Fall 2014

Energy aware and privacy preserving protocols for ad hoc networks with applications to disaster management

Mayank Raj

Department: Computer Science

ENERGY AWARE AND PRIVACY PRESERVING PROTOCOLS FOR AD HOC NETWORKS WITH APPLICATIONS TO DISASTER MANAGEMENT

by

MAYANK RAJ

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2014

Approved by

Dr. Sajal K. Das, Advisor
Dr. Krishna Kant
Dr. Sanjay Madria
Dr. Sriram Chellappan
Dr. Bruce McMillin

ABSTRACT

Disasters can have a serious impact on the functioning of communities and societies. Disaster management aims at providing efficient utilization of resources during pre-disaster (e.g. preparedness and prevention) and post-disaster (e.g. recovery and relief) scenarios to reduce the impact of disasters. Wireless sensors have been extensively used for early detection and prevention of disasters. However, the sensor's operating environment may not always be congenial to these applications. Attackers can observe the traffic flow in the network to determine the location of the sensors and exploit it. For example, in intrusion detection systems, the information can be used to identify coverage gaps and avoid detection. Data source location privacy preservation protocols were designed in this work to address this problem.

Using wireless sensors for disaster preparedness, recovery and relief operations can have high deployment costs. Making use of wireless devices (e.g. smartphones and tablets) widely available among people in the affected region is a more practical approach. Disaster preparedness involves dissemination of information among the people to make them aware of the risks they will face in the event of a disaster and how to actively prepare for them. The content is downloaded by the people on their smartphones and tablets for ubiquitous access. As these devices are primarily constrained by their available energy, this work introduces an energy-aware peer-to-peer file sharing protocol for efficient distribution of the content and maximizing the lifetime of the devices. Finally, the ability of the wireless devices to build an ad hoc network for capturing and collecting data for disaster relief and recovery operations was investigated. Specifically, novel energy-adaptive mechanisms were designed for autonomous creation of the ad hoc network, distribution of data capturing task among the devices, and collection of data with minimum delay.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude towards my advisor Prof. Sajal K. Das and Prof. Krishna Kant from Temple University for their continuous support and guidance during my graduate studies. I have benefited greatly from their knowledge and advice, which was instrumental in the completion of my PhD.

I would like to thank the members of my dissertation committee, Prof. Sanjay Madria, Prof. Bruce McMillin, and Prof. Sriram Chellapan for their constructive comments and feedback. Special thanks goes to Prof. Matthew Wright, Prof. Donggang Liu, Prof. Gautam Das, Prof. Yonghie Liu, and Prof. Qilian Liang from University of Texas at Arlington for their constant guidance and support during my studies there. A special mention must go to Prof. G. N. S. Prasanna and Prof. Debabrata Das from International Institute of Information Technology, Bangalore who introduced me to research and motivated me to do PhD.

During my graduate studies, I was also fortunate to collaborate with many current and past members of the CReWMaN Lab. I would like to acknowledge the contributions of Prof. Mario Di Francesco and Prof. Na Li with whom I spent countless hours in the lab and engaged in fruitful discussions.

I owe a great debt to my father Mahesh Sharan Sinha, my mother Manju Lata, my younger brother Amrit Raj, and my dear wife Aditi Sinha, who have been a constant source of support and motivation. Finally, I would like to dedicate this work to my late grandfather Shri Suresh Nath and my late grandmother Shrimati Pramila Devi for their undying love and faith in me.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF ILLUSTRATIONS x

LIST OF TABLES xii

SECTION

1 INTRODUCTION 1

1.1 DISASTER MANAGEMENT 1

1.1.1 Disaster Prevention 2

1.1.2 Disaster Preparedness 2

1.1.3 Disaster Relief 2

1.1.4 Disaster Recovery 2

1.2 MOTIVATION AND CONTRIBUTIONS 3

1.3 ORGANIZATION 5

2 AN INTRODUCTION TO DISASTER MANAGEMENT: A TECHNOLOGICAL PERSPECTIVE 6

2.1 DISASTER PREVENTION 6

2.1.1 Use of Wireless Sensor Networks for Disaster Prevention 6

2.1.2 Challenges in Use of Wireless Sensor Networks for Disaster Prevention 7

2.1.2.1 Neighbor discovery 7

2.1.2.2 Data gathering 8

2.1.3 Security 9

2.2 DISASTER PREPAREDNESS 9

2.3 DISASTER RELIEF AND RECOVERY 10

3 A SOURCE LOCATION PRIVACY PRESERVING PROTOCOL FOR DISASTER PPREVENTION IN AD HOC NETWORKS . . 12

3.1 SYSTEM MODEL 14

3.1.1 Static Sensors 15

3.1.2 Data Mules 15

3.2 ATTACK MODEL 15

3.2.1 Linear Regression Based Traffic Analysis 16

3.2.2 Compromising Phantom Routing 17

3.3 THE α-ANGLE ANONYMITY 19

3.4 MULES-SAVING-SOURCE (MSS) PROTOCOL 20

3.5 ANALYSIS OF MSS PROTOCOL 23

3.5.1 Buffering Delay at Data Source 24

3.5.2 Carrying Delay At Data Mules 24

3.5.2.1 Expected carrying delay at data mules 26

3.5.2.2 Expected total delay in MSS and DD protocol 28

3.5.2.3 Lower bound of carrying delay at data mules . 28

3.6 EXPERIMENTAL STUDY OF THE MSS PROTOCOL . . . 29

3.6.1 Delay and Data Mules 30

3.6.2 Delay and Privacy Preservation 31

3.6.3 MSS and Direct Delivery 33

3.6.4 Discussion 33

3.7 MULE-SOURCE-SAVING - SHORTEST PATH (MSS-SP) PROTOCOL 34

3.7.1 Description of the MSS-SP Protocol 35

3.7.2 Analysis of MSS-SP Protocol 35

3.7.2.1 Buffering delay at the data source 35
3.7.2.2 Expected carrying delay at the data mules . . 35
3.7.3 Experimental Study of the MSS-SP Protocol 36
3.7.3.1 Delay and data mules 36
3.7.3.2 Delay and privacy preservation 36
3.7.3.3 Comparison of MSS and MSS-SP protocol . . 36
3.8 MULE-SOURCE-SAVING - TWO LEVEL (MSS-TL) PROTOCOL 39
3.8.1 Description of the MSS-TL Protocol 39
3.8.2 Analysis of MSS-TL Protocol 41
3.8.3 Experimental Study of the MSS-TL Protocol 43
3.8.3.1 Delay and data mules 44
3.8.3.2 Delay and privacy preservation 44
3.8.3.3 Comparison of MSS and MSS-TL protocol . . 44
3.9 MSS PROTOCOL - RANDOM WAYPOINT MODEL (MSS-RWP) 45
3.9.1 Experimental Study of the MSS-RWP Protocol 45
3.9.1.1 Comparison of MSS and MSS-RWP protocol . 46
3.10 RELATED WORKS 47
3.11 SUMMARY 48
4 ENERGY ADAPTIVE P2P FILE SHARING APPLICATIONS FOR DISASTER PREPAREDNESS, RELIEF AND RECOVERY OPERATIONS 50
4.1 RELATED WORKS 51
4.2 ASSUMPTIONS 53
4.2.1 Network Scenario 54
4.3 DESCRIPTION OF THE PROPOSED MECHANISM 55
4.4 FUNCTIONING OF THE PROPOSED MECHANISM 58

4.4.1 Initialization Phase 58

4.4.2 Bootstrapping Phase 59

4.4.3 File Download Phase 59

4.5 CHALLENGES AND SOLUTIONS IN IMPLEMENTATION OF THE PROPOSED MECHANIMS 60

4.5.1 Design and Creation of Energy Groups 60

4.5.2 Distribution of Additional Download Rate to Peers . . 61

4.6 IMPLEMENTATION AND SIMULATION OF THE PROPOSED MECHANISM 65

4.6.1 Modifications to the BitTorrent Protocol 65

4.6.2 Simulation and Results 66

4.6.3 Other P2P File Sharing Protocols 69

4.7 SUMMARY 70

5 E-DARWIN: ENERGY AWARE DISASTER RELIEF AND RECOVERY NETWORK 71

5.1 POST-DISASTER NETWORK SCENARIO 72

5.2 RELATED WORKS 73

5.3 OVERVIEW OF THE PROPOSED SOLUTION 75

5.4 E-DARWIN - NETWORK ARCHITECTURE 76

5.4.1 Functioning of E-DARWIN 78

5.5 NETWORK INITIALIZATION AND DATA FORWARDING 80

5.5.1 Activation of E-DARWIN Application 80

5.5.2 Neighbor Discovery and Synchronization 82

5.5.3 Forwarding Data in E-DARWIN 90

5.6 ENERGY AWARE DISTRIBUTED COALITION FORMATION 94

5.6.1 Formulation of Distributed Coalition Formation Game . 98

5.6.2 Implementation of the Coalition Game 100

5.7 PERFORMANCE EVALUATION 100

5.7.1 Prototype-based Evaluation 100

5.7.2 Simulations 103

5.8 UTILIZING RESCUE WORKERS TO COLLECT DATA IN THE E-DARWIN ARCHITECTURE 109

5.8.1 Interaction with Rescue Worker Devices 110

5.9 SUMMARY 111

6 CONCLUSION 113

BIBLIOGRAPHY 115

VITA 127

LIST OF ILLUSTRATIONS

Figure Page

1.1 Phases of Disaster Management 1

3.1 Panda Hunter Game 13

3.2 Traffic Flow in Phantom Routing 16

3.3 Error in Estimated Direction of Source 18

3.4 The Estimated Direction of the Source with Varying Distance from BS . 19

3.5 α-Angle Anonymity 20

3.6 Average Buffering Delay at Data Source in MSS Protocol 31

3.7 Average Buffering Delay at Data Mules in MSS Protocol 31

3.8 Average Total Delay with Varying Number of Data Mules in MSS Protocol 32

3.9 Average Total Delay with Varying α in MSS Protocol 33

3.10 Average Buffering Delay at the Data Source in the MSS-SP Protocol . . 37

3.11 Average Total Delay at Data Mule in MSS-SP Protocol 38

3.12 Comparison of the Average Carrying Delay at Data Mule in MSS and MSS-SP Protocol 38

3.13 Comparison of Average Total Delay at Data Mule in MSS and MSS-SP Protocol 39

3.14 Network Scenario for MSS-TL Protocol 40

3.15 Comparison of Average Total Delay of the MSS-TL and MSS Protocol . 44

3.16 Average Buffering Delay at Data Source in MSS-SP Protocol 46

4.1 Interaction Between Peers in and Across Groups 58

4.2 Average Download Rate of Peers with Varying Additional Download Rate 67

4.3 Average Energy Consumption of Peers with Varying Additional Download Rate 67

4.4 Average Download Rate of Peers with Varying Arrival Rate 68

5.1 Network Architecture of E-DARWIN 74

5.2 Functioning of E-DARWIN 78

5.3 Waveform of Captured Wail Siren 81

5.4 Spectrogram of Captured Wail Siren 82

5.5 Probability of Two Devices in Initialization Phase Discovering One Another 86

5.6 Probability of a Device in Initialize Phase discovering an Initialized Device having m neighbors with $n = 2$ and $T_{initial} = 4 \times C_{Discover} = 20$ mins 89

5.7 Probability of a Device in Initialize Phase discovering an Initialized Device having m neighbors with $n = 4$ and $T_{initial} = 4 \times C_{Discover} = 20$ mins 90

5.8 Energy and Time Consumed in State Transitions in Initialization Phase 101

5.9 Average Power Consumed in Client State in Initialization Phase 102

5.10 Average Power Consumed in Hotspot State in Initialization Phase . . . 103

5.11 Average Delay in Delivering Data to the AP in Data Forwarding Phase 105

5.12 Average Power Consumed in Data Forwarding Phase 106

5.13 Percentage of Data Delivered in Data Forwarding Phase 107

5.14 Convergence of the Coalition Formation Game with Varying Number of Devices and $\theta_1 = 100$ 108

5.15 Convergence of the Coalition Formation Game with Varying θ_1 108

5.16 Comparison of Energy Aware and Agnostic Approach 109

LIST OF TABLES

Table Page

3.1 Terminology Table in MSS Analysis 23

3.2 Simulation Scenario of MSS Protocol 29

5.1 Specifications of Sirens 81

5.2 Simulation Parameters for E-DARWIN Simulation 104

1. INTRODUCTION

The world's population is growing fast. The rapid, and often uncontrollable growth has made us increasingly vulnerable to both natural and man-made disasters, such as earthquakes, tsunamis, tornadoes, and structural collapses. In order to cope with such disasters in a fast and coordinated manner, effective disaster management techniques are needed.

1.1. DISASTER MANAGEMENT

Figure 1.1. Phases of Disaster Management

Disaster management is defined as the efficient utilization of resources (e.g. sensors, smartphones and tablets) to deal with various technological and humanitarian aspects of emergencies (6). It aims at reducing or avoiding the potential losses from emergencies, providing necessary assistance to victims, and aiding in rapid relief and

recovery. The process of disaster management can be broadly divided into four phases, namely, prevention, preparedness, relief and recovery (6; 35; 57; 74), as illustrated in Figure 1.1.

1.1.1. Disaster Prevention. The prevention phase is characterized by the measures taken for early detection of disasters and the corrective actions required for reducing or avoiding their catastrophic impact.

1.1.2. Disaster Preparedness. Local communities are always the first responders in the event of an disaster. The preparedness phase is designed to make them aware of the hazards they may face during or after a disaster, and how to efficiently as well as effectively respond to them. It aims at improving the state of readiness of people prior to a disaster by educating them of the potential hazards, vulnerabilities and appropriate responses.

1.1.3. Disaster Relief. Disaster relief operations involve organization and execution of activities combating a disaster and providing assistance to people in the immediate aftermath of a disaster. It focuses on primarily gaining situational awareness, identifying the immediate needs of the affected population, and addressing them.

1.1.4. Disaster Recovery. The recovery phase focuses on mobilization of resources for stabilizing and rebuilding the infrastructure in the affected region. It involves meeting both short-term and long-term infrastructural and communication requirements of the affected region.

The work discussed in this dissertation was conducted in an attempt to identify resources that could be used in disaster management. Specifically, it addresses various technological issues related to their efficient utilization during various phases of disaster management.

