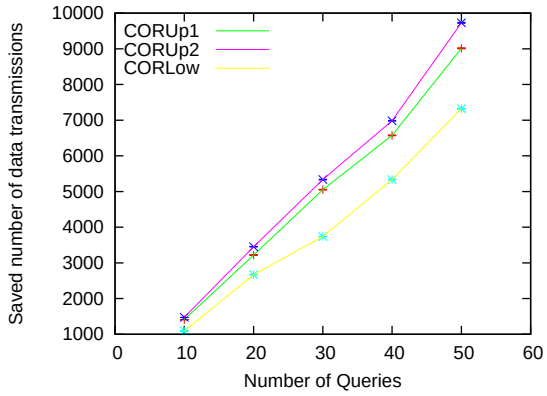

Figure 48: Number of data transmissions saved by correlated multi-query processing at the base station when 1 attribute is sensed

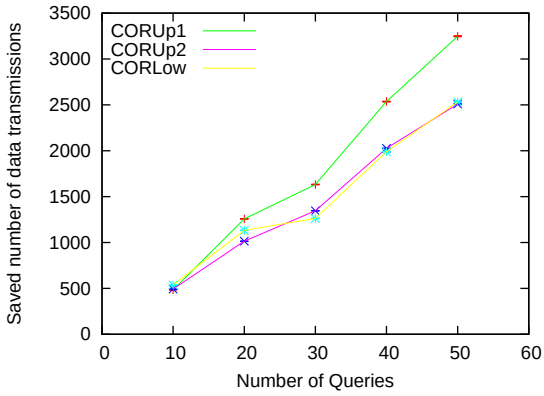

Figure 49: Number of data transmissions saved by correlated multi-query processing at the base station when 2 attributes are sensed

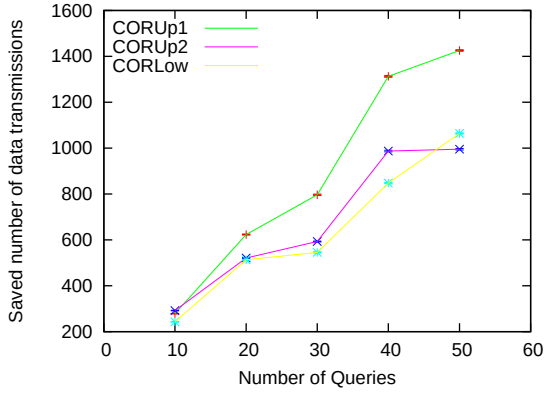

Figure 50: Number of data transmissions saved by correlated multi-query processing at the base station when 3 attributes are sensed

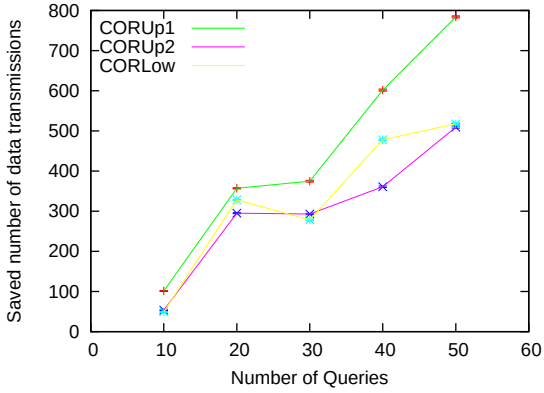

Figure 51: Number of data transmissions saved by correlated multi-query processing at the base station when 4 attributes are sensed

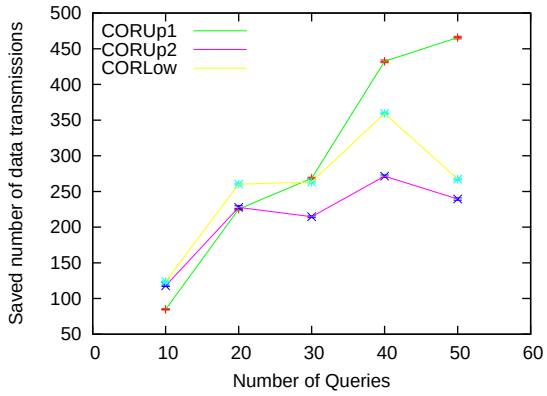

Figure 52: Number of data transmissions saved by correlated multi-query processing at the base station when 5 attributes are sensed

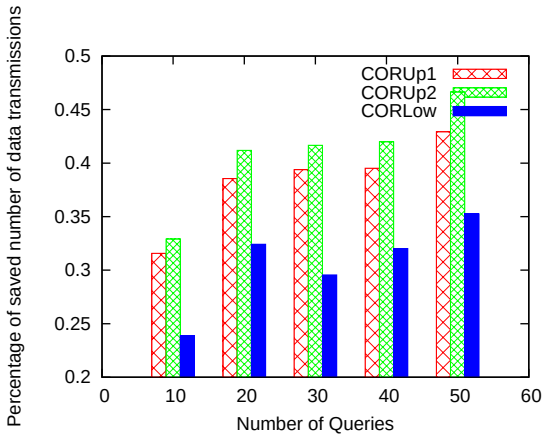

Figure 53: Percentage of data transmissions saved by correlated multi-query processing at the base station when 1 attribute is sensed

data attribute is sensed in the network, "CORUp2" reduces the number of data transmissions for multiple query processing the most, but not in other cases. However, being aggressive, i.e. using "CORUp1" and "CORUp2", leads to more data transmissions being saved in most cases.

Because the correlated query processing at the base station cannot completely eliminate the redundancy of data communications in the network for query processing, correlated data collection at sensor nodes is proposed to further reduce the number of data transmissions in sensor nodes. Figures 58 to 62 show a large number of data transmissions can be saved by using proxies for data collection.

Figures 63 to 77 compare the number of data transmissions saved by correlated query processing at base station to the number of data transmissions saved by correlated data collection at sensor nodes. The results show that when only one attribute is sensed at sensors, the correlated query processing at the base station saves a larger number of data transmissions than the correlated data collection at the sensor nodes because most correlation

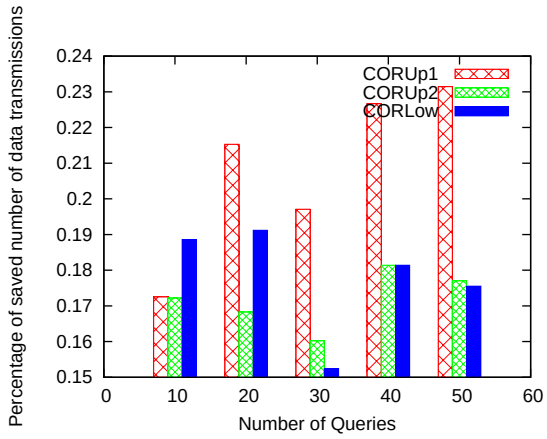

Figure 54: Percentage of data transmissions saved by correlated multi-query processing at the base station when 2 attributes are sensed

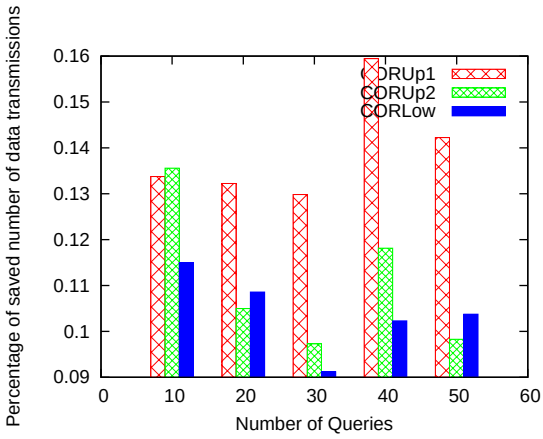

Figure 55: Percentage of data transmissions saved by correlated multi-query processing at the base station when 3 attributes are sensed

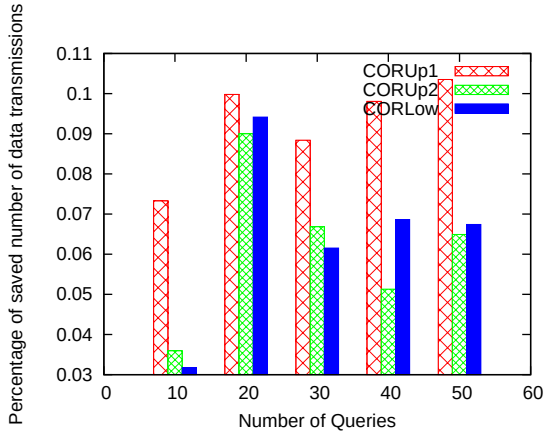

Figure 56: Percentage of data transmissions saved by correlated multi-query processing at the base station when 4 attributes are sensed

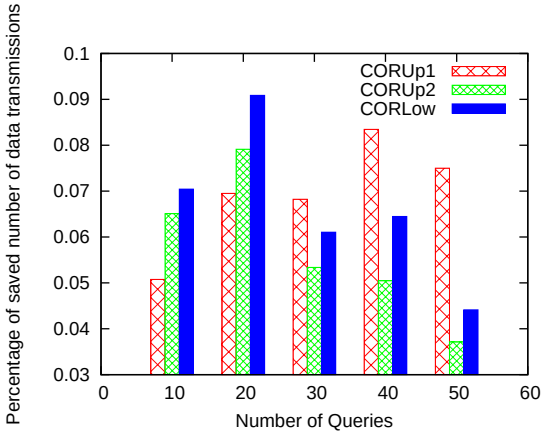

Figure 57: Percentage of data transmissions saved by correlated multi-query processing at the base station when 5 attributes are sensed

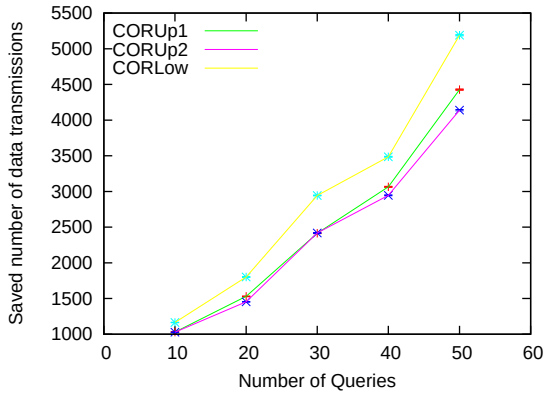

Figure 58: Number of data transmissions saved by correlated data aggregation at sensor nodes when 1 attribute is sensed

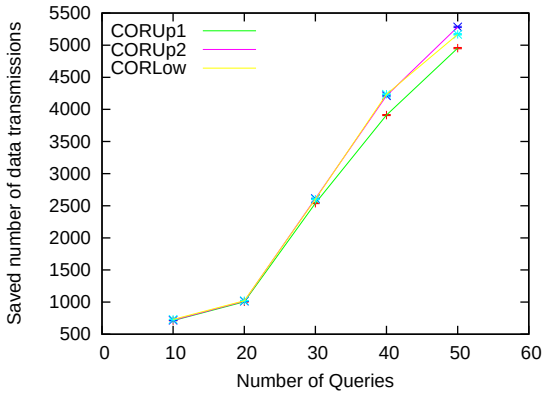

Figure 59: Number of data transmissions saved by correlated data aggregation at sensor nodes when 2 attributes are sensed

has been identified at the base station. In other cases, however, a large amount of correlation is left among queries by the base station, and correlated data collection at the sensor nodes reduces more data transmissions than correlated query processing at the base station.

6.6.3 Communication Overhead

Correlated multi-query processing at the base station uses shared intermediate views to reuse common data between queries. It seems that the base station needs to send the SIVs with rewritten queries to the sensor network for processing, which may incur communication overhead since more queries must be sent. However, a shared intermediate view $SIV(q_i, q_j)$ can be constructed from q_i and q_j, as well as the rewritten queries of q_i and q_j. The base station can send q_i and q_j together, and each sensor checks if its data satisfies $q_i \cap q_j$, $q_i \cap \neg q_j$ or $\neg q_i \cap q_j$. In this way, a collection/aggregation tree can be constructed for $SIV(q_i, q_j)$ and the rewritten queries of q_i and q_j by only sending queries q_i and q_j.

In the correlated data collection scheme at sensor nodes, a proxy node is established at almost no additional communication overhead since the data must be aggregated anyway.

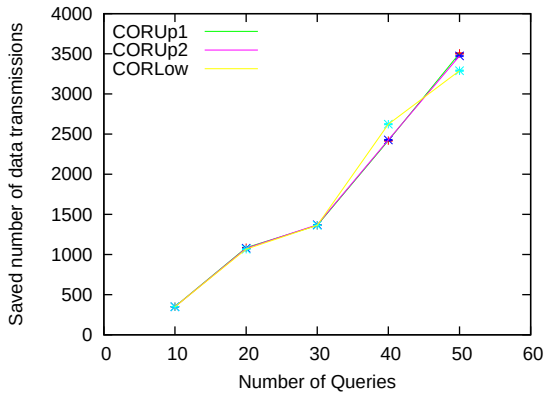

Figure 60: Number of data transmissions saved by correlated data aggregation at sensor nodes when 3 attributes are sensed

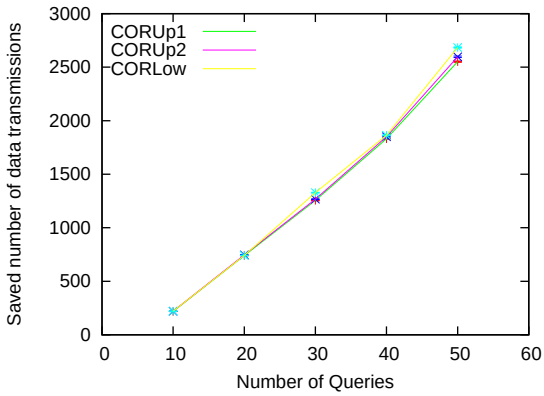

Figure 61: Number of data transmissions saved by correlated data aggregation at sensor nodes when 4 attributes are sensed

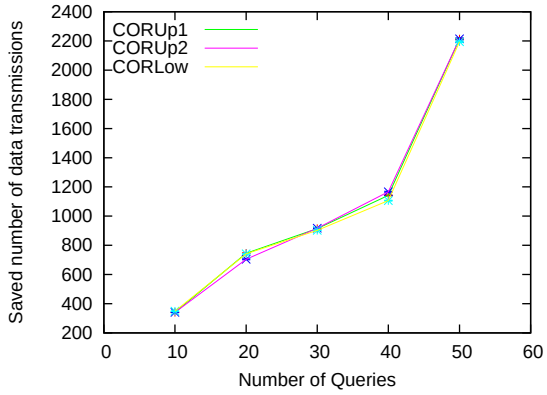

Figure 62: Number of data transmissions saved by correlated data aggregation at sensor nodes when 5 attributes are sensed

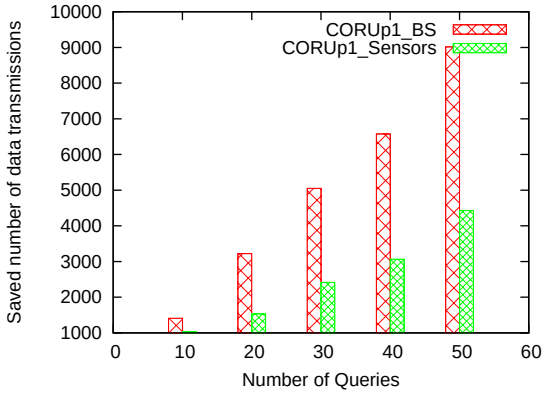

Figure 63: Number of data transmissions saved using CORUp1 when 1 attribute is sensed

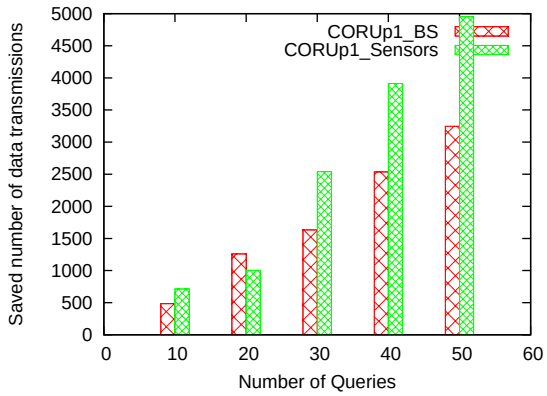

Figure 64: Number of data transmissions saved using CORUp1 when 2 attributes are sensed

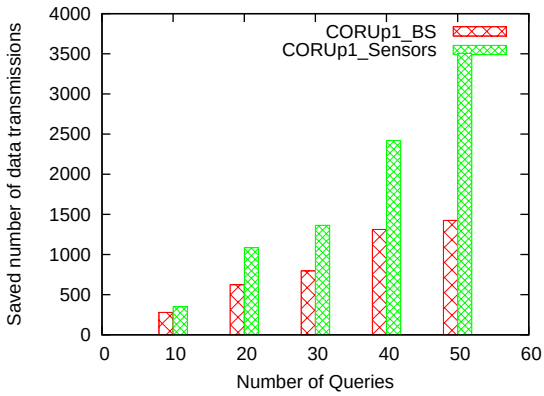

Figure 65: Number of data transmissions saved using CORUp1 when 3 attributes are sensed

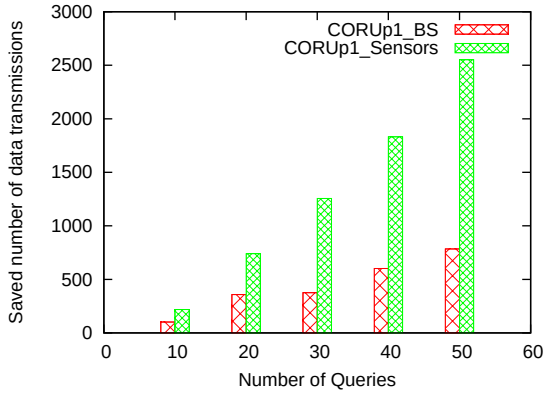

Figure 66: Number of data transmissions saved using CORUp1 when 4 attributes are sensed

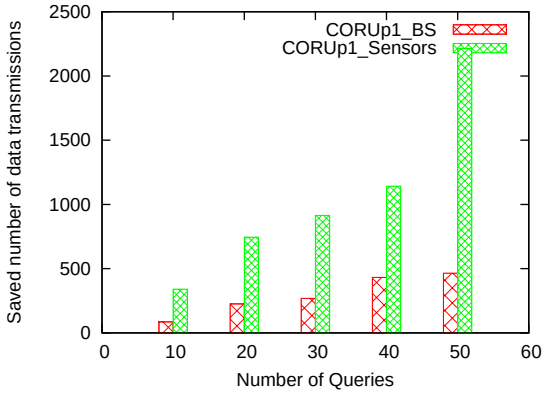

Figure 67: Number of data transmissions saved using CORUp1 when 5 attributes are sensed

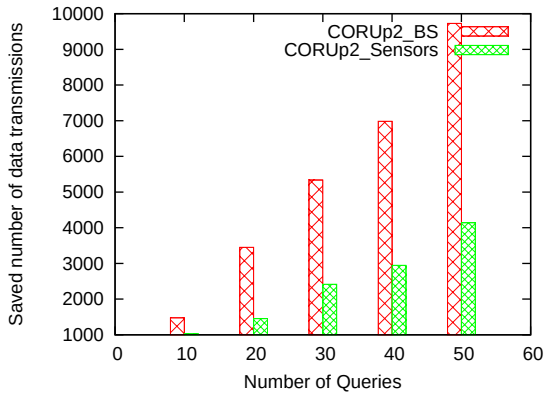

Figure 68: Number of data transmissions saved using CORUp2 when 1 attribute is sensed

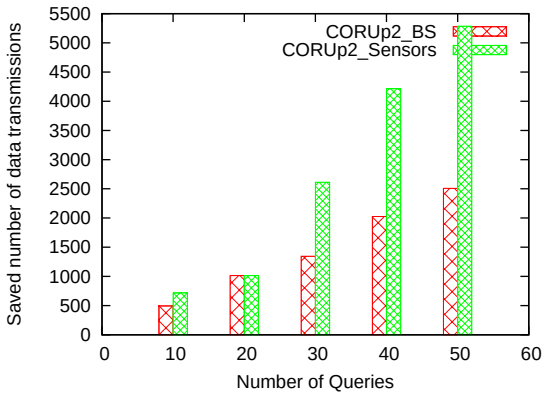

Figure 69: Number of data transmissions saved using CORUp2 when 2 attributes are sensed

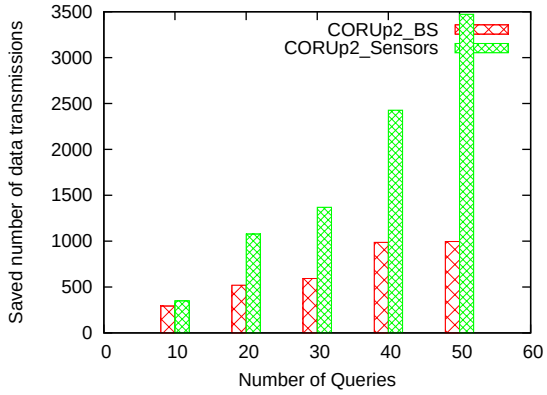

Figure 70: Number of data transmissions saved using CORUp2 when 3 attributes are sensed

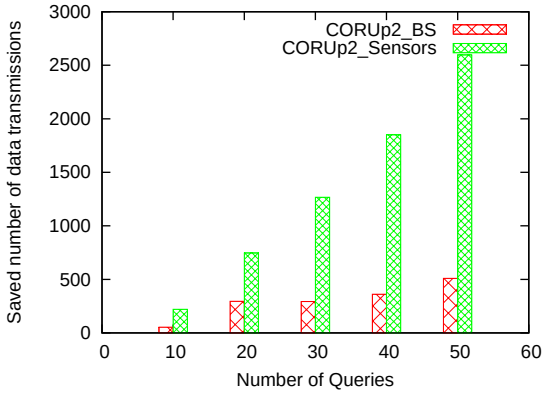

Figure 71: Number of data transmissions saved using CORUp2 when 4 attributes are sensed

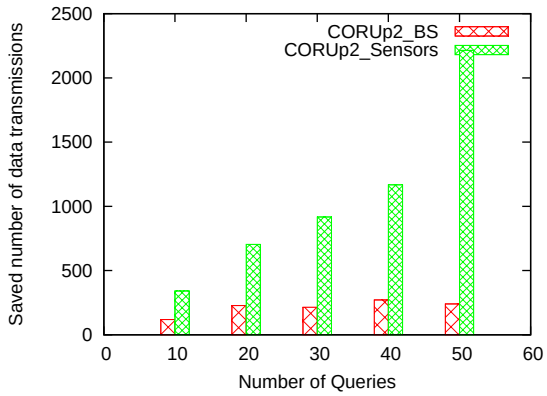

Figure 72: Number of data transmissions saved using CORUp2 when 5 attributes are sensed

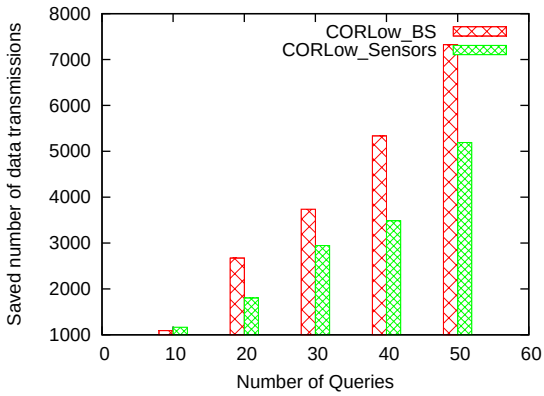

Figure 73: Number of data transmissions saved using CORLow when 1 attribute is sensed

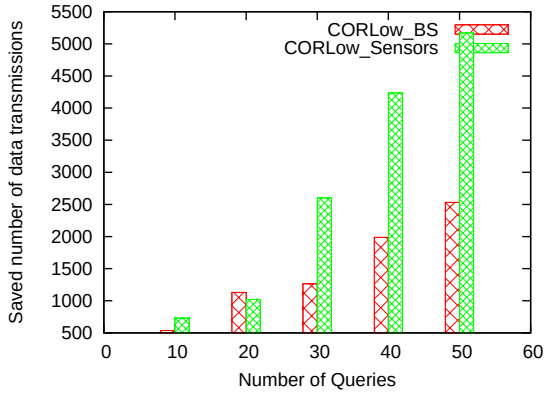

Figure 74: Number of data transmissions saved using CORLow when 2 attributes are sensed

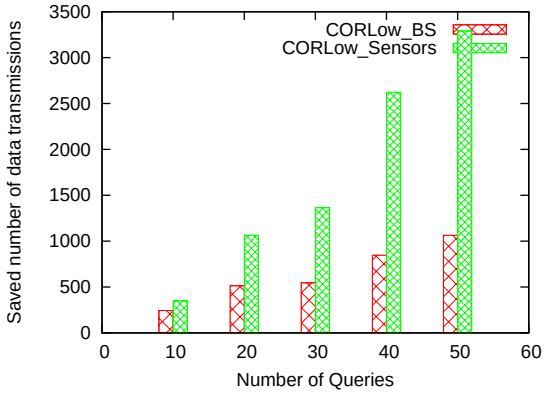

Figure 75: Number of data transmissions saved using CORLow when 3 attributes are sensed

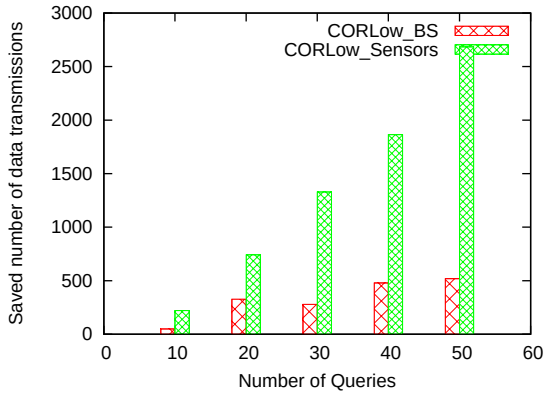

Figure 76: Number of data transmissions saved using CORLow when 4 attributes are sensed

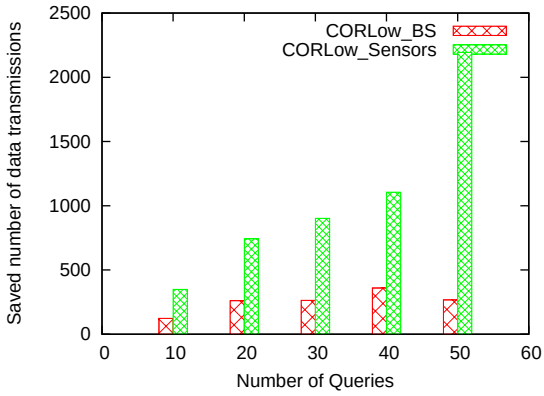

Figure 77: Number of data transmissions saved using CORLow when 5 attributes are sensed

A sensor node doesn't need to know what its proxy node is so no message needs to be sent back from the proxy node to the sensor node once the data aggregation is done. The only communication overhead is when the proxy node fails. One of its neighboring sensors must update the sensor node that its proxy has failed and a new proxy must be selected if needed. This process does incur additional communications. However, the overhead is not significant if sensor failures are not frequent.

6.7 SUMMARY

This chapter details how to process the correlation among semantic views. A numerical model is developed to estimate the volume of data shared between two queries. Based on these estimation values, a set of shared intermediate views, which captures the actual set of data shared between queries, are constructed. These shared intermediated views are processed only once and their results are reused to provide results for the original queries from which

the shared intermediate views are constructed. The goal is to eliminate the overlapping of sensor data among queries so that each sensor only need to transmit/aggregate its data once for processing semantic views. The sensor nodes, after receiving queries, use correlated data collection to reduce the number of data transmissions for correlated queries. In correlated data collection, each sensor node stores its data to a proxy node which is closer to the base station. It is shown through simulations that these two techniques can effectively reduce the communication cost of data processing for correlated queries among semantic views.

7.0 LOCATION DISCOVERY FOR SEMANTIC VIEW PROCESSING

In this chapter, we first introduce some background of location discovery using multi-lateration in sensor networks, and then define out-of-range information and show how to use out-of-range information to resolve location ambiguities in several scenarios. Based on these observations, the multi-lateration scheme is modified to utilize out-of-range information for location discovery when sufficient reference nodes are not available. The design is validated through simulations at the end of the chapter.