1.2. MOTIVATION AND CONTRIBUTIONS

Wireless sensor have been extensively used in literature for many monitoring and tracking based applications (11; 21; 33; 36; 69; 82; 101; 119; 130). In such applications, sensors have been deployed to create an ad hoc network for collecting data and detecting events that may cause disaster as well as provide early warnings. The early warnings can be used to take corrective measures for preventing the disasters from occurring or allow people to prepare for them.

However, a number of these applications are operated in hostile environments, like, conflict areas and wars. These networks face threats from elements both inside and outside the network. Previous works have focused primarily on optimizing the data collection and securing the communication between the sensor nodes. Little attention has been given to privacy. Preserving the sensor's location privacy implies that the flow of traffic in the network is resistant to tracking, i.e., the location of the sensors cannot be revealed. The location of the object being monitored is tightly coupled with the sensor that is either detecting or tracking it. Thus, preserving the data source's location is important to protect the sensor or the object from being traced. This work addresses the problem by designing protocols for preserving the location privacy of the data sources in wireless sensor networks.

Disasters can occur with varying degree of temporal and spatial dimensions. Utilizing sensors in large-scale disaster preparedness, relief, and recovery operations is infeasible given the amount of resources required. With the number of wireless devices (e.g., smartphones, personal digital assistants (PDAs) and tablets), gaining prevalence worldwide (3), they can be assumed to be abundantly present in the affected areas (26; 46; 92). As these devices are already available in the affected region, they can form an ad hoc network to aid in disaster preparedness, relief, and recovery operations.

Disaster preparedness involves making people aware of the risks they face in the event of a disaster and how to actively prepare for them. Many governmental and human response organizations have created databases of such information. The information can be downloaded to wireless devices using a peer-to-peer (P2P) file sharing applications (4; 10; 88) over the ad hoc network. Unfortunately, P2P file sharing applications are known to be bandwidth intensive and as a result may exhaust the device's battery before the download is complete. A number of studies have explored mechanisms to extend the device's lifetime by maximizing the device's sleep cycle (72; 116; 117). Such approaches known as *energy efficient computing* have their limitations as the amount of energy conserved depends on the of resources demanded by the applications. Consecutively, they cannot ensure that the applications will finish their activities within the constraints of the device's available energy. A more fundamental approach involves allowing the applications to define an energy budget for the tasks they are going to execute. The applications could then adapt their resource usage so that the task could be completed within the energy budget. Such approaches are known as *energy adaptive computing* (61; 62) and can maximize the lifetime of the devices (43). To this end, this work presents an energy-aware P2P file sharing application, which aims at providing differentiated service to devices based on their available energy. The differentiated service enables the devices to successfully download the file within the constraints of their available energy.

Wireless devices can further aid in disaster relief and recovery operations. In the event of a disaster, the immediate availability of health, environmental and infrastructure data is crucial for relief and recovery operations. With the network infrastructure destroyed, these devices can be used to rapidly build the communication network in the affected region. This work explores the use of wireless devices and technologies available on them for creating the desired ad hoc network infrastructure. Additionally, wireless devices are increasingly being equipped with multi-modal

sensors, such as temperature, accelerometer, pressure, GPS, microphone and camera. As a result, they can act as rich sources of sensory information in disaster scenarios (46; 114). As the network is crucial to relief and rescue efforts, the device's lifetime must also be maximized. To address this problem, this work designs novel mechanisms for autonomous organization of devices, distribution of data capturing task among them based on their available energy, and collecting the data with minimum delay.

1.3. ORGANIZATION

A survey of state-of-art research work in utilization of various resources for disaster management has been presented in Chapter 2. Chapter 3 presents a novel data source location privacy preserving protocol for disaster prevention. Chapter 4, introduces an energy-adaptive peer-to-peer (P2P) file sharing protocol for efficient distribution of content among the users to help them better prepare for disasters. Chapter 5 presents solutions for autonomous ad hoc network creation and efficient organization of the devices for capturing and collecting data in post-disaster scenarios.

2. AN INTRODUCTION TO DISASTER MANAGEMENT: A TECHNOLOGICAL PERSPECTIVE

Technology is changing how people prepare, respond and recover from disasters. Technological resources (e.g., sensors, smartphones, and tablets) and associated services (e.g., SMS and social networks) have increased the capability of people to capture, collect and disseminate useful and actionable information for disaster management. This has significantly reduced the number of people who have died or have been affected as a result of disaster in the last three years (51; 92). At the same time efforts must be made to continuously improve and innovate to make disaster prevention, preparedness, recovery and relief more effective and efficient.

2.1. DISASTER PREVENTION

Disasters can happen as a result of a number of hazardous events of varying spatial and temporal dimensions. Timely detection of these events is crucial for early prediction and prevention of the disasters. Wireless sensor networks (WSNs) have been extensively used in many monitoring and tracking based applications for early detection of such hazardous events (11; 48; 94; 135). These applications can be categorized in two categories based on the type of events they monitor, i.e., natural events and events caused by human activities.

2.1.1. Use of Wireless Sensor Networks for Disaster Prevention. Underwater wireless sensor networks have been used to monitor seismic activities in ocean beds to detect earthquakes and issue early tsunami warnings (12; 118). On land, WSNs have been instrumental in collecting data for early prediction of a number of natural disasters. For example, WSNs have been used for collecting meteorological

and hydrological data for early detection of landslides and flash floods (31), and monitoring seismic activities for predicting volcanic eruptions (120; 128).

Rapid urbanization has adversely impacted natural resources, which can result in an unbalanced ecosystem. An unbalanced ecosystem can cause unpredictable climate changes, which will result in widespread disasters. WSNs have been used to monitor the environment and provide us with the necessary feedback for maintaining a balanced ecosystem, like, pollution monitoring (135), habitat monitoring (36; 82; 101; 119), and wildfire monitoring (39; 54). Furthermore, WSNs have also been used for monitoring the structural integrity of civic infrastructures and provide timely warnings for applying reinforcements before a collapse occurs (69; 130).

2.1.2. Challenges in Use of Wireless Sensor Networks for Disaster Prevention. The operating environment of the senors can be remote, uncertain and dynamic in these applications. Thus, the deployment of wireless sensor network for disaster prevention poses significant technical challenges, such as neighbor discovery, data gathering, and security (33).

2.1.2.1. Neighbor discovery. Knowledge of the network is essential for a sensor in the network to operate properly. The sensors must be aware of the identity and location of other sensors in their communication range. Only after neighbor discovery, sensors in the network can communicate with other sensors in the network for collaborative data capture and gathering. Two sensors can discover one another if they have their radios turned on simultaneously (44). A trivial solution to this problem is that the sensors always keep their radios on. However, keeping the radio always on can significantly drain the battery of the sensors, which is not desirable. As a compromise, sensors only switch on their radios periodically for fixed time intervals, with the portion of time in the ON state characterized by duty cycle. The neighbor discovery protocols aim at modifying the duty cycling schedule of the sensors to help them discover one another faster, and with minimum energy consumption. These

protocols can be roughly classified into four categories based on their underlying design principle: randomness, over-half occupation, rotation-resistant intersection, and coprime cycles (71). Randomness based protocols allow the devices to turn on their radios with a given probability during discrete time slots. By exploiting the Birthday paradox, fast neighbor discovery is achieved on an average while keeping the duty cycle low (84). One-half based protocols work on the simple principle that neighbor discovery is guaranteed if the sensors radios are turned on for at least half of the period. Evidently, the protocol has high energy cost, and several modifications to the approach have been proposed to reduce the energy consumption (24; 71). Rotation-resistant intersection based protocols aim at scheduling the ON state of the sensors in discrete time slots such that the at least two sensors are active in the same slot. The problem is modeled as a block design problem, where the object is to find a schedule that will result in minimum energy consumption for the sensors (124). Finally, discovery can also be guaranteed with coprime cycles as given by Chinese Remainder Theorem (41; 56; 60).

2.1.2.2. Data gathering. On discovering their neighbors, the sensors in the network must collaborate with one another to capture and collect data. Multihop routing protocols must be designed to let sensors communicate with one another and collect data. As the operating environment of the sensors can be remote, the network must operate autonomously while continuously adopting itself to the ever changing energy, bandwidth and processing constraints of the sensors. A number of approaches have been discussed in literature for designing robust routing protocols (11). They can be classified as data-agnostic protocols and data-centric protocols based on their design principle.

Data-agnostic protocols route the data to the base station or sink without processing the data. They aim at finding the optimal route to deliver the data from the data source to the sink. A number of approaches have been proposed in literature

to select route based on varying characteristics. The characteristics can vary from finding a route with the best link quality, minimum number of hops, minimum energy consumption, maximum available energy among the sensor, and any combination of these.

In data-centric protocols, the sink sends a query to the network and the relevant sensors reply with the data. As data is being requested through queries, attributed-based naming is needed to describe the data. SPIN (55) was the first data-centric protocol, which considers data negotiation between nodes in order to eliminate redundant data and save energy. Directed Diffusion was developed later, which performs in-network data agreegration by eliminating redundancy, minimizing the number of transmissions and energy consumption (134). Many variants of the directed diffusion protocols have been proposed with varying degree of optimization (13).

2.1.3. Security. The operating environment of the sensors can be hostile. Thus, security should be built into their design. Novel techniques are needed for creation of low-latency, energy-efficient, and secure networks. The communication between the sensors must be secure to prevent leakage of information. To ensure seamless functioning of the sensors, the network must be protected against intrusion and spoofing.

2.2. DISASTER PREPAREDNESS

Current approaches in disaster preparedness involve design of social media platforms and mobile applications for pushing disaster-related information to the suers. Social media has brought a fundamental change in how humans communicate with one another. The propoularity of various social networks has prompted humanitarian response organization to use it for educating people of the risks they might face in the event of a disaster. For example, the American Red Cross Digital Operations

Center (DigiDOC) has set up a social media platform for spreading disaster awareness and preparedness information (92). The system relies on a set of trained digital volunteers working remotely to provide people with real-time tips and resources.

To overcome the restrictions of availability of trained volunteers, a number of online websites have been established by various government organizations (e.g., ready.gov (10) and the Indian Resource Network (88)), and humanitarian response organizations (e.g., disasterready.org (4)) to provide resources to users on-demand. The sites provide content in the form of both text-based and illustrative documents with varying degree of details about the disasters and the corresponding actions people must take to keep themselves safe. Additionally, the sites also provide a number of instructional videos as visual demonstration of the actions they must take in the event of the disaster, like, how to give first aid. The videos can be of varying length, ranging from a few minutes to an hour depending on the content. Furthermore, to improve accessibility and distribution of the content, American Red Cross has designed a suite of mobile phone applications for each disaster. The disaster specific applications can be used to access the corresponding disaster preparedness information by the users.

2.3. DISASTER RELIEF AND RECOVERY

Social networks can also act as a crucial component in disaster relief and response operations. In the event of a disaste,r people use a variety of social networking platforms, like, Instagram, Facebook, and Twitter, to share personal and local information. For example, in Japan, Twitter users posted more than 177 million disaster-related tweets the day after the 2011 earthquake (92). A number of data-driven approaches have been discussed in literature to extract useful information from user posts (112). In disaster scenarios, they can be used to gain situational awareness and aid in relief and recovery operations.

Cellular networks are the fundamental infrastructure being used by people for communicating on social networks (3). However, the cellular infrastructure is vulnerable to large-scale disasters. Even in small-scale disasters, e.g. the storm in west Norway in December 2011, thousands of people can lose network connectivity for weeks (49). As the availability of the network is not guaranteed in disaster scenarios, users may not always be able to access social networks. Thus, building communication capabilities in the affected region is one of the primary challenges in post-disaster scenarios. Using sensors for constructing a disaster recovery network has been extensively explored in literature (22; 45; 50). However, making use of wireless devices (e.g, smartphones and tablets) widely available among people in the affected region is a more practical solution (46; 114). If accessibility is available to the affected region, mobile base stations can be deployed there with satellite gateways. The infrastructure can be used in conjunction with ad hoc networks of wireless devices (32; 47; 81; 107) to achieve large-scale coverage and communication.

3. A SOURCE LOCATION PRIVACY PRESERVING PROTOCOL FOR DISASTER PPREVENTION IN AD HOC NETWORKS

The world is full of hazards. Hazards are extreme or rare events that can adversely impact humans to the extent of causing disaster. These hazards result from both natural causes as well as human activities and can affect us directly or indirectly. The importance of early detection of these hazards and issuing early warning signals cannot be overstated. For example, underwater earthquakes can be detected to issue early timely tsunami warnings and save countless lives (12).

Traditionally, wireless sensor networks (WSNs) have been deployed to create ad hoc networks for collecting data and early detection of hazards in many monitoring and tracking based applications (11), like, structural monitoring (69; 130), habitat monitoring (36; 82; 101; 119) as well as reconnaissance and surveillance (21; 33). Unfortunately, the operating environment of these sensors can be hostile. The monitoring process can be either hindered or exploited by both internal and external threats. For example, wireless sensors are being used in many sensitive areas (e.g. borders and conflict zones) to monitor, detect and track intruders (37; 89). However, as the monitoring area can be large in some scenarios, it may not always be possible to achieve complete coverage due to infrastructure costs. Intruders (e.g. terrorists) may monitor the sensor transmissions to discover their locations and identify coverage gaps for avoiding detection (73; 127). A number of approaches have been discussed in literature to secure the communication between the sensors (40; 63; 98; 99). However, little attention has been given to privacy. Preserving the sensor location privacy implies making them resistant to tracking. The location of the object being monitored or tracked is tightly coupled with the sensor detecting it, also called the data source. An adversary can easily monitor the radio transmissions in the network and discover the location of the data source. The information can then be exploited to

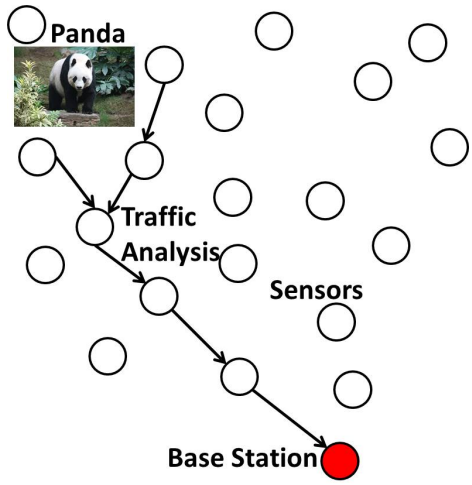

Figure 3.1. Panda Hunter Game

launch attacks against the data source or the objects being monitored. Hence, source location privacy preserving routing protocols must be designed.