7.1 PROBLEM STATEMENT

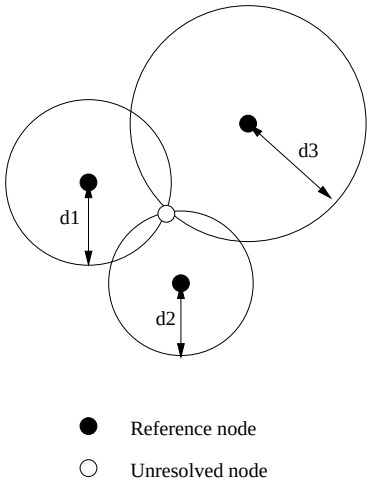

Figure 78: An example of trilateration

In location discovery using multi-lateration, a sensor obtaining its location through external devices is called an anchor or reference sensor node. Unknown sensors exchange messages with neighboring reference sensor nodes. From these messages, unknown sensors can find out where these neighboring reference sensors are. They can also compute the distances to the reference sensor nodes through received signal strength or time of transmission. As shown in Figure 78, in a two dimensional space, given the locations of three neighboring reference nodes and distances to these reference nodes, an unknown sensor node can compute its own location if at least one of these reference nodes is not collinear with the rest of the reference nodes. Since the distance between sensors is not directly measured but estimated, inaccuracies may be introduced in the location discovery process. When more than three neighboring reference nodes are available, an unknown sensor can minimize the location estimation error using a maximum likelihood estimator [106]. If fewer than three reference sensors exist in the neighboring nodes, an unknown sensor cannot uniquely determine its own position.

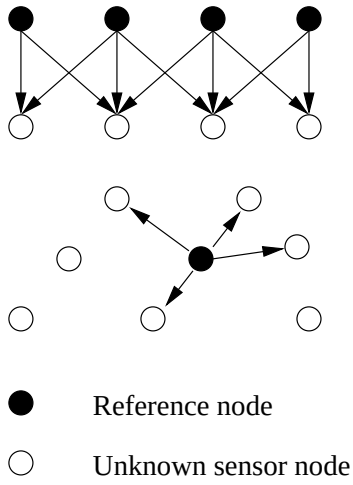

Figure 79: Location discovery through multi-lateration

Using the atomic multi-lateration process, some unknown sensor nodes can estimate their locations from an initial set of reference nodes. These nodes then act as if they are newly available reference nodes and propagate their locations to other sensors in the network. The process continues until all unknown sensors with three or more neighboring reference sensor nodes or resolved sensor nodes estimate their locations. Apparently, the ultimate number

of sensors which can resolve their locations after the process depends on the number and locations of the initial reference sensor nodes. In the example shown in Figure 79, all sensors can discover their locations using the five initial reference sensors. With simple analysis, it is also easy to see that all sensors can discover their locations with the four reference sensors on the top of the network topology.

Given that each reference sensor can cost much more than a normal sensor node and consumes significantly more energy in obtaining its location through external devices, it is preferable to use as few reference nodes as possible. It is necessary then to look at possible ways to minimize the number of initial reference sensors in location discovery using multilateration schemes. In cases where sensors can be deployed at specific locations, the reference sensors can be strategically positioned in the network such that the minimal number of sensors is needed to enable the location discovery of all other sensor nodes. However, in many scenarios, it is very difficult, if not possible, to deploy sensors at specific locations. Therefore, approaches which don't rely on the specific deployment of initial reference sensors will be investigated.

7.2 OUT-OF-RANGE INFORMATION

The definition for Out-of-Range information is based on the following observation: if two sensor nodes, $N1$ and $N2$, cannot hear from each other, then the distance between them must be larger than $r1$, the transmission range of $N1$, and $r2$, the transmission range of $N2$. In reality, the transmission range of a sensor may be irregular [140], so the transmission range of a sensor node may depend on where the destination is. However, the observation is still valid if $r1$ and $r2$ is replaced with the minimum range over all directions that $N1$ and $N2$'s signal propagates. The observation is formally defined as follows:

$$N1,\ N2 \textit{ are not neighboring nodes} \ \Rightarrow$$
$$dist(N1, N2) > \max(\min(r1_\alpha), \min(r2_\alpha)) \ . \tag{7.1}$$

For simplicity, let r be $\max(\min(r1_\alpha), \min(r2_\alpha))$ in the rest of the chapter.

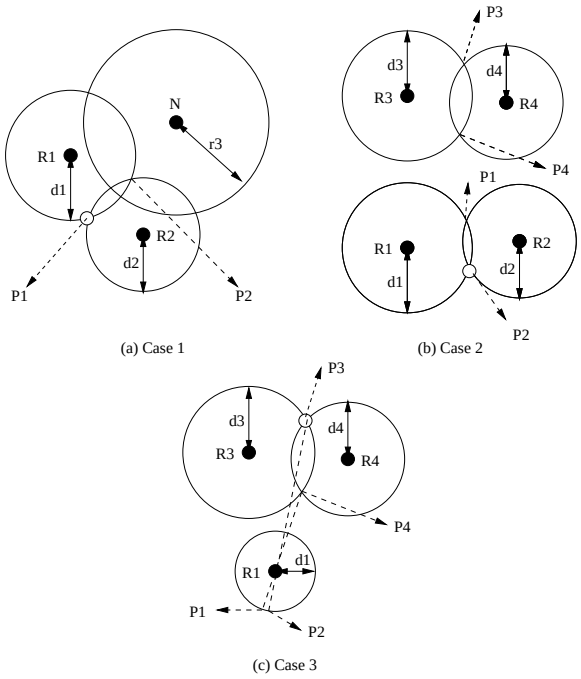

Figure 80: Using Out-of-Range information to resolve an unknown node's position

The following sections illustrate how to utilize Out-of-Range information to resolve an unknown node's position by introducing several scenarios. In all of the following cases, it is assumed that node N is out of the transmission range of node U. Furthermore, it is assumed that the network is connected. Hence, N can reach U through multi-hop flooding. For simplicity, the other nodes in the network are not shown in Figure 80.

7.2.1 Case 1: N is a reference node and U has two neighboring reference nodes

In the scenario presented in Figure 80(a), the unknown node has two neighboring reference nodes, $R1$ and $R2$. The distance measured from the unknown node to $R1$ and $R2$, is $d1$ and $d2$ respectively. Obviously, there are two possible positions that the unknown node might be, P1 and P2, given only this knowledge. If another reference node, N, exists in the network and the unknown node cannot hear from N. Furthermore, P2 is in N's transmission range

while P1 is not, then it can be inferred that the unknown node can only reside in P1 because it would hear from N if it were at P2.

Let (x_N, y_N) be the coordinate of the reference node N, (x_1, y_1) and $(x_1^{'}, y_1^{'})$ be the two possible positions of the unknown node U, and r be the minimum transmission range of N, N can resolve U's location if:

$$\begin{aligned} & (\sqrt{(x_N - x_1)^2 + (y_N - y_1)^2} > r \quad \&\& \\ & \sqrt{(x_N - x_1^{'})^2 + (y_N - y_1^{'})^2} \leq r) \\ \| \quad & (\sqrt{(x_N - x_1^{'})^2 + (y_N - y_1^{'})^2} > r \quad \&\& \\ & \sqrt{(x_N - x_1)^2 + (y_N - y_1)^2} \leq r) \ . \end{aligned} \tag{7.2}$$

7.2.2 Case 2: N is an unknown node with two neighboring reference nodes and U has two neighboring reference nodes

In the scenario described in Figure 80(b), an unknown node, N, has two neighboring reference nodes, $R3$ and $R4$. Given its distance to $R3$ and $R4$, $d3$ and $d4$, N can calculate its two potential locations: P3 and P4. Similarly, the unknown node, U, has two neighboring reference nodes, $R1$ and $R2$, and computes its own possible positions: P1 and P2. Furthermore, the distance between P3 and P1 and the distance between P4 and P1 are smaller than U's minimum transmission range, r. Based on the fact that N is not a neighbor of U, U can determine that it must be located at P2.

In general, an unknown node N, located at either (x_1, y_1) or $(x_1^{'}, y_1^{'})$, can determine the location of another unknown node, U, located at either (x, y) or $(x^{'}, y^{'})$, if:

$$\begin{aligned} & (\sqrt{(x - x_1)^2 + (y - y_1)^2} \leq r \quad \&\& \\ & \sqrt{(x - x_1^{'})^2 + (y - y_1^{'})^2} \leq r) \\ \| \quad & (\sqrt{(x^{'} - x_1)^2 + (y^{'} - y_1)^2} \leq r \quad \&\& \\ & \sqrt{(x^{'} - x_1^{'})^2 + (y^{'} - y_1^{'})^2} \leq r) \ . \end{aligned} \tag{7.3}$$

7.2.3 Case 3: N is an unknown node with one neighboring reference node and U has two neighboring reference nodes

Similar to case 2, the scenario presented in Figure 80(c) also describes how an unknown node, N, can help to determine the location of another unknown node, U. This case differs from case 2 in that the unknown node, N, only has one neighboring reference node. In Figure 80(c), U is located at either P3 or P4. N is located at $d1$ away from the reference node $R1$. P1 is the farthest point from P4 among all the possible locations N might be. If P4P1 is smaller than r, it can be easily concluded that U must reside at P3, because otherwise U would be a neighboring node of N.

Generally, consider an unknown node U, located at either (x, y) or $(x^{'}, y^{'})$. An unknown node N, which is d away from its neighboring reference node $R1$, can determine U's position under the following condition:

$$\begin{aligned} & \sqrt{(x-x_0)^2+(y-y_0)^2} \le r,\ \forall (x_0, y_0), \\ & \sqrt{(x_0-x_{R1})^2+(y_1-y_{R1})^2} = d \\ \| \quad & \sqrt{(x'-x_0)^2+(y'-y_0)^2} \le r,\ \forall (x_0, y_0), \\ & \sqrt{(x_0-x_{R1})^2+(y_1-y_{R1})^2} = d\ . \end{aligned} \tag{7.4}$$

Condition 7.4 can be simplified into the following equivalent condition after a short derivation:

$$\begin{aligned} & \sqrt{(x-x_{R1})^2+(y-y_{R1})^2} \le r-d1 \\ \| \quad & \sqrt{(x'-x_{R1})^2+(y'-y_{R1})^2} \le r-d1\ . \end{aligned} \tag{7.5}$$

7.3 LOCALIZATION SCHEME

In the proposed location discovery scheme, anchor nodes disseminate their positions to neighboring unknown sensor nodes. An unknown sensor node measures its distance to each of the neighboring reference/anchor nodes respectively. It's assumed that the distance between two sensors can be estimated using methods such as RSSI or ToA. If more than three neighbor nodes are reference nodes, an unknown node then estimates its own location using trilateration. In addition, the least square method is used to refine a sensor node's location in an over determined system. Otherwise, the unknown sensor node sends messages to non-neighboring nodes to check if they can help to resolve its location using Out-of-Range information. Once its location is resolved, an unknown node becomes a reference node and disseminates its position to other unknown nodes in the network to enable the continuation of the location discovery process.

Figure 81 presents the major steps of the localization process executed at a reference node R. The reference node starts the localization process by announcing its location to neighboring nodes. It then keeps waiting for messages from other nodes. Based on the type of message received, the reference node, R, responds as follows:

- If a "Location_help" message for U is received, R simply discards this message if it has already processed the help request from U. Otherwise, R checks condition 7.2 and sends "Location_help_reply" to U if it can determine the location of U using "Out-of-Range" information. If R cannot utilize its "Out-of-Range" information to uniquely locate U's position, R decreases the TTL of the "Location_help" message by one and forwards the "Location_help" message to its neighbors if the TTL is still bigger than zero

The main steps of the localization scheme at an unknown node are described in Figure 82. Each unknown node U basically waits for messages from other nodes and acts according to the type of message as follows:

- After receiving a "Location_announce" message from R, U puts R's id and location information in its reference node table. U also starts a timer t if the "Location_announce" is the first announce message it receives

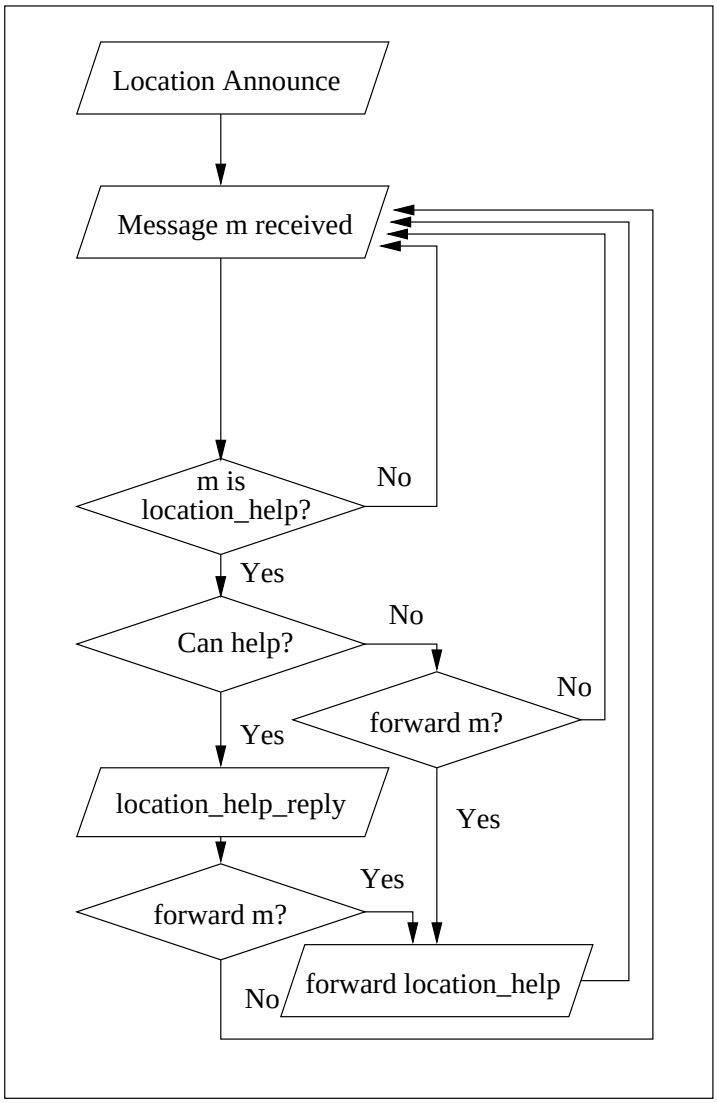

Figure 81: Localization algorithm at reference nodes

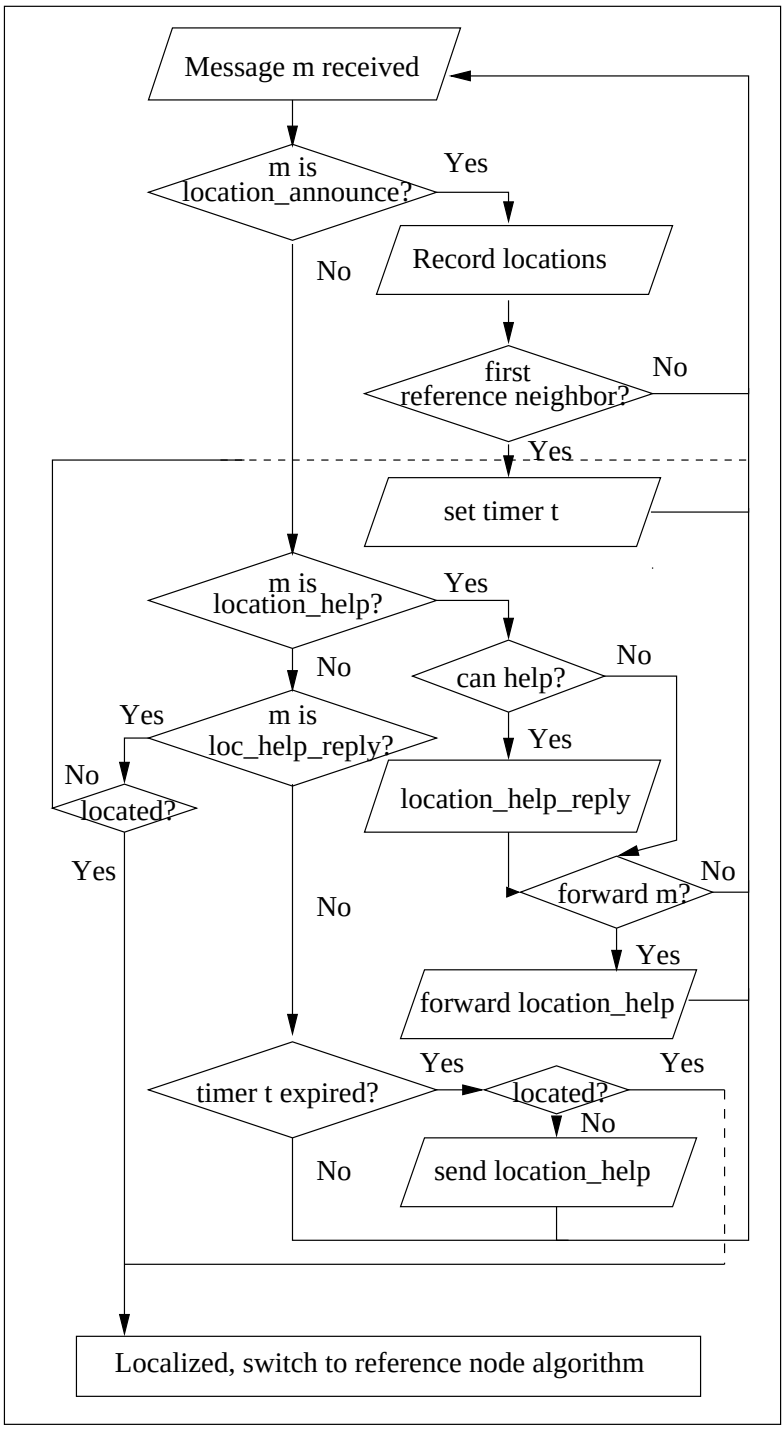

Figure 82: Localization algorithm at unknown nodes

- If the timer, t, expires and U still cannot resolve its location, U initializes the TTL of the "Location_help" message to be h and sends it out to its neighboring nodes
- Upon receiving a "Location_help" message for another unknown node $U1$, node U simply discards this message if it has already processed the message. Otherwise, it extracts the reference node information in the message and determines if its location information can help to determine $U1$'s position. The conditions in case 2 and 3 in section 7.2.2 and 7.2.3 are checked for based on the role of U and $U1$. If U's information can be utilized to determine $U1$'s location, U replies to $U1$ with a "Location_help_reply" message. Otherwise, U decreases the TTL of "Location_help" by one and forwards the "Location_help" to its neighbors if the TTL is still bigger than zero
- After receiving a "Location_help_reply", U extracts the reply information and resolves its location. If U's location is resolved, U becomes a reference node and executes the localization algorithm for reference nodes

7.4 SIMULATION RESULTS

7.4.1 Methodology

The main metric that we are interested in the simulations is how many number of sensors can be resolved after the location discovery process completes. In addition, we also want to know how many initial reference sensors are needed to discover the locations of all sensors nodes in a high density network. The simulation is developed using Glomosim 2.03. The location discovery scheme using Out-of-Range information is simulated at the application layer. The initially configured anchor nodes start to broadcast location information at the beginning of simulation. All the messages are delivered using UDP and retransmitted three times if not received. Each node maintains a neighbor table so that it knows if it is out of the range of another node.

A various levels of densities is simulated by modifying the number of sensors in a field of 100X100m. The average node degree of these networks are listed in Table 10. In the

Table 10: Average node degree deg in a network of n nodes for location discovery

n	30	40	50	60	65	70	75	80	90	100
deg	2.86	3.95	5.6	5.8	6.4	7.0	7.1	7.2	9.2	9.5

simulations, a transmission range of 20m is considered. Furthermore, for simplicity, it is assumed that all sensors have the same value of transmission range.

7.4.2 Effect of h

The value of h is critical to the performance of the proposed scheme. On the one hand, a large value of h allows an unknown node to reach more sensors for help and thus have a higher chance of resolving its location ambiguity. However, on the other hand, it also leads to a high level of communication overhead. Therefore, care must be taken in configuring the value of h to ensure a high possibility of location discovery at a low cost of communications.

Figure 83 presents the number of resolved nodes after the location discovery completes in networks with various levels of densities. In these scenarios, 3 nodes are initially configured as anchor nodes. The results show that the number of resolved nodes remains the same in most cases. It slightly increases in the network of 50 and 80 nodes when h increases from 2 to 3. The reason is that nodes multi-hop away may be geographically too far away from the unknown node U to provide any useful Out-of-Range information. The value of h is set to be 2 in the rest of the simulations.

7.4.3 Performance Comparison

The number of resolved sensors after the location discovery is completed is used to measure the effectiveness of the proposed scheme in comparison to the basic multi-lateration scheme. The number of additional nodes located using Out-of-Range information depends on the network connectivity and topology. Figure 84 presents the number of resolved nodes after location discovery with Out-of-Range and without Out-of-Range information in a set of

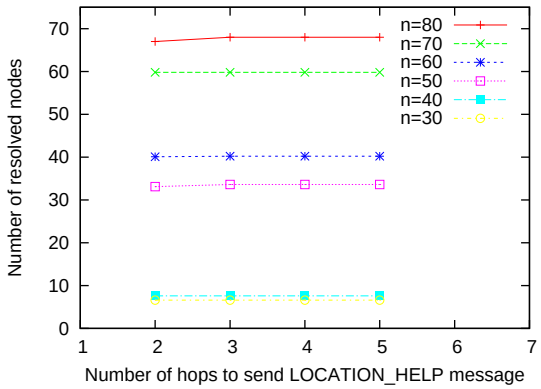

Figure 83: The effect of h in networks with four reference nodes

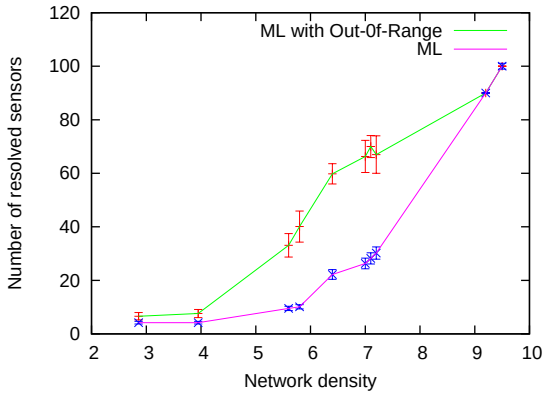

Figure 84: Number of resolved sensors after location discovery

scenarios. Each data point in Figure 84 is the average value of 20 runs. In each run, a set of three nearby sensors which are neighboring to at least one other sensor are chosen randomly as reference sensors to start the location discovery process. In addition to the average number of resolved sensors, the 5% confidence interval is also shown in the figure. The number of anchor nodes remains at 3 in the simulations.

When the average node degree is small, no Out-of-Range information can be used due to the lack of connectivity in the network. On the other hand, when the average node degree is large, no Out-of-Range information is needed since sensors can estimate their locations using reference nodes. In the other cases, the proposed scheme can locate more nodes than the basic multi-lateration scheme. The results show that as the sensor network connectivity starts to decrease, the Out-of-Range information can be used to locate more sensors in the network.

It is worth noting that the average number of resolved sensors does not always increase as the density increases in the network. This is due to the fact that nodes are uniformly placed over the entire area. With the uniform placement, the area is divided into a number of cells and nodes are randomly placed within each cell. A slight increase in the number of nodes can result in an additional cell with only few nodes placed in it. Selecting reference sensors from this additional cell could end up with very few sensors being resolved after the location discovery process. Nonetheless, a relative big increase of network density does result in an increase of number of resolved sensors after the location discovery process.

In addition to Figure 84 which shows the absolute number of resolved sensors, we also present the percentage of resolved sensors after location discovery process in networks with different levels of densities in Figure 85. The data in Figure 85 shows that with Out-of-Range information, a significant larger percentage of sensors can be located in the networks with medium range densities.

Next, Figure 86 shows how many anchor nodes are required in order to discover the locations of all sensors in the network when the network connectivity is high. As the figure shows, the number of anchor nodes required to discover all sensors can be reduced using Out-of-Range information in networks with low and medium density. In high density network, there is no need to use Out-of-Range information for location discovery since sensors can

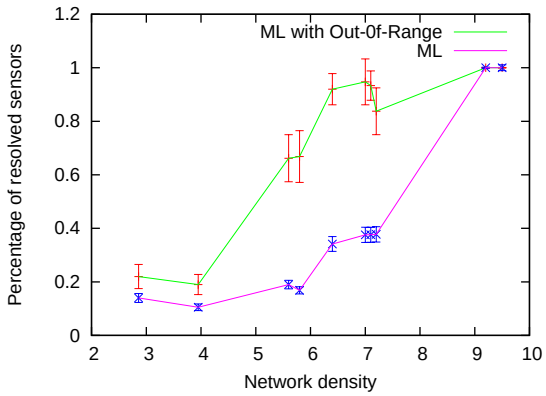

Figure 85: Percentage of resolved sensors after location discovery

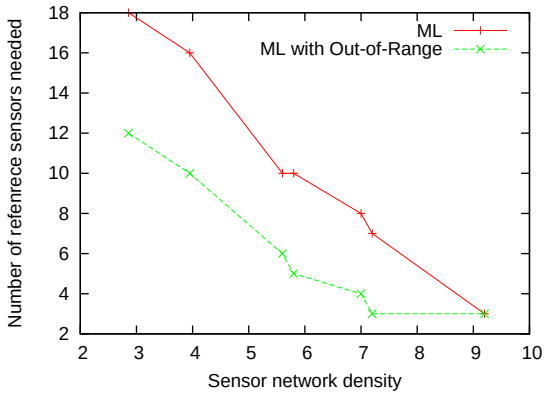

Figure 86: Number of anchor sensors required to resolve locations of all sensors in the network

gain sufficient information about reference sensor nodes and compute their locations using multi-lateration.

7.5 SUMMARY

This chapter presents the location discovery scheme used in the framework for semantic view processing. The location discovery scheme is based on "Out-of-Range" information and it is shown how this information can be used to resolve sensor location ambiguities when combined with multi-lateration. The simulation results show that with out-of-range information, fewer reference nodes are needed to locate sensors in the network, which in turn reduces cost and energy consumption of the whole network since reference nodes are usually much more expensive and consume more energy.

8.0 SECURE MESSAGE EXCHANGE FOR SEMANTIC VIEW PROCESSING

In wireless sensor networks, sensors communicate with each other using open mediums such as electromagnetic and acoustic waves. The data being transmitted in wireless signals can be easily captured by attackers through eavesdropping and traffic analysis if communications among sensors are not encrypted. Even if links between sensors are secured, attackers can still attack sensor nodes. Once a sensor is compromised, its data is revealed to attackers if it is not tamper resistent. In addition, data communications through this sensor for other sensors may also be revealed to attackers.

To secure message exchanges for semantic view processing, an end-to-end pairwise key establishment scheme based on key pre-distribution is presented. This scheme allows any two sensors to set up a common symmetric key after key pre-distribution and path key establishment. These keys are then used to protect data communication links between sensors against packet eavesdropping and traffic analysis by attackers. The scheme also protects data communications among normal sensors from being exposed to compromised sensors.

8.1 ATTACK MODEL

We consider an attack model as follows:

- Attackers can eavesdrop all data communications among sensors through packet sniffing and extract data using traffic analysis.

- Sensors are not tamper resistent.
- Attackers only have limited resources to attack sensor nodes and can randomly compromise a certain number of sensors up to a threshold value.

8.2 END-TO-END PAIRWISE KEY ESTABLISHMENT

Giving the attack model above, our goal is to secure data communications among sensors against eavesdropping attack and compromised sensor nodes. We first briefly describe a random key pre-distribution scheme, on which our scheme is based.

8.2.1 Key Pre-distribution Scheme and Path Key Exposure

In the random key pre-distribution scheme proposed in [28], each node is loaded with a key ring of a set of m keys randomly selected from a large pool of keys, P, before deployment. Two nodes exchange either key identifiers or challenges to discover common keys in their key rings. The common key is used to establish a secure communication link. Since only a small number of keys are loaded in each sensor, node pairs may not always share common keys. Nodes without a common key to other WSN nodes are required to negotiate symmetric keys through a secure path.

To illustrate the key pre-distribution process, consider the network depicted in Fig 87. In this network, as a result of the common key discovery phase, $N1$ shares a key with $N2$ but not with $N3$ or $N4$. Consequently, to communicate with $N3$, $N1$ establishes a secure path to $N3$, e.g., $N1 \rightarrow N2 \rightarrow N4 \rightarrow N3$, and sends a key K to $N3$ through the secure path. As it travels from $N1$ to $N3$, K is encrypted with K_{12}, K_{24} and K_{34}, respectively. Notice, however, that while the pairwise key K is supposed to be exclusively shared between $N1$ and $N3$, the need for successive encryptions and decryptions along the path causes the key to be exposed to the intermediate nodes $N2$ and $N4$. This may lead to potential security compromise if a node along the path is captured. We refer to this problem as the "path key exposure problem". The likelihood of security breaches caused by key exposure is not negligible when

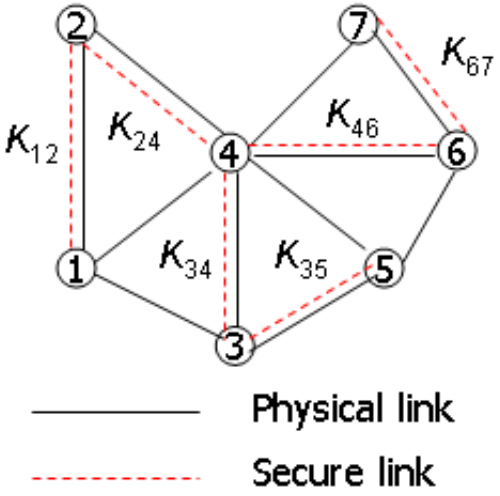

Figure 87: An example sensor network after shared key discovery.

random key pre-distribution is used to establish secure channels between a large number of WSN nodes. This is due to the fact that in key pre-distribution, the likelihood of a given node sharing a common key with a large number of other nodes is relatively small. Therefore, achieving secure communication between nodes which do not share a common key may lead to the establishment of a large number of secure paths for symmetric key negotiation and selection, thereby increasing the likelihood of security breaches.