The example of "Panda-Hunter Game" (59) is used to further illustrate the problem of preserving data source location privacy, as shown in Figure 3.1. Extinction of species can have a significant impact on the environment as it disrupts the natural ecological balance. A disrupted ecosystem can lead to catastrophic changes in climate and environment. Pandas are one such animal, which are on the verge of extinction. They live in bamboo forests and facilitate the growth of vegetation. Thus, protecting the pandas is crucial to sustainability of the forests and maintenance of the ecosystem. Sensors are deployed in the forest to monitor the health and movement of pandas. Each panda is mounted with an actuator that signals the surrounding sensors in its communication range. When the sensor close to the panda receives the signal, it creates and sends data reports to the base station over multi-hop wireless network. A hunter who is monitoring the wireless communication between the sensors will be able to identify the direction of traffic flow. He can trace back the data transmission path to locate the data source, thus, catching the panda. Any WSNs used for such monitoring applications are vulnerable to such kinds of traffic analysis-based attacks.

A realistic attack model, called *the semi-global eavesdropping model*, is first proposed in this chapter. In the proposed attack model, an attacker with limited monitoring capability utilizes a linear-regression based traffic analysis approach to discover the location of data source. An existing data source location privacy preservation routing protocol was then broken to demonstrate the effectiveness of the proposed attack model. Having demonstrated the feasibility of the proposed attack model, an α-angle anonymity model is defined here to study source-location privacy and a Mules-Saving-Source protocol (MSS) is proposed for preserving the source location privacy. The MSS protocol uses data mules to modify the flow of traffic in the network and achieve α-angle anonymity. The protocol is then theoretically analyzed to identify its shortcoming and several variants are proposed to overcome them. More specifically, a Mule-Saving-Source - Shortest Path Protocol (MSS-SP), is proposed to reduce the buffering time at the data mules by modifying the data mules delivery paths. A Mule-Saving-Source - two level (MSS-TL) is also proposed to reduce the total delay by restricting the movement of data mules to local areas in the network. Furthermore, the impact of mobility pattern of the data mules on the MSS protocol is studied by changing the mobility model of data mules to Random Waypoint based mobility model. Finally, both theoretical analysis and comprehensive set of experiments are used to evaluate the proposed variants effectiveness and draw comparisons between them. The research work presented in this chapter has been published in (77; 106).

3.1. SYSTEM MODEL

The underlying network's terrain is assumed to be a finite two-dimensional grid that is divided into cells of equal size. The network is comprised of a base station, static sensors, and mobile agents (also known as data mules).

3.1.1. Static Sensors. All static sensors are assumed to be homogeneous; they have the same lifetime, storage, processing, and communication capabilities. They are deployed uniformly at random in the cells and assumed to guarantee the network's connectivity.

3.1.2. Data Mules. Data mules are mobile agents that can be artificially introduced into the network (113). It is assumed that the data mules move independent of one another, do not communicate with one another, and always know their location. Their mobility is modeled as a random walk on the grid whereby in each transition a data mule moves with equal probability to one of the horizontally or vertically adjacent cells. After a data mule moves into a cell, it stays there for t_{pause} time before its next transition. At the beginning of the pause interval, the data mule announces its arrival by broadcasting a Hello message. On receiving the Hello message, only the data source responds and relays the buffered data to the data mule. It is also assumed that the data mules do not communicate with sensors while moving and their communication range is larger than that of a sensor. Thus, a data source that cannot communicate directly with the data mule uses multi-hop routing to communicate with it.

3.2. ATTACK MODEL

The attacker is assumed to be capable of launching only passive attacks. During these attacks, it can only monitor the data transmission; it can neither decrypt nor modify data packets. It is assumed that the attacker monitors the radio transmissions between sensors in a circular area of radius R_{att}, as shown in Figure 3.2. The monitoring area of the attacker represents his attack strength. Thus, the strength of the attacker increases with the monitoring area. If the monitoring area is large enough to cover the entire network, the attack model is called global eavesdropping.

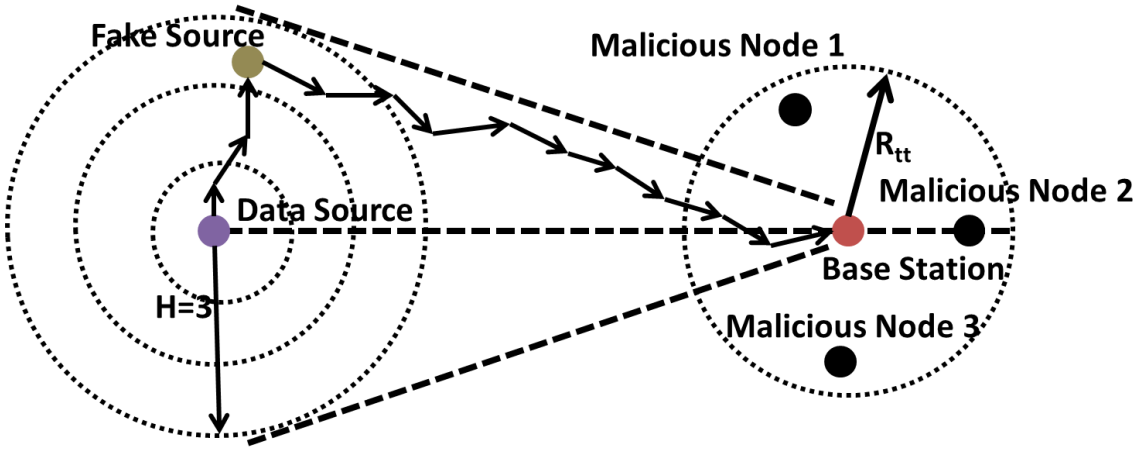

Figure 3.2. Traffic Flow in Phantom Routing

In contrast, if the area is limited only to a few hops, the attack model is called local eavesdropping. *Semi-global eavesdropping* attack model is defined here as an attack model whose strength exists between the two extreme attack models. Using the Semi-global eavesdropping attack model, an attacker will launch an attack by collecting traffic around the base station. Intuitively, the base station serves as the ideal point to start the attack as all the network's traffic converges to it. Using the direction of incoming traffic, the attacker can make an initial estimation of the direction in which the data source lies and move in that direction. The attacker can then continuously keep updating its estimation as it moves until the data source is located. Thus, to defeat such an attacker, he must be discouraged from making a good initial estimation of the data source's direction before it starts moving.

3.2.1. Linear Regression Based Traffic Analysis. The attacker begins the attack by observing incoming traffic around the base station and then analyzes them to estimate the direction of the data source. This estimation, however, is not straightforward. The observed transmission paths are neither linear nor constant due to multi-hop data routing or randomness introduced by the routing protocols, such as the random H hops in phantom routing (59). Linear-regression (109) is used in this

work to identify the best fit line that represents transmission path from data source to base station of each data packet. All regression lines are forced to pass through the base station as all the data is delivered to the base station. Ideally, in the absence of spatial randomness introduced along the routing path, walking along the regression line would reveal the source's location.

To estimate the direction of data source, a *traffic vector* is defined with unit magnitude for each data packet observed by the attacker. The vector's direction is given by the direction of the regression line representing the transmission path of the data packet. By doing so, there is a traffic vector for each transmission path observed by the attacker. The direction of the data source can be inferred from the direction of the composite vector formed by summing up the traffic vectors defined for each transmission path.

3.2.2. Compromising Phantom Routing. The attack model and the traffic analysis approach will reveal the direction of the data source in phantom routing. *Phantom Routing Protocol* (59) requires each data packet to be randomly routed H hops from data source. The Hth hop sensor in the random routing is called a *fake source*. The data is then forwarded to the base station along the shortest path. Using this protocol, a backtracking attacker will fail to identify the real data source due to the random H-hop routing. Assuming all sensors are deployed uniformly at random in the network, the statistical distribution of fake sources around the data source forms concentric rings. Sensors on the same ring have the same probability to become fake sources. This is due to two facts; (1) symmetric deployment of sensors around the real data source, and (2) in the first phase of the routing, the next-hop sensors are selected uniformly at random from the surrounding sensors within the transmission range.

Since the fake sources are distributed symmetrically around the data source, the composite traffic vector for all data transmission paths will give the direction

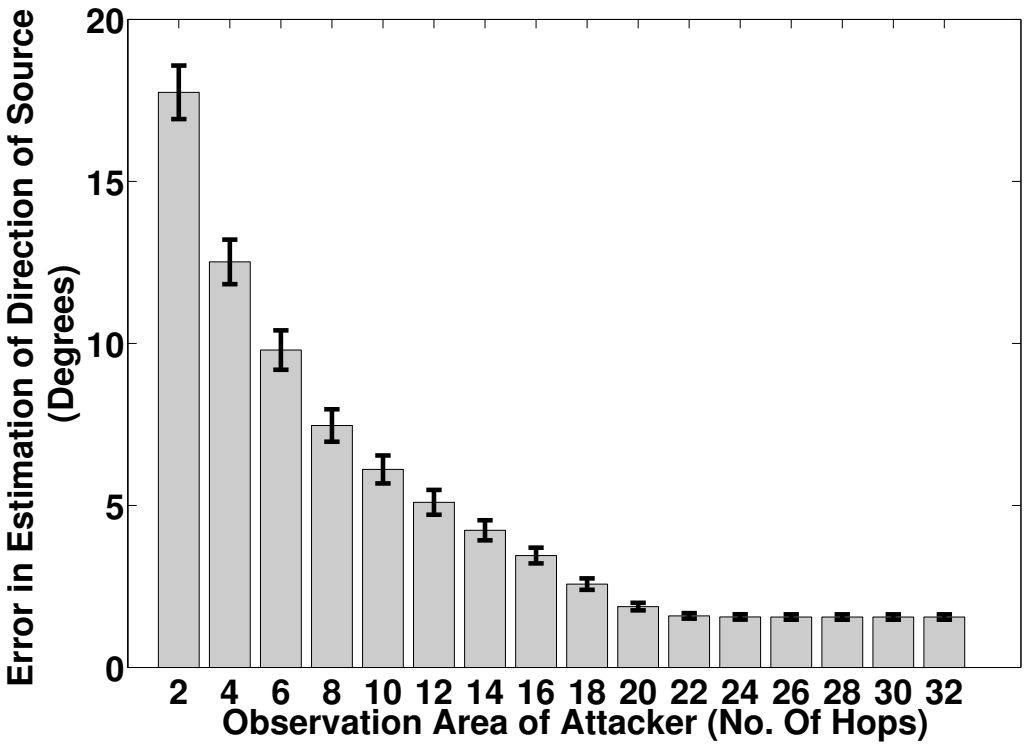

Figure 3.3. Error in Estimated Direction of Source

of the data source. The following simulations were conducted to confirm this claim. Phantom routing was configured with $H = 8$ and the cost of launching the attack over 1000 trials was analyzed. The cost of launching an attack was determined as being equal to the attacker's monitoring area. As shown in Figure 3.3, when the attacker's monitoring area is restricted to a few hops of 2 or 4 (i.e., local eavesdropping), the estimated error is very high as compared to the scenarios when the attacker observes transmissions over a larger number of hops. As a result, the attacker will move further away from the source as he moves along the estimated direction. Hence, the protocol will provide defense against the local eavesdropping adversary. The inference of the source's direction becomes more accurate as the size of the monitoring areas is increased. The cost of launching an attack is further analyzed as the amount of data packets required for the attacker to make a good estimation of the data source's

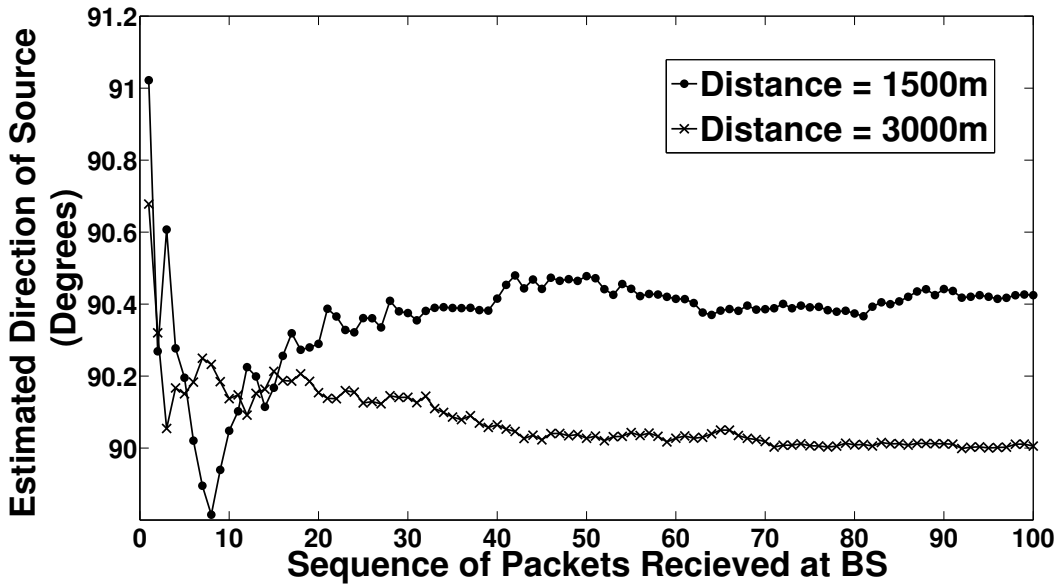

Figure 3.4. The Estimated Direction of the Source with Varying Distance from BS

direction. As shown in Figure 3.4, a semi-global eavesdropper is able to infer the source's direction without observing a large amount of data transmission, i.e. around 60 packets. Therefore, one can see that the proposed attack model is effective in compromising the phantom routing.

3.3. THE α-ANGLE ANONYMITY

In order to anonymize the source's location under a semi-global eavesdropping attack, the α-angle anonymity model is defined here. The model ensures the preservation of source location privacy by enlarging the inference space from which the attacker estimates the real direction of the data source. The inference space is determined by the system variable α. The value of α can be open to the public, including the attacker. However, the knowledge of α should not threaten the privacy of source location. Based on the definition, it can be seen that the larger the value α, the larger the inference area. The shaded area in Figure 3.5 represents the attacker's

inference space. The attacker cannot deterministically estimate either the real direction or the data source's location when the inference space is larger. Thus, source's location privacy is preserved.