8.2.2 End-to-End Pairwise Key Establishment using Multiple Secure Paths

As stated above, the path-key establishment exposes keys to each intermediate node along the routing path. In order to enhance the security of symmetric key establishment, we propose an end-to-end pairwise key establishment scheme which leverages multiple paths for key negotiation and establishment. The following assumptions are made in our scheme and security analysis.

- A Node Disjoint Routing Protocol (*NDRP*), such as the one described in [141], is used

to find node-disjoint paths[1].

- Secure links have been established among neighboring nodes in the network. This can be easily achieved using a secure key pre-distribution scheme such as the one described in [28].

Consider a network with a total number of n nodes, where a secure topology has been established using a shared-key discovery phase. Furthermore, assume that node $N1$ needs to set up a pair-wise key with another node $N2$. This can be achieved using the following steps:

- $N1$ uses $NDRP$ to find a set, PS, of node-disjoint secure paths to $N2$.
- Let $s = \mid PS \mid$, represent the size of the path set PS. Node $N1$ selects a key K and divides it into s fragments, $K_1, K_2 \ldots K_s$, such as $K = K_1 \cup K_2 \cup \ldots \cup K_s$, where $K_i \cup K_{i+1}$ represents the concatenation of K_i and K_{i+1}. Each fragment contains a sequence number, and the last fragment contains a Cyclic Redundancy Checksum code to verify the correctness of the assembled packet.
- $N1$ sends K_i through the i_{th} secure path.
- Upon receiving all s fragments of the key, node $N2$ reproduces the key K, and uses it for secure communication with $N1$.

Notice that the proposed scheme does not depend on the algorithm used to produce the key K. Consequently, any algorithm to produce a secure key can be used. An issue related to path selection, however, still remains to be addressed: how many node-disjoint paths must be discovered by the underlying $NDRP$ to ensure a high level of security with low level of overhead? The answer to this question depends on the number of nodes an attacker may compromise.

Assuming that an attacker can compromise at most x nodes in the network, a trivial solution would be to build a set of at least $x+1$ node-disjoint paths to achieve maximum security. However, if x is large, the network connectivity may not be "rich" enough to produce

[1] A large body of research work has focused on finding node-disjoint paths in a network. The focus of our scheme is not on building node-disjoint paths, but on what paths should be used to ensure the secure exchange of symmetric keys across a routing path. As such, the proposed scheme does not depend on a specific $NDRP$ algorithm. More work on finding node-disjoint secure paths or single-hop paths can be found in [40][41][42][43]

x+1 node-disjoint paths. Furthermore, a first look at the problem may lead to believe that the larger the set of node-disjoint paths is the more secure the scheme would be. Contrary to intuition, however, we will show that this is not always true. To this end, we develop a security analysis model to determine how secure a path-key establishment scheme is, given different sets of node-disjoint paths.

8.3 SECURITY ANALYSIS

In this section, we present a model to determine the probability that a path-key K is revealed if a node-disjoint secure path set, PS , is used. The notation used in this analysis are listed in Table 11. Assume that a node-disjoint path set, PS, is used. The key K is divided into

Table 11: Notation

K	: A pairwise key to be established
n	: Total number of nodes in the network[2]
x	: The maximum number of nodes an attacker can capture
X	: The set of nodes compromised. $\mid X \mid \leq x$
PS	: $\{P_1, P_2 \dots P_s\}$
s	: $\mid PS \mid$, number of secure paths in PS
l_i	: Intermediate hop counts of path P_i
N_{p_i}	: Intermediate nodes set of path $P_i, \forall 1 \leq i \leq s$
$N_{p_{s+1}}$	: The rest of nodes not in PS
NX_i	: The set of nodes compromised in N_{p_i}

$s = \mid PS \mid$ fragments. Each fragment of K is transmitted over a selected path in PS. The key K could be reproduced if and only if all s fragments are received. An attacker, trying to capture K, must compromise at least one node in each path in the set PS. Obviously, if $x < s$ then the probability of this event happening is zero. Otherwise, the probability of a path-key being exposed is the probability of selecting a set X out of n nodes such that

$X \cap N_{p_i} \neq \emptyset, \forall 1 \leq i \leq s$.

In this analysis, the security risk sr is defined as the probability of a key sent through a set of s paths, $\{P_1, P_2 \dots P_s\}$, being revealed when an attacker captures x out of n nodes. We denote this probability as $prob[\{P_1, P_2 \dots P_s\}, x, n]$. There are $\binom{n}{x}$ cases which result in x out of n nodes being randomly compromised. We need to compute how many of these cases will reveal the key K to an attacker. A single selection of x from n nodes could be represented as a (s+1)-tuple, $(NX_1, NX_2 \cdots NX_s, NX_{s+1})$, in which NX_i is the set of nodes captured from P_i and $NX_i \subseteq N_{p_i}$, $\forall\ 1 \leq i \leq s$. Now the cases in which the key K would be exposed are equivalent to those tuples $(NX_1, NX_2 \cdots NX_s, NX_{s+1})$ such that $NX_i \neq \emptyset, \forall 1 \leq i \leq s$. So

$$prob[\{P_1, P_2 \dots P_s\}, x, n] = \frac{\mid \{(NX_1 \cdots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\} \mid}{\binom{n}{x}} \tag{8.1}$$

A simple procedure to compute $\mid \{(NX_1 \cdots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\} \mid$, would be to first fix the number of nodes in NX_i and then determine how many possible cases exist. It is hard, however, to list all possible distributions of x nodes over these s+1 sets. We, therefore, use a different method to compute $\mid \{(NX_1 \cdots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\} \mid$. In the following, we describe the proposed method.

We index nodes in each path P_i from 1 to l_i. So $N_{p_i} = \{Ni_1 \dots Ni_{l_i}\}$. A new s tuple $S(j_1, j_2 \dots j_s)$ denotes the set of cases $\{(NX_1 \cdots NX_s, NX_{s+1}) \mid NX_i \neq \emptyset$, and the largest index of NX_i is $j_i, \forall 1 \leq i \leq s\}$. Since $j_i \leq l_i, \forall 1 \leq i \leq s$, there are totally $l_1 \times l_2 \cdots \times l_s$ possible tuples. For example, if n=6, x=3, PS=$\{P_1, P_2\}$ and $l_1 = 2,\ l_2 = 3$. We index nodes in P_1 *and* P_2 so that $P_1 = \{N1_1, N1_2\}, P_2 = \{N2_1, N2_2, N2_3\}$. The node not in PS is N_6. So S(1,1)=$\{(\{N1_1\}, \{N2_1\}, \{N_6\})\}$, S(2,2)=$\{(\{N1_2\}, \{N2_2\}, \{N_6\}), (\{N1_2, N1_1\}, \{N2_2\}, \emptyset)$, $(\{N1_2\}, \{N2_2, N2_1\}, \emptyset)\}$.

Lemma 1. *$S(j_1, j_2 \dots j_s) \cap S(j_1^{'}, j_2^{'} \dots j_s^{'}) = \emptyset$ if $\exists i, 1 \leq i \leq s$ and $j_i \neq j_i^{'}$*

[2]During the analysis, we assume that the source and destination node will not be compromised, thus they are not counted in the total number of nodes n.

Proof: Let j_i be the largest index of node captured in P_i, if $j_i \neq j_i^{'}$, then $NX_i \neq NX_i^{'}$. Thus $S(j_1, j_2 \ldots j_s) \cap S(j_1^{'}, j_2^{'} \ldots j_s^{'}) = \emptyset$.

Based on Lemma 1, we derive that

$$prob[\{P_1, P_2 \ldots P_s\}, x, n] = \frac{\sum_{j_1=1}^{l_1} \sum_{j_2=1}^{l_2} \cdots \sum_{j_s=1}^{l_s} \mid S(j_1, j_2 \ldots j_s) \mid}{\binom{n}{x}} \tag{8.2}$$

For the cases of $S(j_1, j_2 \ldots j_s)$, $N1_{j_1} \ldots Ns_{j_s}$ are compromised and the rest of the nodes could be captured either from the nodes not in PS or from those nodes in path P_i with smaller index than j_i. There are $n - \sum_{m=1}^{s} l_m$ nodes not in any path and $(\sum_{i=1}^{s} j_i) - s$ nodes in PS with smaller index, So $\mid S(j_1, j_2 \ldots j_s) \mid = \binom{n-\sum_{m=1}^{s} l_m - s + \sum_{i=1}^{s} j_i}{x-s}$. Thus

$$sr = prob[\{P_1, P_2 \ldots P_s\}, x, n] = \frac{\sum_{j_1=1}^{l_1} \sum_{j_2=1}^{l_2} \cdots \sum_{j_s=1}^{l_s} \binom{n-\sum_{m=1}^{s} l_m - s + \sum_{i=1}^{s} j_i}{x-s}}{\binom{n}{x}} \tag{8.3}$$

From Equation 8.3, we know that the security risk of our scheme is related to both the number of node-disjoint paths in PS and the hop count of each path in PS. A path set PS with more paths could be less secure. For example, if *n=100, x=10* then $prob[(3, 4, 8), 100, 10] = 0.045$ however $prob[(1, 5), 100, 10] = 0.038$, so path set $\{1, 5\}$ is more secure than $\{3, 4, 8\}$. In order to study the relation between the security risk of our scheme and the number of node-disjoint paths, we isolate the effect of path hop count by assuming that we always find another node-disjoint path with the same length. Fig 88 depicts the relation between *s, l* and *sr* in a 100 nodes network with 10 of them being captured randomly.

Fig 88 shows that although the security is improved by using more node-disjoint paths, the improvement by adding one more path decreases as the number of node-disjoint paths increases. For example, when one 6-hop path is used, the security risk decreases from 0.47 to 0.21 by adding one more 6-hop node-disjoint paths. However if more than 5 node-disjoint paths are used, the security risk decrease by adding one more path will be less than 0.01.

In a real network, it is unlikely that the node-disjoint paths found would be the same length. We take a path set $\{P_1, P_2, P_3, P_4, P_5, P_6\}$ with $l_1 = 1,\ l_2 = 3,\ l_3 = 4,\ l_4 = 6,\ l_5 = 8$ *and* $l_6 = 9$, and select these node-disjoint paths in the order of their hop count. Fig 89

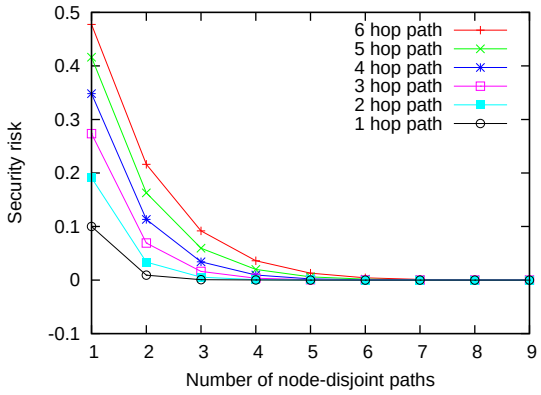

Figure 88: Security analysis of equal path hop count

shows how security risk varies with number of node-disjoint paths. We can see that the benefit from adding one more path is also decreasing. So after a number of node-disjoint paths are selected, it is not worth to find more node-disjoint paths. In other words, given security risk sr we need only use a certain number of node-disjoint paths with hop count constraint. The following properties of path-set can help to define a condition of path-set selection.

Lemma 2. $prob[\{l_1 \dots l_s\}, x, n] \leq prob[\{l \dots l\}_s, x, n]$[3] *if* $\sum_{j=1}^{s} l_j = s \times l$

Lemma 3. $prob[\{l_1 \dots l_s\}, x, n] \leq prob[\{l \dots l\}_s, x, n]$ *if* $\sum_{j=1}^{s} l_j \leq s \times l$

Lemma 2 can be proved by strong induction, and Lemma 3 follows from Lemma 2. See Appendix A.3 for details.

From Lemma 3, we know that given (s,l) a path-key sent through s node-disjoint paths set $\{l_1, l_2 \dots l_s\}$ can be compromised with probability less or equal than $prob[\{l \dots l\}_s, x, n]$ if $\sum_{j=1}^{s} l_j \leq s \times l$. Given sr and s, we can use Equation 8.3 to compute the maximum l such that $prob[\{l \dots l\}_s, x, n] \leq sr$ and $prob[\{l \dots l\}_{s+1}, x, n] > sr$. If node-disjoint path discovering algorithm *NDRP* can find such s node-disjoint paths, then we do not have to find more paths.

[3] $\{l \dots l\}_s$ means there are s paths with the same hop count l in this set.

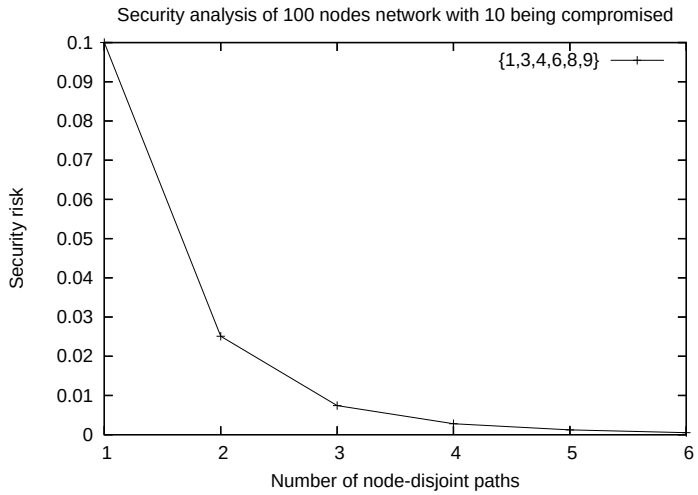

Figure 89: Security analysis of a real path set

8.4 OVERHEAD ANALYSIS

There are two kinds of overhead in our scheme: computational overhead and communication overhead. Given a security risk sr, number of nodes n and maximum number of nodes x an attacker could compromise, we compute a sequence of tuples (s, l) and use these tuples to determine when *NDRP* stops. It could be done off-line before sensors are deployed and a table of (s, l) could be loaded into each sensor.

In our scheme, the *NDRP* will need to find multiple node-disjoint paths for key establishment. It will incur extra routing overhead. However since the path-key establishment only needs to be run once for data communications unless the path-key is revoked and a new key needs to be negotiated. Generally, the path-key establishment occurs once during a long period of time for a pair of nodes. Also because we only send a small piece of key information through each path, every path is used for a very short period of time. So path maintenance is unnecessary if mobility is not extremely high.

8.5 SUMMARY

This chapter details the end-to-end pairwise key establishment scheme for secure message exchange in semantic view processing. The scheme is based on key pre-distribution and improves the level of security of pairwise keys by using multiple secure paths during establishment. The symmetric keys can then be used to secure all data communications between sensors against packet eavesdropping and traffic analysis by attackers in semantic view processing.

9.0 CONCLUSION AND FUTURE WORK

Data intensive applications, in particular disaster management and emergency response have very stringent requirement of efficient information delivery for data processing. The effectiveness of sensor networks in providing information is determined by human's capacity to recognize and comprehend information from the raw data collected, and act accordingly. Sensor networks should not only collect data from the physical world, but also facilitate users to extract and absorb information specific to their needs. Transmitting users irrelevant data during data processing not only overloads users with unneeded data but also incurs unnecessary communication overhead. This efficient information delivery requirement creates additional challenges for data processing in wireless sensor networks when they are deployed to collect data for these applications, because the user interests can be diversified and yet correlated in these applications. To bridge the gap between data collected by sensors and the information interests of users, the concept of "semantic view" is proposed in this thesis. The semantic view is a powerful abstraction, which allows the fusion of multi-sensor and multi-source data into a virtual data gathering and analysis infrastructure commensurate with the interest of the underlying organization.

In order to support semantic views in sensor networks, the relevant data of semantic views must be identified, selected and collected from sensors. To this end, a framework for enabling semantic views is proposed. The main components of the framework are sensing scheduling, query dissemination, query processing, location discovery and pairwise key establishment. "Query Aware Sensing" addresses how to schedule sensors to achieve a desired level of sensing coverage for a dynamic changing set of queries in sensor networks using a minimum amount of energy. "Probabilistic Query Dissemination" deals with how to reduce the energy consumption of delivering queries to their relevant sensor nodes in the

network. “Correlated Multi-query Processing” aims to reduce transmission cost of data collection for multiple queries by reusing shared data among them. “Location Discovery using Out-of-Range information with multi-lateration” is presented to provide locations, which is absolutely necessary in order to provide coverage and geographical filtering of data in sensor networks to sensor nodes. “End-to-End pairwise key establishment” is used to establish symmetric keys between sensors using multiple secure paths for securing message exchanges in semantic view processing against traffic analysis and compromised sensor nodes. The ultimate goal is to develop a general framework towards enabling energy efficient semantic views in sensor networks for data intensive applications.

In query aware sensing, the set of active sensors is dynamically adjusted to achieve the required level of coverage for the current set of semantic views. When new queries arrive at the base station, the base station first derives the level of coverage requirement, COV, from these queries. Then based on the current sensing scheduling, it uses GRASS, a greedy algorithm for sensing scheduling, to compute a minimum set of sensors which must be additionally activated to sense in order to provide the desired level of coverage. This set of sensors is added to the current set of active sensors. After one sensing period ends, the current set of active sensors is updated from the set of queries which still need to be processed in the next sensing period. Sensors do not sample data and their sensing boards are turned off unless they are instructed by the base station to sense, thereby further reducing energy consumption.

The queries in semantic views are also used at the same time by correlated multi-query processing at the base station. An estimation model is used to measure the size of the shared data between two queries. Based on the estimation value, pairs of queries are selected in such a way that the estimated size of shared data among all these pairs is maximal. A shared intermediate view is constructed for each pair of queries, which captures the actual set of sensor data shared by these two queries. To ensure semantic correctness, the original queries are rewritten into a different set of queries such that the data for the original query is now divided into two disjoint sets of sensors of the shared intermediate view and the rewritten query. The set of shared intermediate views is also dynamically updated when new queries arrive at the base station. Similar to query aware sensing, the set of shared

intermediate views is cleared at the end of a sensing period and rebuilt at the beginning of the next sensing period. The goal of correlated multi-query processing at the base station is to eliminate the overlapping of sensor data among queries so that each sensor only need to transmit/aggregate its data once.

These shared intermediate views, along with rewritten queries, are delivered to sensor nodes in the network using probabilistic query dissemination. In probabilistic query dissemination, each sensor forwards a query with certain probability. This probability is adapted to each sensor node's local information, such as the additional area its forwarding can cover, the additional number of sensor nodes its transmission can reach, or the number of messages with the same query it has already overheard. By adapting the forwarding probability to local topology information, these schemes can further reduce the number of messages needed to disseminate semantic views in sensor networks, in comparison to other gossip based broadcast schemes.

After sensors receive the queries, they use correlated data collection to reduce the number of data transmissions for correlated queries. In correlated data collection, each sensor node stores its data to a proxy sensor node which is closer to the base station. A proxy node is established when the data at the sensor node is first acquired by a query. The node in the routing tree which first aggregates the value of a sensor node becomes the proxy sensor for the sensor node. After a proxy sensor is established by a sensor node, any later query requesting its data shall retrieve the data from its proxy node. If the proxy node of a sensor fails, a new proxy node is established when the next query requests data from the sensor. Correlated data collection at sensor nodes is used to further reduce the number of data transmissions in sensor nodes, since the correlated query processing at the base station does not completely eliminate the redundancy of data communications in the network for multiple query processing.

Locations of sensor nodes must be known in the proposed framework for several reasons. First, sensors' locations must be known in order to compute the coverage level of a deployment field. With this information, the desired level of coverage can be ensured when sensors are turned off to preserve energy consumption of the sensor nodes. The second reason is that when the queries in user semantic views specify geographical constraints, e.g. the data from

a particular area is needed, a sensor needs to know its location in order to determine if its data is required for those queries. To this end, a location discovery scheme using out-of-range information with multi-lateration is proposed. The out-of-range information is based on the observation that if two sensors cannot hear from each other, then the distance between them must be larger than the transmission ranges of both sensors. This information can be easily obtained by maintaining a neighbor list at each sensor node. Any non-neighboring sensor of a sensor node can be inferred as out of its range. The out-of-range information, when combined with multi-lateration scheme, can be very useful to resolve location ambiguities of unknown sensors. The conditions that this out-of-range information can be used to resolve location ambiguities are developed for reference nodes and unknown nodes in different scenarios. An unknown sensor with location ambiguity then reaches these out-of-range nodes through multi-hop paths and the out-of-range nodes determine if they can help to resolve its location by checking the conditions which may apply. It is shown that, with out-of-range information, fewer reference nodes are needed to locate sensors in the network, which in turn reduces cost and energy consumption of the whole network since reference nodes are usually much more expensive and consumes more energy than normal sensors.

Furthermore, since semantic view processing relies on information exchange among sensors for collecting data from sensors, attackers can also gain these data by capturing and analyzing all the messages exchanged among sensors. To secure message exchanges for semantic view processing, an end-to-end pairwise key establishment scheme based on key pre-distribution is presented. This scheme allows any two sensors to set up a common symmetric key after key pre-distribution and path key establishment. These keys are then used to protect data communication links between sensors against packet eavesdropping and traffic analysis by attackers. The scheme also protects data communications among normal sensors from being exposed to compromised sensors.

In essence, semantic views support mission-aware information delivery commensurate with and relevant to the goals and needs of associated organizations in sensor networks. These protocols and algorithms towards enabling semantic views allow sensors to self-organize to identify, collect, and deliver the information specific to each decision maker efficiently. Since the effectiveness of sensor networks as a decision making tool is essentially limited by

human being's capacity to absorb and react on the information provided by these networks, "semantic views" can increase decision makers' capacity to recognize critical events such as disasters by allowing them to focus on data through "semantic views" that are directly relevant to their responsibilities. The "semantic view" reduces the overload of less relevant information and time required for information processing and facilitates rapid absorption of critical information by decision makers. Consequently, more prompt and accurate reaction can be taken on the critical events recognized by decision makers. This increased ability to recognize and react to events may significantly benefit human beings in various scenarios. For instance, in disaster management, an early perception of risk conditions such as tsunamis using sensor networks can provide us precious time to evacuate and may save a significant number of human lives when tsunami strikes. Although, the ultimate success of disaster management depends on how communities react to risk conditions, "semantic views" extends the capability of sensor networks towards such success in a social system from the technology's perspective.

9.1 FUTURE WORK

Currently, the framework and the designed algorithms and protocols have been evaluated through simulations. However, implementation of these algorithms and protocols in new research projects being funded and conducted to develop sensor networks for disaster management and emergency response, for instance, in an underwater sensor network being designed and implemented for near shore tsunami detection in city of Padang, Indonesia [1], would allow for further evaluation through actual experimentations.

The gap between the processing results of the queries in user semantic views and information requests from users must also be further examined. Even though semantic technologies such as metadata and ontology enables bridging and interconnection of data, content, and processes, and facilitates information integration from different sources of data, information discovery from a large amount of data and fast information retrieval after relevant data is generated [142][143][144][145][146], selecting the one most appropriate for sensor networks is

not an easy task but worth a deep study and analysis, because the tradeoffs among all these approaches must also be carefully considered in order to pick the best method for sensor networks.

APPENDIX

ALGORITHMS AND THEOREMS

A.1 MINIMUM SET K COVERING

In this section, we analyze the greedy algorithm used in query aware sensing scheduling for semantic view processing. For simplicity of analysis, the GRASS algorithm is transformed into an equivalent greedy algorithm for minimum set k covering problem at first, and an approximation factor is then developed for the greedy algorithm.

Theorem 3. *The greedy algorithm has a H_{mk} approximation factor for the minimum set k covering problem, where $H_{mk} = \sum_{1 \leq i \leq mk} \frac{1}{mk}$*

Proof. The proposed algorithm is equivalent to the following greedy algorithm for set k covering problem.

Let $x_1, x_2, \cdots, x_{mk}$ be the order of elements in U being selected by the greedy algorithm. When x_l is selected, there are at least $mk - l + 1$ elements left uncovered. The price of x_l, $price(x_l)$ must be less or equal to $\frac{OPT}{|U|}$, where OPT is the optimal number of sets selected to cover U for k times. Otherwise, the optimal solution will incur cost $price(x) \times |U| > OPT$. Therefore,

$$price(x_l) \leq \frac{OPT}{mk - l + 1} \tag{.1}$$

Algorithm 12 Greedy algorithm for set k covering

1: **for all** $a_i \in U$ **do**
2: $d(a_i) = k$
3: **end for**
4: $C = \emptyset$
5: **while** $U \neq \emptyset$ **do**
6: Find set $F \in S - C$, that minimizes $\alpha = \frac{1}{F \cap U}$
7: **for all** $a \in F \cap U$ **do**
8: $price(a) = \alpha$
9: **end for**
10: $C = C \cup F$
11: **for all** $a \in F \cap U$ **do**
12: $d(a) = d(a) - 1$
13: **if** $d(a) == 0$ **then**
14: $U = U - F$
15: **end if**
16: **end for**
17: **end while**

Due to the nature of the weight definition, the cost of the greedy algorithm $Cost$ is:

$$Cost = \sum_{1 \leq i \leq mk} price(x_i) \tag{.2}$$

From inequality .1, it is easy to derive the $Cost$ as:

$$\begin{aligned} Cost &\leq OPT \times \sum_{1 \leq i \leq mk} \frac{1}{mk - i + 1} \\ &= OPT \times H_{mk} \end{aligned} \tag{.3}$$

□

A.2 GABOW'S ALGORITHM FOR MAXIMUM MATCHING ON GRAPHS

The main sketch of Gabow's algorithm is described in this section, more details and discussions can be found at [131]. The algorithm consists of three routines, E, L and R. E is the main routine and it uses subroutines L and R.

L assigns the edge label $n(xy)$ to nonouter vertices edge xy, which joins outer vertices x, y. The main steps of L are described in Algorithm 14:

$R(v, w)$ rematches edges in the augmenting path. The main steps of R are described in Algorithm 15.

A.3 SECURITY ANALYSIS

In this section, we show how to select from two sets of node disjoint secure paths in our end-to-end pairwise key establishment for a higher level of security based on the number of hops in these paths.

Algorithm 13 E

1: **(Initialize)** Read the graph into adjacency list, numbering the vertices 1 to V, and the edges $V+1$ to $V+2W$. Create a dummy vertex 0. For $0 \leq i \leq V$, set $LABEL(i) = 1, MATE(i) = 0, u = 0$

2: **(Find unmatched vertex)** Set $u = u + 1$. If $u > V$, return; Otherwise, if vertex u is matched, repeat step 1, otherwise set $LABEL(u) = FIRST(u) = 0$

3: **(Choose an edge)** Choose an edge xy, where x is an outer vertex. If no such edge exists, go to step 8

4: **(Augment the matching)** If y is unmatched and $y \neq u$, set $MATE(y) = x$, call $R(x, y)$, then go to step 8

5: **(Assign edge labels)** If y is outer, call L, then go to step 3

6: **(Assign a vertex label)** Set $v = MATE(y)$. If v is nonouter, set $LABEL(v) = x, FIRST(v) = y$, and go to step 3

7: **(Get next edge)** Go to step 3

8: **(Stop the search)** Set $LABEL(0) = -1$. For all outer vertices i, set $LABEL(i) = LABEL(MATE(i)) = -1$, then go to step 1

Algorithm 14 L

1: **(Initialize)** set $r = FIRST(x), s = FIRST(y)$. If $r == s$, return. Otherwise flag r and s

2: **(Switch paths)** If $s \neq 0$, interchange r and s

3: **(Next nonouter vertex)** Set $r = FIRST(LABEL(MATE(r)))$. If r is not flagged, flag r and go to step 2, otherwise, set $join = r$ and go to step 4

4: **(Label vertices in $P(x), P(y)$)** Set $v = FIRST(x)$, do step 5. Set $v = FIRST(y)$, do step 5. Then go to step 6

5: **(Label v)** if $v \neq join$, set $LABEL(v) = n(xy), FIRST(v) = join, v = FIRST(LABEL(MATE(v)))$, repeat current step. Otherwise, continue as specified in step 4

6: **(Update $FIRST$)** For each outer vertex i, if $FIRST(i)$ is outer, set $FIRST(i) = join$

7: **(Done)** Return

Algorithm 15 R

1: **(Match v to w)** Set $t = MATE(v), MATE(v) = w$. If $MATE(t) \neq v$, return

2: **(Rematch a path)** If v has a vertex label, set $MATE(t) = LABEL(v)$, call $R(LABEL(v), t)$ recursively and then return

3: **(Rematch two paths)** Set x, y to vertices so $LABEL(v) = n(xy)$, call $R(x, y)$ recursively, call $R(y, x)$ recursively, and then return

A.3.1 Proof of Lemma 2

Proof. We reform Equation 8.3 into another form in order to prove Lemma 2.

$$\begin{aligned} prob[\{l_1 \dots l_s\}, x, n] &= \frac{\sum_{j_1=1}^{l_1} \cdots \sum_{j_s=1}^{l_s} \binom{n-\sum_{m=1}^{s} l_m - s + \sum_{i=1}^{s} j_i}{x-s}}{\binom{n}{x}} \\ &= \frac{\sum_{j=s}^{\sum_{i=1}^{s} l_i} e_j \times \binom{j+n-s-\sum_{m=1}^{s} l_m}{x-s}}{\binom{n}{x}} \end{aligned}$$

e_j is the number of s-tuples whose summation of all elements is j.

$$\begin{aligned} e_j &= e([l_1, l_2 \dots l_s], j) \\ &= \mid E([l_1, l_2 \dots l_s], j) \mid \\ &= \mid \{(i_1 \dots i_s) \mid \sum_{m=1}^{s} i_m = j \\ &\quad \text{and } i_m \leq l_m \forall 1 \leq m \leq s\} \mid \end{aligned}$$

Before we prove Lemma 2, let us look at three properties of e_j.