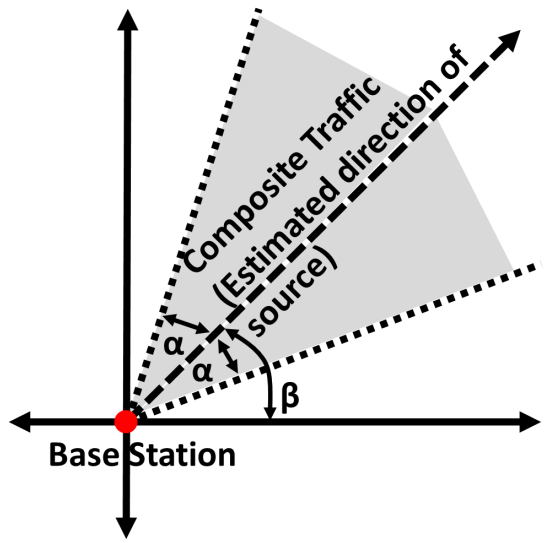

Figure 3.5. α-Angle Anonymity

Definition 3.3.1 ***α-Angle Anonymity***. *A protocol is α-angle anonymous if the real direction of data source is equally likely distributed in the angle range $[\beta-\alpha, \beta+\alpha]$, where β is the angle of the direction inferred by the attacker based on his observation.*

3.4. MULES-SAVING-SOURCE (MSS) PROTOCOL

The *Mules-Saving-Source* protocol was designed to protect data source location privacy against a semi-global eavesdropper and achieve α-angle anonymity. The protocol exploits the random mobility of data mules to establish a false data transmission path, which effectively preserve the location privacy of data source. Specifically, it modifies the traditional function of data mules by having them offload data to regular sensors at only specific locations in the network. The data is then routed towards base station along the shortest path. The sensors to which the data will be offloaded

are selected so as to bias the direction of composite vector estimated by the attacker based on data transmissions he observes around base station.

In fact, solely allowing data mules to deliver data directly to base station can completely preserve source location privacy against a semi-global eavesdropper. This is because the data transmission between data source and base station is completely hidden by the random movement of the data mules which ferry the data. Such a protocol is called Direct Delivery (DD) protocol. However, delivering data directly to the base station results in non-trivial delay, which may not tolerable in large-scale wireless sensor networks.

In this section, the MSS protocol is described first and is then proved to be α-angle anonymous. The protocol defines a coordinate system with the base station at the origin, which is assumed to be known to all data mules. The protocol includes three phases, 1) choose a fake direction at the source, 2) use data mules to carry and unload the data, and 3) route the data to the base station.

Phase 1. Choose a fake direction at the source - When a target is detected by the sensors, the sensors coordinate among themselves and allow the sensor closest to the target to become the data source (137). The data source periodically generates and sends data reports towards base station. It also generates a value of β as the fake direction of data source. The fake direction is used to bias the attacker's observation of the direction of the incoming traffic at the base station. More specifically, the data source selects β uniformly at random from the range $[\theta - \alpha, \theta + \alpha]$, where θ is the absolute angle between the direction of the data source and x-axis in the coordinate system. α is a predefined as the privacy preservation level of the network.

Phase 2. Using data mules to carry and unload the data - When a data mule moves into a cell, only the data source within its communication range responds with the buffered packets. The data source also sends the value of β angle to the data mule. Once data mule receives the data, it roams around the network until it reaches

the *dropping point*. The dropping point is referred to as any point located on the *dropping line*. The *Dropping Line* is the line drawn from base station at an angle β in the coordinate system. After arriving in a cell that intersect with the dropping line (also called as dropping cell), the data mule offloads the data to the sensor that is closest to the dropping line in that cell.

Phase 3. Route the data at sensors - After the data packets are offloaded to a sensor by the data mule, they are routed towards the base station along the shortest path. Ideally, the transmission path will be along the dropping line. The data transmission path may have trivial deviation from the dropping line due to the nonlinear multi-hop routing. However, this does not affects privacy preservation as the traffic flow moves towards the base station roughly along the β angle direction, thereby, successfully biasing the attacker's inference of data source direction.

The MSS protocol's effectiveness for preserving source location privacy is further demonstrated by Theorem 3.4.1.

Theorem 3.4.1 *Mules-Saving-Source protocol is α-angle anonymous for source location privacy.*

Proof All the data from a data source in MSS protocol is forced to come towards base station, along the fake direction, at an angle of β. Thus, the composite traffic vector will be along the same direction. Although the attacker knows the rule of selecting the fake direction $\theta - \alpha \leq \beta \leq \theta + \alpha$, where θ is the absolute angle of the source direction, which is unknown to the attacker. Thus, he can only deduce that the data source lies in a region given by $\beta - \alpha \leq \theta \leq \beta + \alpha$. Therefore, MSS achieves α-angle anonymity in terms of Definition 3.3.1.

Table 3.1. Terminology Table in MSS Analysis

Notations	**Definitions**
N_{mules}	The number of data mules in the network
L_n	The length of the network's $(L_n \times L_n)$ edge
L_c	The length of each cell's $(L_c \times L_c)$ edge
v_{mule}	The velocity of each data mule's movement
t_{move}	The transition time of data mule from one cell to another (L_c/v_{mule})
t_{pause}	The pausing time of data mule in each cell
D_{src}	Random variable for the data source buffering time
D_{mule}	Random variable for the data mule carrying time
E_{absorb}	The expected number of transitions until data packet reaches any absorbing state
$E_{D_{mule}}$	The expected delay at data mule
L_{lc}	The length of each large cell $(L_{lc} \times L_{lc})$

3.5. ANALYSIS OF MSS PROTOCOL

Table 3.1 describes the notations used in the analysis of the MSS protocol. A discrete-time Markov chain is used here to model the mobility pattern of the data mules (113). Each state in the Markov chain represents the condition when the data mule is present in a particular cell in the network. Let P be the transitional probability matrix for the defined Markov chain model, where the entry $p_{ij} \in P$ represents the probability of a data mule transitioning from one state s_i to another state s_j for a Markov chain with state space S.

$$p_{ij} = \begin{cases} \frac{1}{q}, & \text{if } s_i \text{ and } s_j \text{ are adjacent} \\ 0, & \text{Otherwise} \end{cases} \tag{3.1}$$

In Equation 3.1, two states being adjacent means that their corresponding cells are adjacent to one another either horizontally or vertically, and q is the number of adjacent states of s_i. Additionally, it is assumed that the mobility pattern of data mules has achieved stationary distribution.

The MSS protocol is compared with the direct delivery protocol (DD) in the following discussion. The end-to-end data delivery delay (D_{total}^{MSS}) in MSS protocol consists of two parts, (1) D_{src}^{MSS} - the data buffering time at data source until a data mule picks it up, and (2) D_{mule}^{MSS} - the data buffering time at the data mule until it offloads the data to a sensor. The transmission delay from the the static sensors to the base station is assumed to be comparatively trivial, and, thus, ignorable. One time unit (t_{unit}) is defined as the total time spent in one transition, i.e., $t_{unit} = t_{move} + t_{pause}$.

3.5.1. Buffering Delay at Data Source. The analysis of the buffering delay at data source (D_{src}) is the same for MSS (D_{src}^{MSS}) and DD (D_{src}^{MSS}) protocols, i.e. $D_{src} = D_{src}^{MSS} = D_{src}^{DD}$, as this delay is not affected by where data mules drop data. One analytic result derived in (113), formalizing the distribution of buffering delay at data source (D_{src}) is used here, as shown in Equation 3.2. Here, one time unit is given by t_{unit}, and the buffering capacity at data source is assumed to be unlimited. From Equation 3.2, one can see that the probability of having small buffering delay at the data source increases with the number of data mules and decreases with the number of cells in the network $(\frac{L_n}{L_c})^2$.

$$P\{D_{src} \leq t\} \approx 1 - \exp\left(\frac{-t}{0.68\frac{L_n^2}{L_c^2 \times N_{mules}}\log(\frac{L_n}{L_c})}\right) \tag{3.2}$$

3.5.2. Carrying Delay At Data Mules. The carrying delay at the data mule in the MSS protocol (D_{mule}^{MSS}) differs from that in DD protocol (D_{mule}^{DD}). A result derived in (113), which is relevant to the carrying delay in DD protocol is first introduced here. The result is then used to compare the expected carrying delay at data mule for both MSS and DD protocols. Equation 3.3 derived in (113) formulates the delay distribution of carrying data at data mule with a single base station located at the center of the network. The carrying delay increases with the number of cells

in the network, $(\frac{L_n}{L_c})^2$. Additionally, it is associated with the moving speed of data mule, which is captured by the time unit definition.

$$P\{D_{mule}^{DD} \leq t\} \approx 1 - \exp\left(\frac{-t}{0.68(\frac{L_n}{L_c})^2 \log(\frac{L_n}{L_c})}\right) \tag{3.3}$$

As that the analysis given in Equation 3.3 assumes a single base station located at the center of the network, it represents the distribution of the delay of carrying data at data mules due to DD protocol. To analyze the carrying delay at data mules in the MSS protocol an absorbing Markov chain base model is used here.

Definition 3.5.1 *Absorbing Markov Chain. A state s_i in a Markov chain is called an absorbing state if the probability of staying in it after transitioning into it is one. The rest of the states which are not absorbing are called transient states. An absorbing Markov chain is one, which has at least one absorbing state and all absorbing states are reachable from each transient state.*

According to Definition 3.5.1, DD protocol can be modeled as an absorbing Markov chain with one absorbing state, in which the absorbing state is the state representing the event when a data mule is in the cell having the base station. The state is called an absorbing state due to the fact that when the data mule reaches that state, the data are delivered to the base station and they do not transit to adjacent cells with it any more. Thus, data transitions in DD protocol can be modeled as an absorbing Markov chain with a single absorbing state. On the other hand, the MSS protocol can be modeled as an absorbing Markov chain with multiple absorbing states. All cells which intersect with the dropping line form the absorbing states because the data carried by the data mule are unloaded to static sensors in those cells and no longer transit to adjacent cells with the data mule. Thus, data transitions in MSS can be modeled as an absorbing Markov chain with multiple absorbing states.

3.5.2.1. Expected carrying delay at data mules. Based on the absorbing Markov chain models, the expected delay of carrying data at data mules ($E_{D_{mule}}$) is equal to the expected number of transitions (E_{absorb}) until it reaches the absorbing state, as shown by Equation 3.4.

$$E_{D_{mule}} = E_{absorb} \times t_{unit} \tag{3.4}$$

E_{absorb} is computed as follows. Given an absorbing Markov chain, *Canonical Form* of transition matrix can be derived as shown in Equation 3.5, where there are t transient states and r absorbing states.

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{3.5}$$

where $\mathbf{I}$ is an r-by-r identity matrix, $\mathbf{0}$ is an r-by-t zero matrix, $\mathbf{R}$ is a nonzero t-by-r matrix representing the transition probabilities from transient states to absorbing states, and $\mathbf{Q}$ is a t-by-t matrix of the transition probabilities between transient states. Once the absorbing state is reached, the data remains in the absorbing state with probability 1. Based on the canonical form, the *Fundamental Matrix* N for an absorbing Markov chain is defined by Equation 3.6.

$$N = (I - Q)^{-1} \tag{3.6}$$

From (68), it is known that the expected number of transitions before the chain is absorbed is given as $\mathbf{w} = \mathbf{Nc}$. An ith entry $w_i \in w$ is the expected number of transitions before the chain is absorbed, given the chain starts from the ith transient state in $\mathbf{P}$. $\mathbf{c}$ is a column vectors with all entries equal to one. Assuming the probability that the markov chain will start in a transient state is $\frac{1}{t}$, E_{absorb} can be

computed using Equation 3.7.

$$E_{absorb} = \frac{1}{t}\sum_{i=1}^{t} w_i \tag{3.7}$$

By using the aforementioned formulas, the expected number of transitions until the chain is absorbed while starting from any state can be derived. The expected carrying delay for the DD protocol (E_{mule}^{DD}) can be computed using Equation 3.4 and Equation 3.7, as shown in Equation 3.8. In Equation 3.8, t = $(\frac{L_n}{L_c})^2 - 1$ as there is only one absorption state.

$$E_{mule}^{DD} = E_{absorb} \times t_{unit} = \frac{t_{unit}}{t}\sum_{i=1}^{t} w_i \tag{3.8}$$

Equation 3.8 can be used for calculating the expected carrying delay of data mules ($E_{mule}^{MSS}(\beta)$) for a given dropping angle β in MSS protocol with t given as $t = (\frac{L_n}{L_c})^2 - f(\beta)$, where $f(\beta)$ gives the number of absorbing states or the number of corresponding dropping cells in the network for a given β angle. The expected carrying delay of the data mule for a given privacy level of α angle ($ED_{mule}^{MSS}(\alpha)$) can be computed as shown in Equation 3.9.

$$ED_{mule}^{MSS}(\alpha) = \frac{\sum_{\beta=\theta-\alpha}^{\beta=\theta+\alpha} E_{mule}^{MSS}(\beta)}{2\alpha} \tag{3.9}$$

Where θ is the absolute angle between the direction of data source and the direction of x-axis in the coordinate system. Based on the model, it can be concluded that the expected number of transitions before the chain is absorbed in the MSS protocol is less than that of the DD protocol as the MSS protocol has more absorbing states than the DD protocol. Additionally, the absorbing state of DD protocol is included in the absorbing states of MSS protocol.

3.5.2.2. Expected total delay in MSS and DD protocol. The total delay of the data mules in the DD (D^{DD}_{total}) and MSS (D^{MSS}_{total}) protocols is the sum of buffering delay at the data source and the data mule for the respective protocols. Using linearity of expectation, the expected total delay in DD protocol is given as the sum of expected buffering delay at the data source (E^{DD}_{src}) and the expected buffering delay at the dat mule (E^{DD}_{mule}), i.e., $E^{DD}_{total} = E^{DD}_{src} + E^{DD}_{mule}$. Similarly, the expected total delay in MSS protocol due to dropping angle β is given as the sum of expected buffering delay at data source (E^{MSS}_{src}) and the expected buffering delay at data mule due to a dropping angle β ($E^{MSS}_{mule}(\beta)$), i.e., $E^{MSS}_{total}(\beta) = E^{MSS}_{src} + E^{MSS}_{mule}(\beta)$. Hence, the expected total delay due to a privacy level of α angle ($TD^{MSS}_{total}(\alpha)$) in MSS protocol can be computed as $TD^{MSS}_{total}(\alpha) = \frac{\sum_{\beta=\theta-\alpha}^{\beta=\theta+\alpha} E^{MSS}_{total}(\beta)}{2\alpha}$.