Property 1. $e([l_1 \dots l_{s-1}, l_s], j) = e([l_1 \dots l_{s-1}, l_s - 1], j) + e([l_1 \dots l_{s-1}],$
$j - l_s)$

For all s-tuples in $\{(i_1 \dots i_s) \mid \sum_{m=1}^{s} i_m = j \; and \; i_m \leq l_m \forall 1 \leq m \leq s\}$, the s_{th} index i_s is either l_s or less than l_s. If $i_s = l_s$ then the summation of all other elements should be $j - l_s$. There is totally $e([l_1 \dots l_{s-1}], j - l_s)$ such s-tuples. When $i_s < l_s$, the number of tuples equal to $e([l_1 \dots l_{s-1}, l_s - 1], j)$. Thus $e([l_1 \dots l_{s-1}, l_s], j) = e([l_1 \dots l_{s-1}, l_s - 1], j) + e([l_1 \dots l_{s-1}], j - l_s)$

Property 2. $e([l_1, l_2 \dots l_s], j) = e([l_2 \dots l_s], j-1) + e([l_2 \dots l_s], j-2) + \cdots + e([l_2 \dots l_s], j - l_1)$

Property 2 can be proved by applying Property 1 on l_1.

$$\begin{aligned} e([l_1, l_2 \dots l_s], j) &= e([l_1 - 1, l_2 \dots l_s], j) + e([l_2 \dots l_s], j - l_1) \\ &= e([l_1 - 2, l_2 \dots l_s], j) + e([l_2 \dots l_s], j - (l_1 - 1)) + \\ &\quad e([l_2 \dots l_s], j - l_1) \\ &= e([0, l_2 \dots l_s], j) + e([l_2 \dots l_s], j - 1) + \\ &\quad e([l_2 \dots l_s], j - 2) + \dots + e([l_2 \dots l_s], j - l_1) \\ &= e([l_2 \dots l_s], j - 1) + e([l_2 \dots l_s], j - 2) + \dots + \\ &\quad e([l_2 \dots l_s], j - l_1) \end{aligned}$$

Property 3. $e([l_1 \dots l_m], j) \leq e([l_1 \dots l_m + y], j + y)$

For any s-tuple $(i_1 \dots i_{m-1}, i_m)$ in $E([l_1 \dots l_m], j)$, we can construct another s-tuple $(i_1^{'} \dots, i_{m-1}^{'}, i_m^{'})$ in $E([l_1 \dots l_m + y], j + y)$ as following: $i_m^{'} = i_m + y$ and $i_t^{'} = i_t \; \forall \; 1 \leq t \leq m - 1$. $\sum_{t=1}^{m} i_t^{'} = \sum_{t=1}^{m} i_t \; + y = j + y$ and $i_t^{'} = i_t \leq l_t \; \forall \; 1 \leq t \leq m - 1; i_m^{'} = i_m + y \leq l_m + y$. Thus, this new tuple belongs to $E([l_1 \dots l_m + y], j + y)$.

If $\sum_{j=1}^{s} l_j = s \times l$, then

$$\begin{aligned} prob[\{l_1 \dots l_s\}, x, n] &= \frac{\sum_{j=s}^{l \times s} e([l_1 \dots l_s], j) \times \binom{j+n-s-s \times l}{x-s}}{\binom{n}{x}} \\ prob[\{l \dots l\}, x, n] &= \frac{\sum_{j=s}^{l \times s} e([l \dots l], j) \times \binom{j+n-s-s \times l}{x-s}}{\binom{n}{x}} \end{aligned}$$

So we can prove Lemma 2 by proving $e([l_1 \dots l_s], j) \leq e([l \dots l], j) \; \forall \; s \leq j \leq s \times l$ using strong induction. The following describes the main steps in the induction.

Step 1: $s = 1$. Since $l_1 = l$, $e([l_1], j) = e([l], j) \leq e([l], j)$

Step 2: Assume $e([l_1 \dots l_m], j) \leq e([l \dots l]_m, j)\ \forall\ m \leq j \leq m \times l$ for all $m \leq s-1$.

$$
\begin{aligned}
e([l_1, l_2 \dots l_s], j) &= e([l_2 \dots l_s], j-1) + e([l_2 \dots l_s], j-2) + \cdots \\
&\quad + e([l_2 \dots l_s], j - l_1) \\
&= e([l_2 \dots l_s'], j-1) + e([l_2 \dots l_{s-1}], j-1-l_s) + \\
&\quad \cdots + e([l_2 \dots l_{s-1}], j-1-l_s + (l - l_1 - 1)) + \\
&\quad e([l_2 \dots l_s'], j-2) + e([l_2 \dots l_{s-1}], j-2-l_s) + \\
&\quad \cdots + e([l_2 \dots l_{s-1}], j-2-l_s + (l - l_1 - 1)) + \cdots + \\
&\quad e([l_2 \dots l_s'], j - l_1) + e([l_2 \dots l_{s-1}], j - l_1 - l_s) + \\
&\quad \cdots + e([l_2 \dots l_{s-1}], j - l_1 - l_s + (l - l_1 - 1)) \\
&= e([l_2 \dots l_s'], j-1) + e([l_2 \dots l_s'], j-2) + \\
&\quad \cdots + e([l_2 \dots l_s'], j - l_1) + \\
&\quad e([l_1, l_2 \dots l_{s-1}], j - l_s) + e([l_1, l_2 \dots l_{s-1}], j - l_s + 1) \\
&\quad + \cdots + e([l_1, l_2 \dots l_{s-1}], j - l_s + (l - l_1 - 1)) \\
&\leq e([l \dots l]_{s-1}, j-1) + e([l \dots l]_{s-1}, j-2) + \\
&\quad \cdots + e([l \dots l]_{s-1}, j - l_1) + \\
&\quad e([l_1, l_2 \dots l_{s-1}], j - l_s) + e([l_1, l_2 \dots l_{s-1}], j - l_s + 1) \\
&\quad + \cdots + e([l_1, l_2 \dots l_{s-1}], j - l_s + (l - l_1 - 1)) \\
&\leq e([l \dots l]_{s-1}, j-1) + e([l \dots l]_{s-1}, j-2) + \\
&\quad \cdots + e([l \dots l]_{s-1}, j - l_1) + \\
&\quad e([l_1, l_2 \dots l_{s-1} - l + l_s], j - l) + \\
&\quad e([l_1, l_2 \dots l_{s-1} - l + l_s], j - l + 1) + \\
&\quad \cdots + e([l_1, l_2 \dots l_{s-1} - l + l_s], j - l_1 - 1) \\
&\leq e([l \dots l]_{s-1}, j-1) + e([l \dots l]_{s-1}, j-2) + \\
&\quad \cdots + e([l \dots l]_{s-1}, j - l_1) + \\
&\quad e([l \dots l]_{s-1}, j - l) + e([l \dots l]_{s-1}, j - l + 1) + \\
&\quad \cdots + e([l \dots l]_{s-1}, j - l_1 - 1) \\
&= e([l, l \dots l]_s, j).
\end{aligned}
$$

□

A.3.2 Proof of Lemma 3

Proof. Obviously, $prob[\{l_1 \dots l_s\}, x, n] \leq prob[\{l_1 \dots l_s + y\}, x, n]$ if $y \geq 0$. So if $\sum_{j=1}^{s} l_j \leq s \times l$, then

$$\begin{aligned} prob[\{l_1 \dots l_s\}, x, n] &\leq prob[\{l_1 \dots l_s + s \times l - \sum_{j=1}^{s} l_j\}, x, n] \\ &\leq prob[\{\frac{l_1 + \dots + l_s + s \times l - \sum_{j=1}^{s} l_j}{s} \dots \\ &\quad \frac{l_1 + \dots + l_s + s \times l - \sum_{j=1}^{s} l_j}{s}\}_s, x, n] \\ &= prob[\{l \dots l\}_s, x, n] \end{aligned}$$

□

BIBLIOGRAPHY

[1] L. Comfort, R. Melhem, and D. Mosse, "DRU: Designing Resilience for Communities at Risk: Decision Support for Collective Action under Stress." [Online]. Available: http://www.iisis.pitt.edu/projects/resilience.shtml

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 88–97.

[3] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi, "Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet," in *In MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004, pp. 256–269.

[4] CMU, "Center for Sensed Critical Infrastructure Research." [Online]. Available: http://www.ices.cmu.edu/censcir

[5] A. Hasler, I. Talzi, J. Beutel, C. Tschudin, and S. Gruber, "Wireless sensor networks in permafrost research c concept, requirements, implementation and challenges," in *Proceedings of the 9th International Conference on Permafrost 2008*, 2008, pp. 669–674.

[6] W. M. Cohen and D. A. Levinthal, "Absorptive capacity: A new perspective on learning and innovation," *Administrative Science Quarterly*, vol. 35, pp. 128–152, 1990.

[7] L. Comfort, D. Mosse, and T. Znati, "Managing risk in real time: Integrating information technology into disaster risk reduction and response," *Commonwealth: A Journal of Political Science Policy Issue on Emergency Management*, vol. 15-4, May 2009.

[8] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-hoc Sensor Networks," *SIGOPS Operating System Review*, vol. 36, no. SI, pp. 131–146, 2002.

[9] U. Srivastava, K. Munagala, and J. Widom, "Operator Placement for in-network Stream Query Processing," in *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2005, pp. 250–258.

[10] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, June 2002.

[11] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 95–107.

[12] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, L.A. CA USA, November 2003.

[13] S. Du, A. K. Saha, and D. B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, May 2007.

[14] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks," in *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2008, pp. 53–62.

[15] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, June 1999.

[16] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: a media access protocol for wireless lan's," in *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*. New York, NY, USA: ACM, 1994, pp. 212–225.

[17] S. Singh and C. S. Raghavendra, "Pamas—power aware multi-access protocol with signalling for ad hoc networks," *SIGCOMM Computer Communications Review*, vol. 28, no. 3, pp. 5–26, 1998.

[18] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1997, pp. 90–100.

[19] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Kluwer Academic Publishers, 1996, pp. 153–181.

[20] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Computer Communications Review*, vol. 24, no. 4, pp. 234–244, 1994.

[21] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of IEEE Conference on Computer Commu-*

nications (INFOCOM). Washington, DC, USA: IEEE Computer Society, 1997, p. 1405.

[22] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.

[23] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of 6th Annual International Conference on Mobile Computing and Networking (Mobicom)*, Boston, MA, USA, 2000, pp. 56–67.

[24] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 169–185, 2002.

[25] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, 2002, pp. 1125–1130.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 1–10.

[27] D. Braginsky and D. Estrin, "Rumor Routing Algorthim for Sensor Networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 22–31.

[28] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in *Proceedings of the 9th ACM conference on Computer and Communication Security (CCS)*, November 2002, pp. 41–47.

[29] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security and Privacy*, May 2003, pp. 197–213.

[30] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communication Security(CCS)*, October 2003, pp. 42–51.

[31] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2004.

[32] R. D. Pietro, L. V. Mancini, and A. Mei, "Random key assignment for secure wireless sensor networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks*, October 2003.

[33] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach," in *Proceedings of the 11th IEEE International Conference on Networking Protocols (ICNP)*, Atlanta, GA, November 2003.

[34] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, Washington, DC, October 2003.

[35] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing Sensor Networks with Location-Based Keys," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2005*, New Orleans, LA, 2005, pp. 1909–1914.

[36] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (SASN'03)*, Fairfax, Virginia, 2003, pp. 72–82.

[37] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Miami, FL, March 2005.

[38] D. Liu, P. Ning, and W. Du, "Group-Based key Pre-Distribution in Wireless Sensor Networks," in *Proceedings of ACM Workshop on Wireless Security (Wise'05)*, Cologne, Germany, 2005.

[39] H. Ling and T. Znati, "End-to-End Pairwise Key Establishment using Multi-path in Wireless Sensor Network," in *Proceedings of the 2005 IEEE Global Communications Conference(GLOBECOM 2005)*, St. Louis, MO, November 2005.

[40] G. Li, H. Ling, and T. Znati, "Path Key Establishment using Multiple Secured Paths in Wireless Sensor Networks," in *Proceedings of the ACM International Conference on emerging Networking EXperiments and Technologies (CoNext) 2005*, Toulouse, France, October 2005.

[41] H. Ling and T. Znati, "End-to-end pairwise key establishment using node disjoint secure paths in wireless sensor networks," *International Journal of Security and Networks (IJSN)*, vol. 2, no. 1/2, pp. 109–121, 2007.

[42] G. Li, H. Ling, T. Znati, and W. Wu, "A robust on-demand path-key establishment framework via random key predistribution for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking (WCN)*, vol. 2006, no. 2, pp. 80–80, 2006.

[43] H. Ling and T. Znati, "Establishing pairwise keys in wireless sensor networks using multiple paths," *Ad Hoc & Sensor Wireless Networks*, vol. 4, no. 1-2, pp. 43–68, 2007.

[44] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 1999.

[45] D. M. Balenson, D. McGrew, and A. Sherman, "Key Management for Large Dynamic Groups: One-way Function Trees and Amortized Initialization," in *IETF Internet draft(work in progress)*, August 2000.

[46] S. Setia, S. Koussih, and S. Jajodia, "Kronos: A Scalable Group Re-Keying Approach for Secure Multicast," in *IEEE Symposium on Security and Privacy*, Oakland,CA, May 2000.

[47] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam, "Reliable Group Rekeying: a Performance Analysis," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 2001, pp. 27–38.

[48] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," in *Lecture Notes in Computer Science*, vol. 2139, 2001, pp. 41–62.

[49] B. Briscoe, "MARKS: Zero Side-effect Multicast Key Management using Arbitrarily Revealed Key Sequences," in *First International Workshop on Networked Group Communication*, 1999.

[50] A. Perrig, D. Song, and J. D. Tygar, "ELK, a New Protocol for Efficient Large-group Key Distribution," in *IEEE Symposium on Security and Privacy*, Oakland,CA, May 2001.

[51] L. Lazos and R. Poovendran, "Energy-Aware Secure Multicast Communication in Ad-hoc Networks Using Geographic Location Information," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hongkong, China, April 2003.

[52] D. Liu, P. Ning, and K. Sun, "Efficient Self-Healing Group Key Distribution with Revocation Capability," in *Proceedings of the 10th ACM Conference on Computer and Communication Security(CCS)*, October 2003.

[53] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean, "Self-Healing Key Distribution with Revocation," in *IEEE Symposium on Security and Privacy*, Oakland,CA, May 2002.

[54] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks," in *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems*, 2004, pp. 42–51.

[55] H. Ling and T. Znati, "Gkm: A group dynamics aware key management scheme for multicast communications in ad-hoc sensor networks," in *the 26th IEEE International*

Performance Computing and Communications Conference (IPCCC), New Orleans, Louisiana, USA, April 2007, pp. 459–466.

[56] Y. Yao and J. Gehrke, "The Cougar Approach to in-network Query Processing in Sensor Networks," *SIGMOD Record*, vol. 31, no. 3, pp. 9–18, 2002.

[57] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an Acquisitional Query Processing System for Sensor Networks," *ACM Transaction Database System*, vol. 30, no. 1, pp. 122–173, 2005.

[58] A. Silberstein, R. Braynard, and J. Yang, "Constraint Chaining: on Energy-efficient Continuous Monitoring in Sensor Networks," in *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 2006, pp. 157–168.

[59] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven Data Acquisition in Sensor Networks," in *In Proceedings of Conference on Very Large Data Bases (VLDB)*, August 2004.

[60] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems.* New York, NY, USA: ACM, 2006, pp. 427–428.

[61] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services.* New York, NY, USA: ACM, 2004, pp. 270–283.

[62] S. Coleri, S. Y. Cheung, and P. Varaiya, "Sensor networks for monitoring traffic," in *Forty-Second Annual Allerton Conference on Communication, Control, and Computing*, September 2004.

[63] E. Cayirci and T. Coplu, "Sendrom: sensor networks for disaster relief operations management," *Wireless Networks*, vol. 13, no. 3, pp. 409–423, 2007.

[64] L. M. Ni, Y. Zhu, J. Ma, Q. Luo, Y. Liu, S. C. Cheung, Q. Yang, M. Li, and M. Wu, "Semantic sensor net: an extensible framework," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 4, no. 3/4, pp. 157–167, 2009.

[65] M. Imai, Y. Hirota, S. Satake, and H. Kawashima, "Semantic sensor network for physically grounded applications," in *9th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2006, pp. 1–6.

[66] A. Sheth, C. Henson, and S. S. Sahoo, "Semantic sensor web," *IEEE Internet Computing*, vol. 12, no. 4, pp. 78–83, 2008.

[67] C. Goodwin and D. Russomanno, "An ontology-based sensor network prototype environment," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, 2006, pp. 1–2.

[68] A. Wun, M. Petrovi, and H.-A. Jacobsen, "A system for semantic data fusion in sensor networks," in *DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems.* New York, NY, USA: ACM, 2007, pp. 75–79.

[69] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: Challenges and opportunities," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16–23, 2004.

[70] A. Rajasekar, S. Lu, R. Moore, F. Vernon, J. Orcutt, and K. Lindquist, "Accessing sensor data using meta data: a virtual object ring buffer framework," in *DMSN '05: Proceedings of the 2nd international workshop on Data management for sensor networks.* New York, NY, USA: ACM, 2005, pp. 35–42.

[71] S. Avancha, C. Patel, and A. Joshi, "Ontology-driven Adaptive Sensor Networks," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, August 2004, pp. 194–202.

[72] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, Florence, Italy, May 2006.

[73] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, USC/Information Sciences Institute. Rome, Italy: ACM, July 2001, pp. 70–84.

[74] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Providence, RI, USA, May 2003, pp. 28–37.

[75] D. Tian and N. D. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA).* New York, NY, USA: ACM Press, 2002, pp. 32–41.

[76] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," in *Proceedings of ACM SenSys 2003*, L.A. CA USA, November 2003.

[77] H. Zhang and J. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc and Sensor Wireless Networks, an international journal*, vol. 1, pp. 89–123, January 2005.

[78] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[79] M. Hefeeda and H. Ahmadi, "A Probabilistic Coverage Protocol for Wireless Sensor Networks," in *Proceedings of the 15th IEEE International Conference on Networking Protocols*, Beijing, China, October 2007.

[80] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.

[81] S. Yang, F. Dai, M. Cardei, J. Wu, and F. Patterson, "On connected multiple point coverage in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 13, no. 4, pp. 289–301, October 2006.

[82] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2003, pp. 1609–1614.

[83] J. Wang and N. Zhong, "Efficient point coverage in wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 291–304, May 2006.

[84] J. Lu, L. Bao, and T. Suda, "Coverage-aware sensor engagement in dense sensor networks," *Journal of Embedded Computing*, vol. 3, no. 1, pp. 3–18, 2009.

[85] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 413 – 420, 2006.

[86] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 151–162.

[87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the 6th annual ACM Symposium on Principles of Distributed Computing*, 1987, pp. 1–12.

[88] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, June 2002, pp. 1707–1716.

[89] L. Rodrigues, S. Handurukande, J. Pereira, R. Guerraoui, and A.-M. Kermarrec, "Adaptive Gossip-Based Broadcast," in *DSN '03: Proceedings of the 2003 Interna-*

tional Conference on Dependable Systems and Networks, Washington, DC, USA, 2003, pp. 47–56.

[90] H. Ling, D. Mosse, and T. Znati, "Coverage-based Probabilistic Forwarding for Ad Hoc Routing," in *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN 2005)*, October 2005, pp. 13–18.

[91] Q. Sun and D. C. Sturman, "A Gossip-Based Reliable Multicast for Large-Scale High-Throughput Applications," in *DSN '00: Proceedings of the 2000 International Conference on Dependable Systems and Networks*, Washington, DC, USA, 2000, pp. 347–358.

[92] L. Alvisi, J. Doumen, R. Guerraoui, B. Koldehofe, H. Li, R. van Renesse, and G. Tredan, "How Robust are Gossip-based Communication Protocols?" *SIGOPS Operating Systems Review*, vol. 41, no. 5, pp. 14–18, 2007.

[93] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith, "Parametric Probabilistic Sensor Network Routing," in *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 122–131.

[94] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized Techniques for Broadcasting in Wireless Sensor Networks," in *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*, Philadelphia, PA, USA, October 2004.

[95] A. V. Kini, V. Veeraraghavan, N. Singhal, and S. Weber, "SmartGossip: An Improved Randomized Broadcast Protocol for Sensor Networks," in *Proceedings of Fifth International Conference on Information Processing in Sensor Networks (IPSN 2006)*, April 2006, pp. 210–217.

[96] P. Roy, S. Seshadri, S. Sudarshan, and S. Bhobe, "Efficient and Extensible Algorithms for Multi Query Optimization," *SIGMOD Record*, vol. 29, no. 2, pp. 249–260, 2000.

[97] N. Trigoni, Y. Yao, A. J. Demers, J. Gehrke, and R. Rajaraman, "Multi-query Optimization for Sensor Networks," in *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2005, pp. 307–321.

[98] S. Xiang, H. B. Lim, and K.-L. Tan, "Impact of Multi-query Optimization in Sensor Networks," in *DMSN '06: Proceedings of the 3rd workshop on Data management for sensor networks*, 2006, pp. 7–12.

[99] S. Xiang, H. B. Lim, K.-L. Tan, and Y. Zhou, "Two-Tier Multiple Query Optimization for Sensor Networks," in *ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems*, 2007, pp. 39–47.

[100] A. Silberstein and J. Yang, "Many-to-Many Aggregation for Sensor Networks," in *Proceedings of the 2007 IEEE International Conference on Data Engineering (ICDE)*, 2007, pp. 986–995.

[101] S. Krishnamurthy, C. Wu, and M. J. Franklin, "On-the-Fly Sharing for Streamed Aggregation," in *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 2006.

[102] J. Gibson, Ed., *The Mobile Communications Handbook*. IEEE Press, 1999.

[103] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, 4th ed. Springer-Verlag, 1997.

[104] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-support System," in *Proceedings of the 6th annual international conference on Mobile computing and networking (MOBICOM)*, 2000, pp. 32–43.

[105] N. Malhotra, M. Krasniewski, C. Yang, S. Bagchi, and W. Chappell, "Location Estimation in Ad-Hoc Networks with Directional Antennas," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Columbus, Ohio, USA, June 6-10 2005.

[106] A. Savvides, C. chieh Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM01)*, Rome, Italy, July 2001.

[107] A. Savvides, H. Park, and M. B. Srivastava, "The Bits and Flops of the n-hop Multilateration Primitive for Node Localization Problems," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA)*, 2002, pp. 112–121.

[108] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2001.

[109] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, April 2003.

[110] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Fort Lauderdale, FL, October 2004.

[111] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Miami, FL, March 2005.

[112] Z. Guo, Y. Guo, F. Hong, X. Yang, Y. He, Y. Feng, and Y. Liu, "Perpendicular intersection: Locating wireless sensors with mobile beacon," *IEEE International Real-Time Systems Symposium*, vol. 0, pp. 93–102, 2008.

[113] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, 2003.

[114] S. Y. Wong, J. Lim, S. V. Rao, and W. K. Seah, "Multihop Localization with Density and Path Length Awareness in Non-Uniform Wireless Sensor Networks," in *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC2005-Spring)*, Stockhom, Sweden, May-June 2005.

[115] V. Chandrasekhar and W. K.G. Seah, "Area Localization Scheme for Underwater Sensor Networks," in *Proceedings of the IEEE OCEANS Asia Pacific Conference*, Singapore, May 2006.

[116] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes for Large Scale Sensor Networks," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 81–95.

[117] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha, "Resilient Localization for Sensor Networks in Outdoor Environments," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Columbus, Ohio, USA, June 6-10 2005, pp. 643–652.

[118] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, Baltimore, MD, November 2004, pp. 50–61.

[119] Y. Wei, Z. Yu, and Y. Guan, "COTA: A Robust Multi-hop Localization Scheme in Wireless Sensor Networks," in *Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS 2006)*, June 2006.

[120] L. Fang, W. Du, and P. Ning, "A beacon-less location discovery scheme for wireless sensor networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Miami, FL, USA, March 2005.

[121] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2004, pp. 2652 – 2661.

[122] D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. D. Anderson, "Network Localization in Partially Localizable Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Maimi, FL, March 2005.

[123] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur, "A Theory of Network Localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[124] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. Anderson, A. S. Morse, and Y. R. Yang, "Localization in Sparse Networks using Sweeps," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, September 2006.

[125] M. L. Sichitiu and C. Veerarittiphan, "Simple, Accurate Time Synchronization for Wireless Sensor Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2003, pp. 1266–1273.

[126] P. Berman, G. Galinescu, C. Shah, and A. Zelikovsky, "Power Efficient Monitoring Management in Sensor Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, Georgia, USA, March 2004, pp. 2329–2334.

[127] K. Rosen, *Discrete Mathematics and Its Applications*. WCB/McGraw-Hill, 1999.

[128] GNU, "GLPK (GNU Linear Programming Kit)." [Online]. Available: http://www.gnu.org/software/glpk/

[129] J. Edmonds, "Path, Trees and Flowers," *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965.

[130] G. N. Harold, "Implementation of algorithms for maximum matching on nonbipartite graphs," Ph.D. dissertation, Department of Computer Science, Standford University, Stanford, CA, USA, 1974.

[131] G. Harold, "An Efficient Implementation of Edmonds Algorithm for Maximum Matching on Graphs," *Journal of ACM*, vol. 23, pp. 221–234, 1976.

[132] G. N. Harold, "A scaling algorithm for weighted matching on general graphs," in *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1985, pp. 90–100.

[133] G. Zvi, "Efficient algorithms for finding maximum matching in graphs," *ACM Computing Surveys*, vol. 18, no. 1, pp. 23–38, 1986.

[134] C. Chekuri and A. Rajaraman, "Conjunctive query containment revisited," in *ICDT '97: Proceedings of the 6th International Conference on Database Theory*. London, UK: Springer-Verlag, 1997, pp. 56–70.

[135] A. Y. Halevy, "Answering queries using views: A survey," *The VLDB Journal*, vol. 10, no. 4, pp. 270–294, 2001.

[136] D. Calvanese, G. De Giacomo, and M. Lenzerini, "On the decidability of query containment under constraints," in *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* New York, NY, USA: ACM, 1998, pp. 149–158.

[137] G. Gottlob, N. Leone, and F. Scarcello, "The complexity of acyclic conjunctive queries," *Journal of ACM*, vol. 48, no. 3, pp. 431–498, 2001.

[138] P. G. Kolaitis, D. L. Martin, and M. N. Thakur, "On the complexity of the containment problem for conjunctive queries with built-in predicates," in *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* New York, NY, USA: ACM, 1998, pp. 197–204.

[139] S. Chaudhuri and M. Y. Vardi, "On the complexity of equivalence between recursive and nonrecursive datalog programs," in *PODS '94: Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* New York, NY, USA: ACM, 1994, pp. 107–116.

[140] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," in *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.

[141] X. Li and L. Cuthbert, "Node-disjointness-based multipath routing for mobile ad hoc networks," *Proceedings of the 1st ACM international workshop on PE-WASUN*, pp. 23–29, October 2004.

[142] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Humman-Computer Studies*, vol. 43, no. 5-6, pp. 907–928, 1995.

[143] D. L. McGuinness, R. Fikes, J. Hendler, and L. A. Stein, "Daml+oil: An ontology language for the semantic web," *IEEE Intelligent Systems*, vol. 17, no. 5, pp. 72–80, October 2002.

[144] C. Sangpachatanaruk, "An overlay architecture for personalized object access and sharing in a peer-to-peer environment," Ph.D. dissertation, School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA, 2006.

[145] Webont, "Web Ontology Language," 2001. [Online]. Available: http://www.w3.org/2001/sw/WebOnt/

[146] F. P. Bretherton and P. T. Singley, "Metadata: a user's view," in *SSDBM'1994: Proceedings of the 7th international conference on Scientific and Statistical Database Management.* Washington, DC, USA: IEEE Computer Society, 1994, pp. 166–174.