3.5.2.3. Lower bound of carrying delay at data mules. The minimum delay depends on the minimum number of transitions needed to enter the absorption state, called the lower bound of transition number (L), and can be determined by $L \times t_{unit}$. For the sake of simplifying the analytical description, each cell in the network is indexed by the indices of its row and column, respectively. For example, a cell in row i and column j is indexed as (i, j), and is represented as $cell_{(i,j)}$. Suppose the data source is located in $cell_{(i_{src},j_{src})}$, and base station is in $cell_{(i_{bs},j_{bs})}$. The cells corresponding to the multiple absorbing states are represented by $cell_{(i_{ab_k},j_{ab_k})}$ ($k = 1, ...n_{ab}$), where n_{ab} is the total number of absorbing states. In DD protocol $n_{ab} = 1$. The lower bound of carrying delay at data mules is computed as follows.

Lemma 3.5.2 *Given $cell_{(i_{src},j_{src})}$ and $cell_{(i_{ab_k},j_{ab_k})}$ which is associated with any absorbing state s_{ab_k}, $L_{s_{ab_k}} = |i_{src} - i_{ab_k}| + |j_{src} - j_{ab_k}|$.*

Proof Based on the mobility pattern of data mules, the minimum number of transitions required to reach the absorbing state s_{ab_k} from s_{src} is determined by the

Manhattan Distance of the two cells, $cell_{(i_{src},j_{src})}$ and $cell_{(i_{ab_k},j_{ab_k})}$, which is exactly $|i_{src} - i_{ab_k}| + |j_{src} - j_{ab_k}|$.

Theorem 3.5.3 *The lower bound of transition number in MSS protocol* (L_{MSS})*, is not larger than that in DD protocol* (L_{DD})*.*

Proof According to the definition of the lower bound transition number, it is known that $L_{MSS} = \min(L_{s_{ab_k}})$, where s_{ab_k} is any absorbing state. Since the base station is one of those absorbing states, $L_{MSS} \leq L_{s_{base\_station}}$. From $L_{s_{base\_station}} = L_{DD}$, it can be derived that $L_{MSS} \leq L_{DD}$. Therefore, the theorem holds.

3.6. EXPERIMENTAL STUDY OF THE MSS PROTOCOL

Table 3.2. Simulation Scenario of MSS Protocol

Total no of sensors	10000
Total deployment area	10^8 sq. m
Communication range of sensors	100 m
Data generation rate	1 per s
Speed of data mules	15m/s
Pause time of data mules	1 second
Dimensions of logical cells of data mules	200mx200m
Communication range of data mules	100 m
Total simulation time	15000 seconds

A comprehensive set of simulations were conducted using a customized C++ based simulator to evaluate the performance of MSS and DD protocols. The delay in

delivering data to the base station in the MSS protocol was studied with varying number of data mules and privacy levels (α angle). Furthermore, it was also established that the MSS protocol gives better performance than DD protocol.

The simulation configuration is given in Table 3.2. It was assumed that the base station is located at the center of the network at $(5000, 5000)$, and the data source is at $(5000, 8000)$. Sensors are deployed uniformly at random in the network. At initialization, a tree routing topology is constructed with the base station as the root node. The data mules were deployed uniformly at random in the network. The data mules move in the network based on the mobility model introduced in Section 3.5. The data mules move in logical cells of size $200m \times 200m$. In all the results discussed hereafter, each data point represents the average result of 1000 trials of each experiment.

3.6.1. Delay and Data Mules. The total data delay is constituted of two components, i.e., the buffering delay at the data source and the carrying delay at the data mules. As the number of data mules is increased in the network, the frequency with which they visit a sensor increases. Hence, the average buffering delay at the data source decreases with increase in the number of data mules, as shown in Figure 3.6. After a data mule picks up the data from the data source, the carrying delay solely depends on the location of the dropping cells and the mobility pattern of the data mules. Thus, the average carrying delay at the data mule is independent of the number of data mules, as shown in Figure 3.7. Therefore, the average total delay at data mule also decreases with increase in the number of data mules, as shown in Figure 3.8.

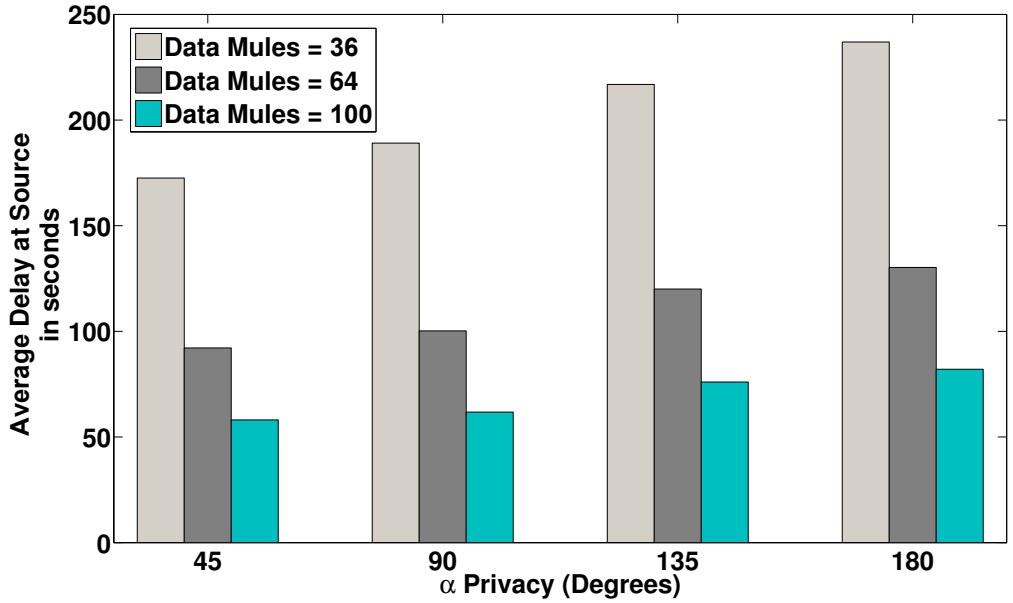

Figure 3.6. Average Buffering Delay at Data Source in MSS Protocol

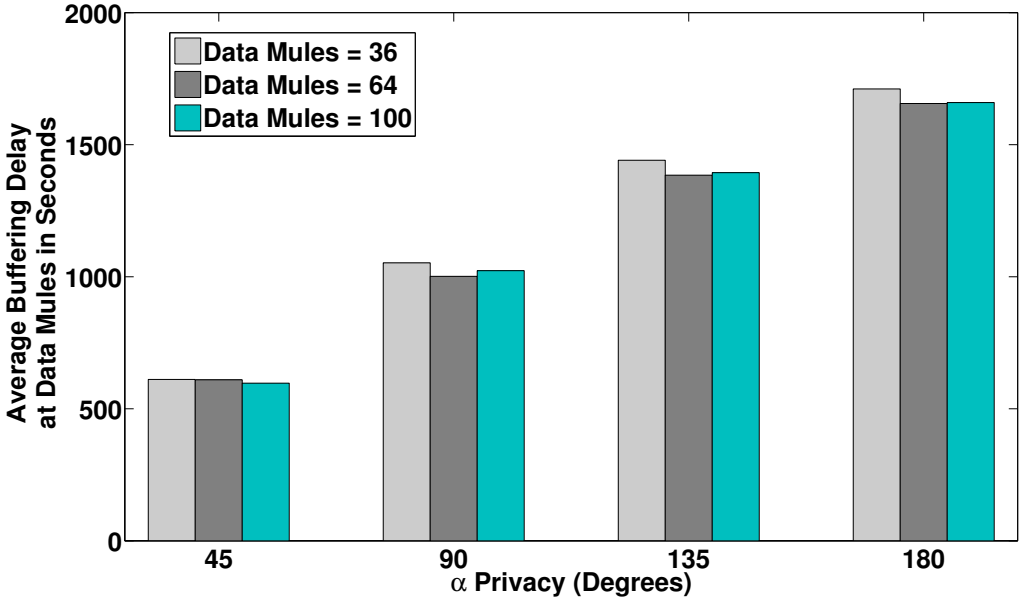

Figure 3.7. Average Buffering Delay at Data Mules in MSS Protocol

3.6.2. Delay and Privacy Preservation. α represents the privacy level in the network. For a large α, the attacker has to infer the real direction of source from a wider inference space, which reduces his probability of succeeding. However,

increasing the value of α has an adverse impact on the delay of the data packets. From Figure 3.6 it can be observed that for a given number of data mules, varying α does not significantly influence the buffering delay at data source. This is because the buffering delay at source is impacted by the network configuration, such as the number of data mules rather than the α value as shown in Equation 3.2. The carrying delay at the data mule increases with α for a given number of data mules, as shown in Figure 3.7. This is because setting a larger value for α leads to the possibility of selecting a β value with higher deviation from the data source. As data mules can choose dropping points further away from the real source to offload data, it causes a larger carrying delay. This results in an increase in total delay with increasing α, as shown in Figure 3.8.

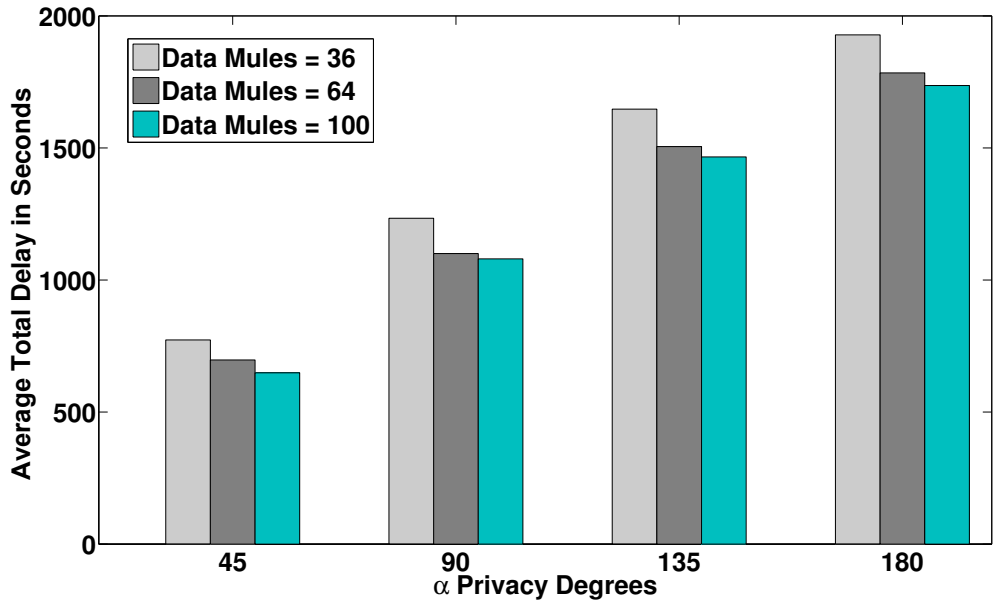

Figure 3.8. Average Total Delay with Varying Number of Data Mules in MSS Protocol

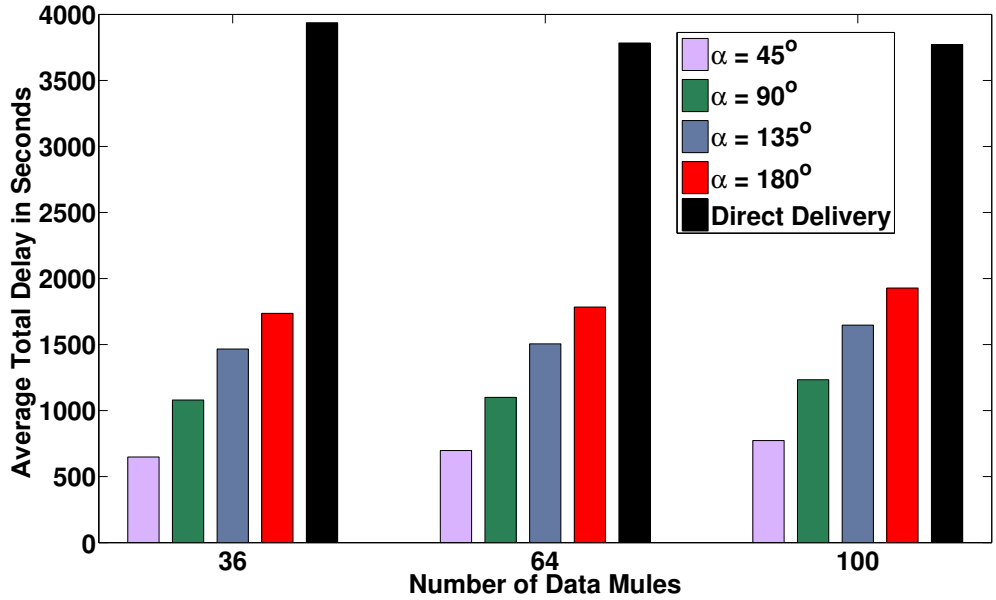

Figure 3.9. Average Total Delay with Varying α in MSS Protocol

3.6.3. MSS and Direct Delivery. Based the analysis given in Section 3.5, the DD protocol will have larger data delay when compared with MSS protocol. As shown in Figure 3.9, given a fixed number of data mules, the total delay of the DD protocol is much larger than that of the MSS protocol. Thus, though DD protocol guarantees the complete preservation of the location privacy of data source, it causes high delay, as compared to MSS protocol.

3.6.4. Discussion. Based on the above discussions, it can be concluded that the MSS protocol is α-angle anonymous and delivers significantly lower delay than DD protocol. The total delay of data packets in MSS protocol is directly dependent on the network configuration, like, the number of data mules, the size of the network, and the mobility pattern of the data mules. Thus, based on the network configuration and deployment, the delay observed in MSS protocol can still be significant. The delay in MSS protocol is composed of two components:

- The buffering delay at data source in Phase 1.

- The buffering delay at data mule in Phase 2

In this work, two modifications are proposed to the MSS protocol aimed at reducing the delay in these components. In the first approach, a Mule-Source - Saving Shortest Path (MSS-SP) Protocol is proposed, which aims at reducing the carrying delay at the data mule. Specifically, the data mules moves along the shortest path to the dropping cells after picking up data from the data source. In the second approach, a Mule-Source-Saving - Two Level (MSS-TL) Protocol is proposed, which aims at reducing the buffering delay at data source by partitioning the network into smaller blocks. The buffering delay at data source is reduced by restricting the mobility pattern of the data mules to each block. Additionally, the data is carried between the data mules along the shortest path to the dropping cells, to reduce the total delay. The data mules mobility pattern also impacts the arrival rate at different locations in the network. To this end, a study of the impact of mobility pattern of data mules on the MSS protocol is performed in Section 3.9.

3.7. MULE-SOURCE-SAVING - SHORTEST PATH (MSS-SP) PROTOCOL

The MSS-SP protocol aims at reducing the total delay of the data packets by minimizing the carrying delay of the data mules. The carrying delay at the data mule is the total time when the data packets are buffered at the data mule after it picks them up from the data source. The data mules then deliver the data packets to the sensors closest to the dropping line. The carrying delay at the data mules is primarily dependent on the size of network, the number of logical cells in the network, and privacy level specified by α-angle. As modifying these network parameters is beyond the control of the protocol, the carrying delay of the data mules is reduced by altering the mobility path of the data mules after it picks up the data packets from the data source.

3.7.1. Description of the MSS-SP Protocol. The proposed modifications only require changes to Phase 2 of the MSS protocol, i.e., using data mules to carry and unload data. When the data mule does not have a data packet buffered with it, it moves in the network based on the mobility pattern defined in Section 3.1.2. The data mules announce their arrival when they move into a cell and wait for a response from the data source. On receiving a response from the data source along with the dropping angle (β), the data mule determines the location of the dropping cells, and selects a cell uniformly at random from them. The data mule then moves directly to the selected dropping cell along the shortest path to reduce the carrying delay. The shortest path between the cell containing the data source and the selected dropping cell, is determined using Taxicab geometry (108). Thus, the shortest path length is given by the Manhattan distance between the two cells. After the data mule offloads the data to a sensors in the dropping cell, the data is routed to the base station along the shortest path as discussed in phase 3 of the MSS protocol. Therefore, the fake direction with the β angle is still leveraged to bias the attacker's observation. Thus, the MSS-SP protocol is also α-angle anonymous

3.7.2. Analysis of MSS-SP Protocol. The total delay for data delivery in MSS-SP protocol consists of two components; (1) D_{src}^{MSS-SP} - the buffering delay at the data source until a data mule picks it up, and (2) D_{mule}^{MSS-SP} - the buffering time at the data mule until it offloads the data to a sensor. In contrast, the transmission delay at static sensors in WSNs is trivial, and thus, ignorable.

3.7.2.1. Buffering delay at the data source. When the data mule is not carrying any data, its mobility model is equivalent to the model discussed in Section 3.5. Hence, the distribution of buffering delay at the data source in the MSS-SP protocol is also be given by Equation 3.2.

3.7.2.2. Expected carrying delay at the data mules. The expected carrying delay in (E_{mule}^{MSS-SP}) can be calculated as follows. Let C be the set of dropping

cells when the dropping line is at an angle β, where $|C| = f(\beta)$. t_{unit} is the transition time between two adjacent cells. Then the expected carrying delay when the source is in cell s and the dropping angle is β, is given by Equation 3.10, where $d(c, s)$ represents the Manhattan distance between the two cells.

$$E_{mule}^{MSS-SP}(\beta) = \frac{1}{f(\beta)} \sum\nolimits_{\forall c \in C} d(c, s) \times t_{unit} \tag{3.10}$$

The expected carrying delay of data mules in MSS-SP protocol (ED_{mule}^{MSS-SP}) for a given privacy level of α angle is given by Equation 3.11, where θ is the absolute angle between the direction of data source and the x-axis in the coordinate system.

$$ED_{mule}^{MSS-SP}(\alpha) = \frac{\sum_{\beta=\theta-\alpha}^{\beta=\theta+\alpha} E_{mule}^{MSS-SP}(\beta)}{2\alpha} \tag{3.11}$$

3.7.3. Experimental Study of the MSS-SP Protocol. A comprehensive set of simulations were conducted using a customized C++ based simulator to evaluate the performance of MSS-SP protocol. The simulation configuration was kept same as shown in Table 3.2 and discussed in Section 5.7. Only the data mule's mobility pattern was altered as discussed in Section 3.7.

3.7.3.1. Delay and data mules. The total delay in the MSS-SP protocol exhibited characteristics similar to that exhibited by the MSS protocol. The buffering delay at data source decreases as the number of data mules increase, as shown in Figure 3.10. Thus, the total delay decreases as well, as shown in Figure 3.11.

3.7.3.2. Delay and privacy preservation. MSS-SP protocol exhibited a similar relationship between delay and the privacy level as exhibited by the MSS protocol. Higher privacy level comes at the cost of higher delay, as shown in Figure 3.11.

3.7.3.3. Comparison of MSS and MSS-SP protocol. In MSS-SP protocols, the data mules take the shortest path to the dropping cell after picking up data from the data source. Thus, they exhibit lower carrying delay in MSS-SP protocol

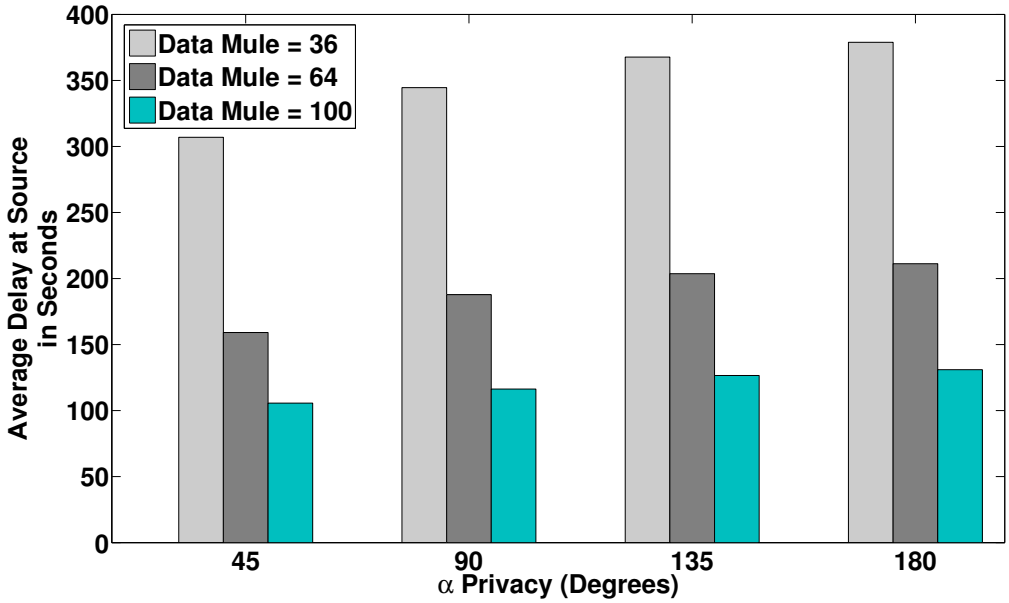

Figure 3.10. Average Buffering Delay at the Data Source in the MSS-SP Protocol

than in the MSS protocol, as shown in Figure 3.12. Hence, the total data delay in MSS-SP protocol is lower than that in the MSS protocol, as shown in Figure 3.13. Based on the above observations, it can concluded that the MSS-SP protocol delivers lower total delay than the MSS protocol.

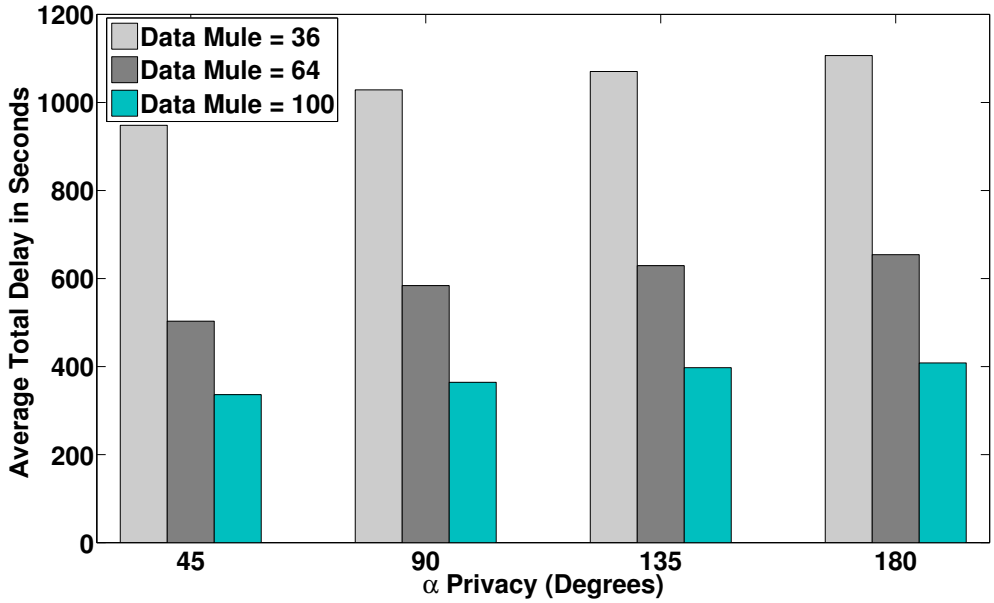

Figure 3.11. Average Total Delay at Data Mule in MSS-SP Protocol

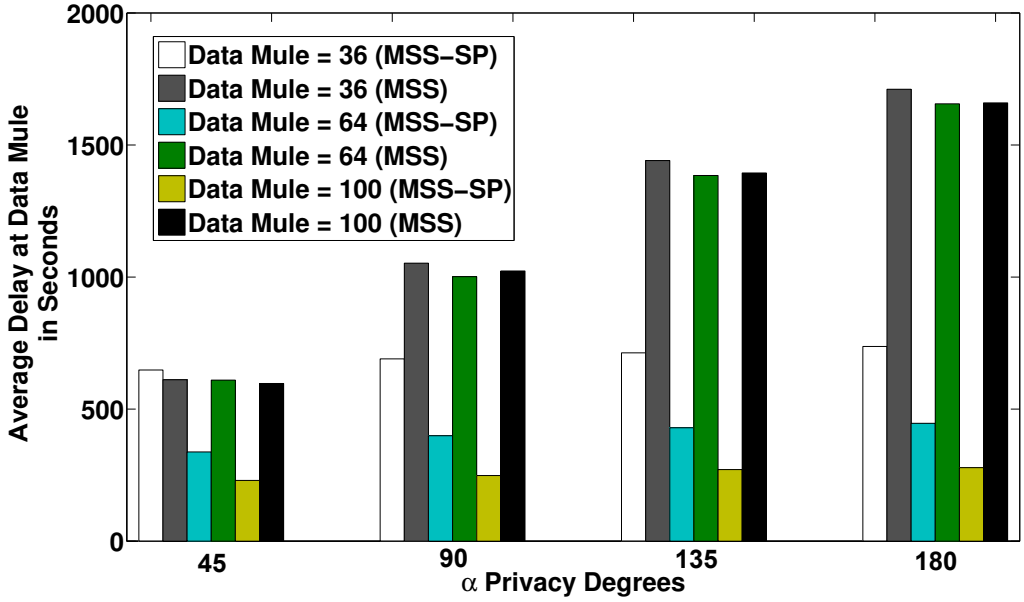

Figure 3.12. Comparison of the Average Carrying Delay at Data Mule in MSS and MSS-SP Protocol

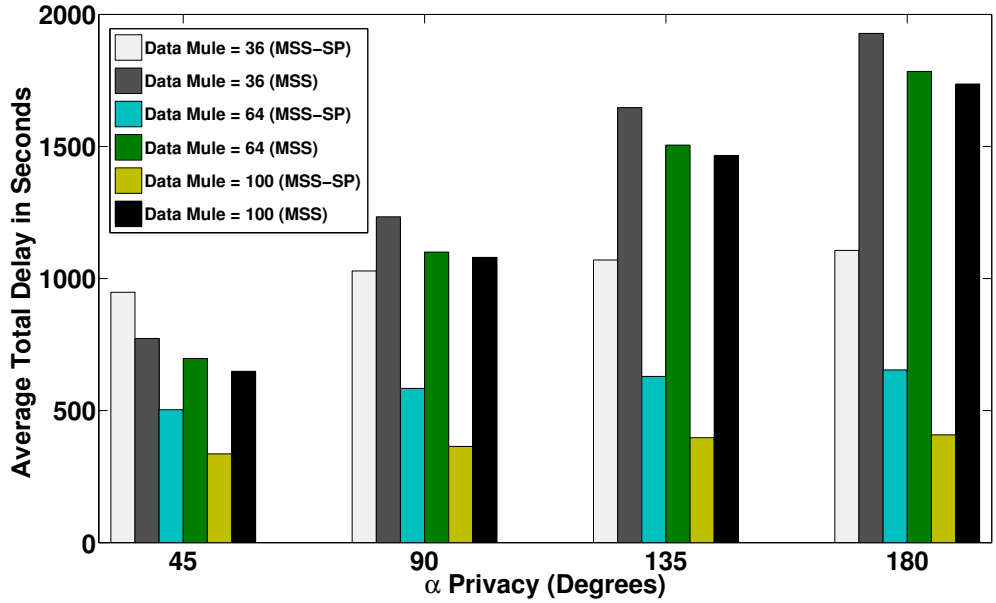

Figure 3.13. Comparison of Average Total Delay at Data Mule in MSS and MSS-SP Protocol

3.8. MULE-SOURCE-SAVING - TWO LEVEL (MSS-TL) PROTOCOL

Allowing a data mule to move throughout the sensor network may not be efficient solution considering the cost involved in their operations and the data delivery delay. To address the problem, the network was partitioned into blocks to restrict the mules's movement in the Mule-Source-Saving - Two Level (MSS-TL) protocol. Restricting the mule's movement to a block reduces the buffering delay at the data source as they can now communicate with sensors in the block more frequently. The data is then transmitted by data mules along the shortest path to the dropping cell to reduce the carrying delay of the data mule.

3.8.1. Description of the MSS-TL Protocol. The network area is divided into blocks in the MSS-TL protocol, called Large blocks (L-block). Each large block covers a set of consecutive cells and is managed by one data mule. The data mule

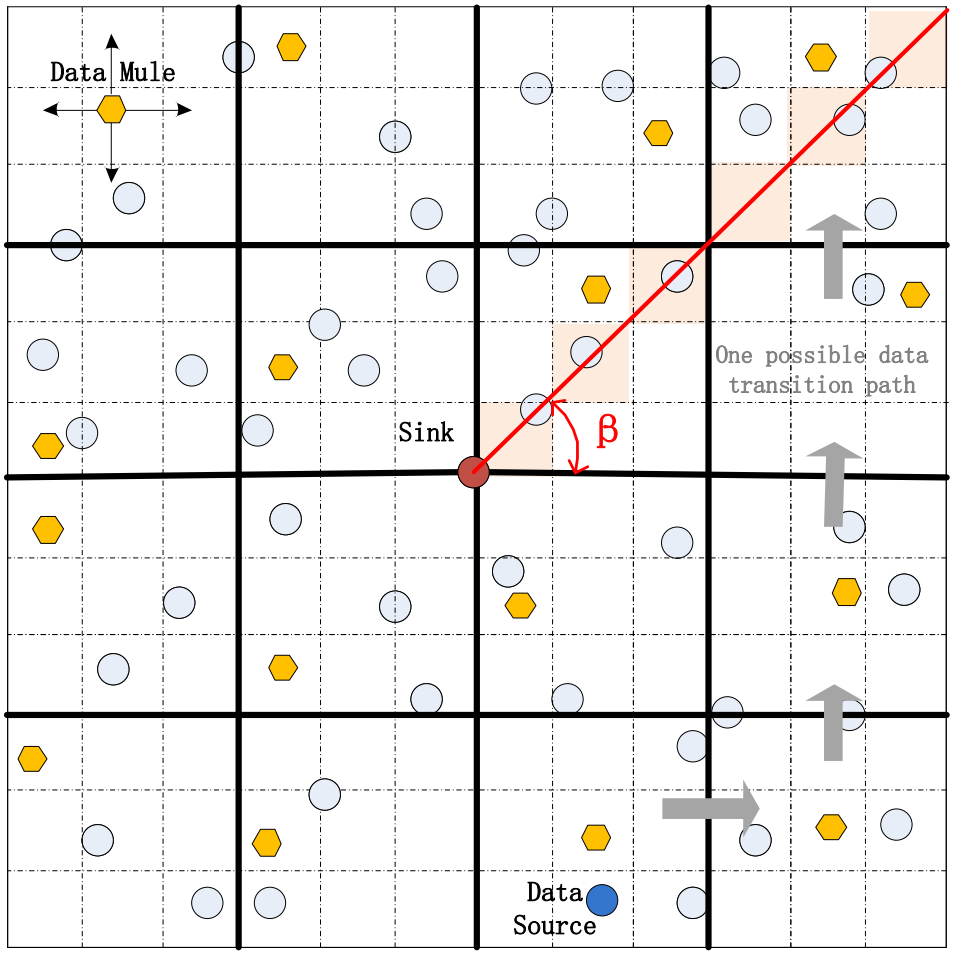

Figure 3.14. Network Scenario for MSS-TL Protocol

in a L-block moves around using the mobility model discussed in the MSS protocol. The MSS-TP protocol consists of three phases, like, the MSS protocol.

Phase 1. Choosing a fake direction of β at the source - The selection of the β angle by the data source remains the same as in Phase 1.

Phase 2. The data mule in the L-block (where the data source is located) seeks data sources by broadcasting Hello messages whenever it transits to a new cell. Once the data source is discovered, it sends both data reports and the β value to the data

mule. Unlike the MSS protocol, the data mule itself may not be able to drop data in a dropping cell as it is restricted to move only within its L-block. Thus, it uses data mules in other L-blocks to relay data towards the fake direction.

To relay the data, the data mule calculates a data transmission path. The path is represented by a sequence of consecutive L-blocks through which data should be passed. To calculate the data transmission path, the data mule first determines the set of L-blocks, which intersect with the dropping line (also called as dropping L-blocks) drawn at an angle β to the x-axis. The data mule then selects one L-block from the set of dropping L-block uniformly at random as the dropping L-block. The mule then determines the sequence of L-blocks from its current block to the dropping block along the shortest Manhattan distance.

The data mule uses the sequence to decide where in its local L-block it can drop the data. It can drop the data once it moves into any of the cells located along an edge that is shared by the current L-block and next L-block in the sequence. The sensor that receives the data will forward the data to a sensor in the neighboring cell that is covered by the next L-block. After the data mule in the new L-block finds the data, it will relay them to the next L-block following this same procedure. This relaying of data continues until the data are received by a data mule in the dropping L-block. In the dropping L-block, the data mule moves in the block until it arrives in the dropping cell. The data mule then offloads the data to the sensor closest to dropping line in the cell.

Phase 3. Routing data at sensors - The sensors on receiving the data from the data mule route the data to the base station along the shortest path.

The MSS-TL protocol remains α-angle anonymous because the fake direction is still leveraged with the β angle, biasing the attacker's observation.

3.8.2. Analysis of MSS-TL Protocol. The end-to-end delay in the MSS-TL protocol is composed of three components:

- The buffering delay at the data source and at the intermediate sensors at the edge of L-blocks
- The sum of carrying delay at the data mules from the data source L-block to the dropping L-block
- The transmission delay from the sensors in the dropping cell to the base station.

The transmission delay from the sensors in the dropping cell to the base station can be ignored as it is trivial when compared to the buffering delay at the data source and the carrying delay at the data mules. Unlike MSS protocol, the data is not only buffered at a data source sensor but also at intermediate sensors, which serve as recipients when the data is relay from its current L-block to its neighboring L-block.

The data delay in the L-block containing the data source is analyzed first. It is assumed that there are k^2 cells in each L-block and the data mule's mobility pattern has achieved stationary distribution. Thus, the probability that the data mule is present in one of the k^2 cells in its L-block is $\frac{1}{k^2}$ (113).

A random variable is defined to represent the inter-arrival time of data mule at a sensor node. Markov chains are used to model the data mule's mobility pattern in the L-block. Using the Markov chains, the average inter-arrival time of a data mule can be calculated as $k^2 \times t_{unit}$. Therefore, the expected buffering time of a data packet at the data source is $\frac{k^2 \times t_{unit}}{2}$ because data is generated at a constant rate. After the data mule picks up data, it keeps moving in the L-block until it reaches one of the k cells along the edge of the L-block, which is shared with the next L-block in the data transmission sequence. These k cells are called the destination cells in the L-block. In stationary distribution, a data mule will move into a destination cell with probability $\frac{1}{k}$. Therefore, the average inter-arrival time of the data mules in the destination cells is kt_{unit}. The inter-arrival time leads to an expected buffering time of $\frac{k \times t_{unit}}{2}$ at the data source.

Similar reasoning can be applied for analyzing the data delay in any of the L-blocks located on the data transmission path except the dropping L-block. The data mule in the source L-block drops data at a regular sensor. The data is then transmitted to a sensor in the neighboring cell located in the next L-block on the data transmission path. The transmission delay between the two sensors can be ignored as compared to the delay caused by data mule's mobility. The expected data buffering time at the recipient sensor is equal to the expected delay of the inter-arrival time of the data mule in that L-block at the sensor node, which is equal to $k^2 \times t_{unit}$. The delay of carrying data at the data mule in the current L-block can be analyzed in the same way as that for the source L-block.

The analysis of data delay at the last L-block on the data transition path is slightly different. Specifically, the data mule in the L-block can drop data at any cell located along the fake direction with β angle in that L-block. Therefore, the number of the cells which can receive data is represented as $f(\beta)$ which is not necessarily equal to k. Rather $f(\beta)$ is dependent on β angle, network area, and size of the cell.

Suppose the expected Manhattan distance from the source L-block to a L-block on the fake direction is ℓ, the total expected delay is given by Equation 3.12, where it is assumed that the random variable representing length of data transition path and data delay in any L-block are independent.

$$\begin{aligned} E[D] &= t_{unit} \times \{[\frac{k^2}{2} + \frac{k}{2}] + [(\ell - 2)(k^2 + \frac{k}{2})] + [k^2 + \frac{k^2}{2n_c}]\} \\ &= t_{unit} \times ((\ell - \frac{1}{2} + \frac{1}{2f(\beta)})k^2 + \frac{\ell - 1}{2}k) \end{aligned} \tag{3.12}$$

3.8.3. Experimental Study of the MSS-TL Protocol. A comprehensive set of simulations were conducted using a customized C++ based simulator to evaluate the performance of MSS-TL protocol. The simulation configuration was kept same as shown in Table 3.2 and discussed in Section 5.7. The size of L-blocks were

varied and the number of data mules equals the number of L-block. Each data mule is assigned to a L-block and the mule's mobility is limited to that block. The MSS-TL protocol was implemented as discussed in Section 3.8.1.

3.8.3.1. Delay and data mules. The number of data mules, the buffering delay at the sensors, and the delay of carrying data at the data mule in each L-block decreases as size of the L-block decreases. The total delay depends on both the delay at each L-block and the number of L-blocks in the network. As the number of blocks increase, more data mules are involved in carrying the data, which increases the sum of carrying delay at data mules.

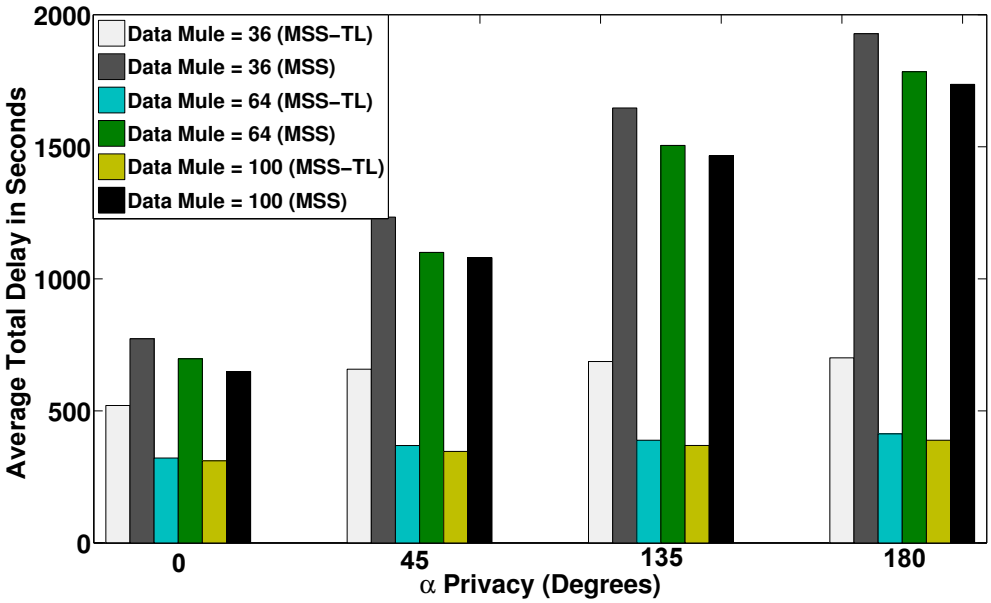

Figure 3.15. Comparison of Average Total Delay of the MSS-TL and MSS Protocol

3.8.3.2. Delay and privacy preservation. MSS-TL protocol shows similar relationship between delay and the privacy level to the MSS protocol. Higher privacy is achieved at the cost of higher delay, as shown in Figure 3.15.

3.8.3.3. Comparison of MSS and MSS-TL protocol. As shown in Figure 3.15, MSS-TL delivers significantly lower delay than the MSS protocol, which demonstrates the effectiveness of two-level based MSS in reducing data delay.

3.9. MSS PROTOCOL - RANDOM WAYPOINT MODEL (MSS-RWP)

The mobility pattern of the data mules has an impact on the spatial and temporal distribution of the data mules in the network. Thus, affecting the inter-arrival time of the data mules at different sensors in the network. In Section 3.4, the mobility pattern of the data mule was modeled as a random walk on the grid, wherein in each transition it moves with equal probability to one of the horizontally and vertically adjacent grids. At stationary distribution, the probability of the data mule being in any of the grids is equiprobable. On the other hand, Random Waypoint Model (RWP) (29) is one of the most popularly used mobility models (23). The mobility pattern of the data mules can still be modelled as a random walk on the grid, but at each transition it moves with equal probability to any other cell in the network. However at the stationary distribution, the data mules are more probable to be at the center of the network than its periphery (27). Hence, depending upon the location of the base station and data source in the network, RWP based mobility models may deliver lower delay.

The RWP based mobility model is explained below. It is assumed that the data mules move independently in the network and do not communicate with one another. The network is divided into $(\frac{L_n}{L_c} \times \frac{L_n}{L_c})$ cells of equal dimensions. At each transition, the data mule moves into the next cell, chosen uniformly at random from the whole network. During transitioning to the next cell, the path taken by data mule is given by the Manhattan distance between the source and the destination cell. On moving into a cell, the data mule stays there for t_{pause} interval before transitioning into the next cell. It is also assumed that the data mules move with a constant velocity.

3.9.1. Experimental Study of the MSS-RWP Protocol. A comprehensive set of simulations were conducted using a customized C++ based simulator to

evaluate the performance of MSS-RWP protocol. The simulation configuration was kept same as shown in Table 3.2 and discussed in Section 5.7. The mobility pattern of the data mule was altered as discussed above.

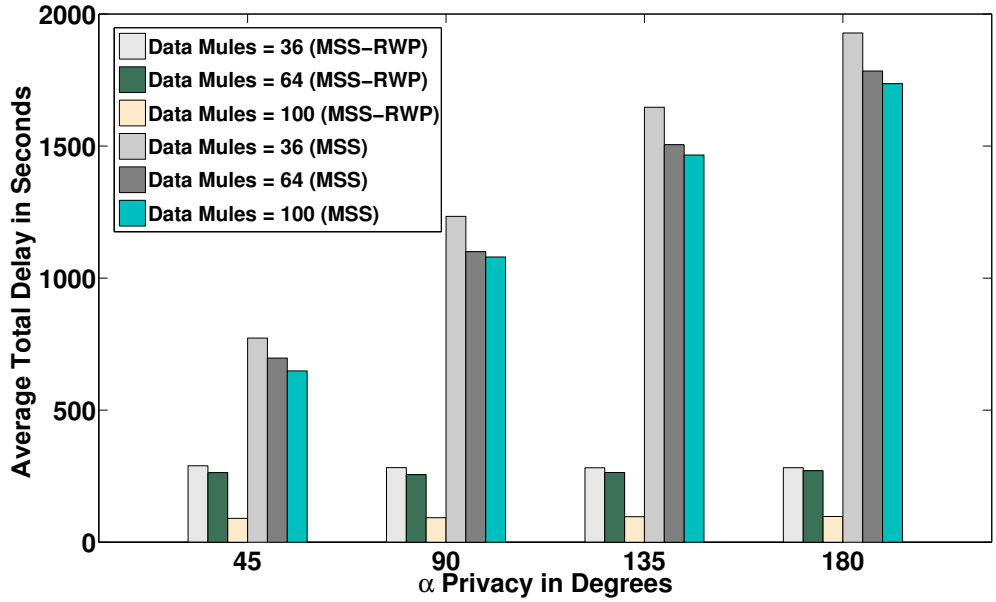

Figure 3.16. Average Buffering Delay at Data Source in MSS-SP Protocol

3.9.1.1. Comparison of MSS and MSS-RWP protocol. The MSS-RWP protocol delivers significantly lower delay than the MSS protocol This is primarily due to the characteristic of the RWP mobility model, wherein the data mules are more probable to be at the center of the network than at the periphery. Hence, after picking up data from the data source, the data mule moves towards the center of the network with a higher probability than the MSS or DD protocols. Since the BS is located at the center of the network, the data mule is able to drop the data to a dropping cell closer to the BS more frequently. This, results in significantly lower delay than the MSS protocol. Based on the above observations, it can be concluded that RWP mobility model will help reduce the delay in the MSS protocol when the BS is located around the center of the network.

3.10. RELATED WORKS

A comprehensive taxonomy of privacy preservation techniques for WSNs has been presented in (76). The work identifies two primary attack models, namely, local-eavesdropping based attack and global-eavesdropping based attack. For local-eavesdropping based attack, flooding based approach was first introduced in (96), where each sensor broadcasts data to all its neighbors. However, this technique suffers from high communication overhead for sensors. In (95), a cyclic entrapment technique is introduced to create looping routes between data source and base station aiming at trapping the attacker in loops when he backtracks along the data transmission path. In (79), each data packet is first relayed to a randomly selected intermediate sensor in the network and then is forwarded towards base station along the shortest path.

For global-eavesdropping based attack, the authors in (86) create $k-1$ fake sources in the network to anonymize the real data source. Additionally, *proxy-based* technique is proposed in (133) wherein a set of proxies are distributed in the network, which is partitioned into cells. Each cell sends traffic including both real and fake packets to its nearest proxy by following an exponential distribution. The proxies filter out some of the dummy packets they might have collected, and then send the remaining data to base station. A similar idea is brought up in (132) in which rather than relying on proxies, cluster-heads first aggregate data and then report them to base station.

The authors in (78) propose a mixing ring-based technique, in which a closed circular routing path is formed around base station. Data source first routes the data packet to a random intermediate sensor in this ring, which provides local source-location privacy preservation. Then the data is routed along the ring and will be forwarded towards base station by any ring-node with a given probability. However,

it is difficult to predetermine the size of the ring without knowing the attacker's monitoring ability.

Different from the above line of research, the proposed protocols use data mules to deliver data. Therefore, reducing energy the consumption due to communication among the sensors. As no physical data transmission path exists between the data source and the sensor to which the data mules offloads the data, the attacker cannot backtrack the transmission to locate the data source. The protocol also allows a system designer to configure the privacy level as desired by him based on the tolerable delay in data delivery, the network area and the number of data mules as well as their speed.

3.11. SUMMARY

In this chapter, a source-location privacy preservation protocol was designed for disaster prevention operations in ad hoc network environments. First, a realistic semi-global eavesdropping attack model was described. The effectiveness of the semi-global eavesdropping attack model was demonstrated by compromising the Phantom Routing protocol. Furthermore, a α-angle anonymity model was described for measuring the location privacy level of data sources in WSNs. A Mule-Source-Saving protocol was then proposed, which modifies the traditional functions of the data mules to make data sources α-angle anonymous. The protocol was then analyzed theoretically, and through a comprehensive set of experiments to study the incurred delay due to MSS protocol. Based on the observations, the shortcomings of the protocol were identified and two modifications were proposed, i.e., Mule-Saving-Source - Shortest Path (MSS-SP) protocol and Mule-Saving-Source - Two Level (MSS-TL) Protocol, to reduce the total delay in MSS protocol. Through analysis and experiments, it was shown that the proposed modifications help in reducing the total delay in data delivery. The impact of mobility mobility of the data mules on the MSS

protocol was then studied, and changes were proposed to reduce the total delay in the network.

Given the MSS protocol and its variants, the analytical models proposed in this work for each protocol can be used to determine the protocol that will give least delay for a given privacy level and network configuration. This is particularly useful as it allows the solution to be adapted and applied to a wide range of applications with varying requirements and network configuration. For example, habitat monitoring is one such application being used for preventing environmental disasters where network configuration may vary depending on the wildlife being monitored and its habitat. Additionally, the desired privacy levels may vary when WSNs are used for monitoring and tracking events in security sensitive areas for intrusion detection based on the requirements of the security infrastructure.

4. ENERGY ADAPTIVE P2P FILE SHARING APPLICATIONS FOR DISASTER PREPAREDNESS, RELIEF AND RECOVERY OPERATIONS

Disaster preparedness constitutes an important phase in disaster management. It involves creating a knowledge database of human, technical, and medical content to help people understand not only the risks they will encounter in the case of an impending disaster but also the steps they must take in order to prepare for them. For example, a number of online knowledge databases have been established by various government organizations (e.g., ready.gov (10) and the Indian Resource Network (88)) and humanitarian response organizations (e.g., disasterready.org (4)). These sites offer content in the form of illustrated document or instructional videos with sizes varying from few MBs to hundreds (4; 10). The content can be freely download by the people on their smartphones, personal digital assistants (PDAs) and tablets. This allows them to have pervasive and immediate access to content, if network connectivity is lost during a disaster.

The online databases utilize a client-server architecture for content distribution. As client-server architectures has a single point of failure, frequent requests can overburden the server and cause network congestion, especially in large-scale disaster scenarios. Peer-to-peer (P2P) networks have emerged as a simple and attractive solution for scalable, robust, rapid and efficient distribution of content (25; 102) without a dedicated infrastructure. As users also act as content sources in a P2P file sharing network, it significantly reduces the load on the server and ensures content availability even if the server fails. Furthermore, as these P2P networks are resilient and can be established without relying on a dedicated infrastructure, they can also be used in post-disaster scenarios, i.e., for sharing data among rescuew workers and victims for relief and recovery operations (74; 85). Availability of network infrastructure cannot

be guaranteed after a disaster occurs. P2P based content sharing network can be used in these instances to share information or data among rescue workers, and other interested entities in an ad hoc network environment (38; 70; 87; 90; 93; 138).

However, P2P file sharing applications are known to be bandwidth and energy intensive. As a result, they can significantly drain the battery of the wireless devices. To address this problem, a mechanism for energy based adaptation among a set of wireless clients * running P2P file sharing applications is discussed in this chapter. Specifically, the proposed mechanism allows clients to define an energy budget for downloading the file. Based on their energy budget, the clients can reduce their contributions to the network, while receiving additional service from energy rich clients. Such an adaptation will allow energy-poor clients to download the file within the restrictions of their available energy, which otherwise may not be possible. Obviously, the mechanism must provide adequate incentives to energy rich clients to participate and ensure fairness so that no client can abuse the privilege. A credit based mechanism is used to achieve these objectives. To prove the feasibility of the proposed mechanism, its implementation is discussed in the context of popular P2P file sharing protocols, like BitTorrent, eMule and Kademlia. The research work presented in this chapter has been published in (105) and a journal version is ready to be submitted.

4.1. RELATED WORKS

Several approaches have been discussed in literature for characterizing the energy consumption of the BitTorrent P2P file sharing applications. They have largely focused on analysis of energy consumption on wireless devices due to live P2P traffic (52; 64; 91). These studies provide insights into the energy consumption of P2P traffic due to various networks conditions and overhead of control messages in the protocol. The studies have shown that the overhead of control messages is minimal

*The term wireless devices and wireless clients have been used interchangeably in this work.

when compared to the data communication cost in P2P file sharing networks. While these experiments establish the feasibility of running P2P clients on wireless devices, they agree that better energy optimization techniques are required to gain widespread acceptance and usage.

To improve the energy efficiency of P2P file sharing application, the authors in (17; 19; 58) discuss a middleware based solution, wherein the task of downloading the file using the P2P protocol is delegated to a proxy. The end device only wakes up to receive the file from the proxy when the download is finished. Thus, by transferring the computational and protocol overheads to an external capable device, the wireless devices can conserve energy in downloading the file. In (66), the authors discuss different mechanisms for deployment of proxy-based solutions. A cloud-based solution is presented in (67), wherein a BitTorrent client is running remotely in the cloud and is controlled by a thin client running on the wireless device. The BitTorrent client running on a server in the cloud is responsible for downloading the file, which is transferred to the wireless device in an energy efficient way.

The authors in (75) introduces an algorithm to distinguish between wired and wireless peers. This allows them to provide wireless peers with high download rates as they can to connect to peers with high transmission rate and low network latency. However, this proposal provides preferential treatment to wireless peers, which may lead to starvation of wired peers. A green BitTorrent protocol is proposed in (28), which introduces a sleep state in the network wherein the peer can drop its TCP connection with other peers to minimize transmissions. To prevent snubbing of sleeping peers, the authors discuss an architecture and mechanism to distinguish between sleeping and dead peers.

The above mentioned solutions can be classified as achieving energy conservation through energy efficient computing (EEC). Through the optimization of system

and protocol operations as well as introduction of middleware network entities to offload network operations, energy consumption is minimized on the devices.

Alternatively, this work proposes modifications to P2P file sharing protocols, which aim at adapting protocol and system behavior based on device's available energy to maximize their chances of completing the file download. Such mechanisms are defined under the domain of Energy Adaptive Computing (EAC) (61; 62). EAC offers a distinct paradigm shift from EEC. It aims at arbitering and delivering maximum benefits to end users under the constraints of energy availability of the respective wireless devices. In the P2P domain, EAC allows the peers or devices to adapt their participation in the network, i.e. to act selfishly in their contributions, given their energy availability. On the other hand, EAC also allows the peers to derive sufficient contributions from neighboring peers, leading to accomplishment of their task in the network and achieve higher network lifetime. EAC mechanism can be used in conjunction with EEC to deliver an optimum energy-utilizing network. To the best of our knowledge, no previous approaches have been discussed in literature, aimed at empowering the P2P file sharing protocol with EAC, and applying it to disaster scenarios.

4.2. ASSUMPTIONS

The proposed mechanism exploits the following two characteristic of wireless devices so that they can adapt their energy consumption based on their energy budgets. 1) Wireless devices consume more energy when transmitting than receiving. Essentially, the signal is amplified during transmission to achieve the desired signal to noise ratio for successful decoding at the receiver. Thus, attributing for the higher energy consumption during transmission. 2) It is much more energy efficient for the wireless device to download the file at high download rates. After each transmission or reception, wireless devices continue to remain in active state for a short duration

of time, called tail time, in anticipation of another packet. Frequent occurrence of tail times can result in significant energy consumption for the wireless devices (115). At high download rates, packets are either received in the tail time or in large single bursts. This prevents tail time from occurring frequently, and reduces the average energy per transfer (115).

4.2.1. Network Scenario. Content in P2P file sharing network can be both free and paid. The humanitarian agencies have made the disaster related content freely available in their knowledge databases. Paid content is offered in collaboration with commercial content providers and can be purchased using credits. Credits can be purchased or gained through participation in the network, as explained in Section 4.3. For accounting and credit management, each user is assigned a unique id, using which it must authenticate itself before joining the network.

The example of a partially decentralized P2P file sharing application is first taken in this chapter and then the proposed mechanism is extended to other types of P2P file sharing protocols. In partially decentralized P2P file sharing applications, a super node or central server monitors and maintains information about the peers and the network. Prior to joining the network, peers must communicate with it to be able to identify the neighboring peers. Neighboring peers of a peer P are defined as those which are downloading the same file as P. Each file being downloaded is divided into smaller pieces of fixed size. Peers exchange or download pieces of mutual interest from one another. After a peer has received all the pieces of the file, it combines them to reconstruct the whole file. On completion of download, a peer may choose to stay back in the network and keep providing pieces of file to other peers as a "seed".

The access network is assumed to be an ad hoc network with a set of gateway nodes providing connectivity to the external world. In disaster scenarios, the cellular network may be partially unavailable due to high congestion or policies enforced by network operators to support high priority emergency services. The ad hoc network

constructed by devices in the region helps reduce stress on the cellular network. The gateway nodes are connected to the Internet through the cellular network. The network is also assumed to be composed of both battery constrained peers, like, smartphones and tablets, and non battery constrained peers, like, desktops. Each battery constrained peer defines an energy budget for downloading the file prior to joining the network. Energy budget of a battery constrained peer is the maximum amount of energy it wants to consume for downloading the file. It is determined while taking into consideration the available energy of the device, the energy consumption and bandwidth usage profile of other applications running on it and the maximum bandwidth the device can allocate to the application for downloading the file.

4.3. DESCRIPTION OF THE PROPOSED MECHANISM

Peers in the proposed mechanism are divided into two groups, namely Energy Sufficient (ES) group and Energy Constrained (EC) group. Peers in ES group are characterized as either those who are not battery constrained or battery constrained peers who can successfully download the file within their specified energy budget. Peers in ES group download the file only from neighboring peers in the same group as themselves. No restrictions are imposed on the bandwidth usage of peers in ES group. Upon joining the network, a battery constrained peer can determine whether it will be able to download the file successfully within its given energy budget based on the average upload and download rate of the peers in the ES group. If it cannot download the file successfully within the constraints of its energy budget, it becomes part of the EC group. The energy consumption of the battery constrained peer for a given bandwidth usage profile can be determined using the Stochastic KiBaM model (97). The model is fast and reliable with a maximum error of 2.65%. Thus, it is suitable for real-time applications.

The peers in EC group are further divided into smaller energy groups to facilitate them in downloading the file within the restrictions of their energy budget. This subdivision allows peers with similar energy budget to be grouped together and receive differentiated service based on it. The differentiated service provides peers with the necessary data rates required for downloading the file within their energy budget. Each energy group is characterized as having a unique energy consumption profile. The energy consumption profile of a group depends on the maximum upload and download rate that the peers can use in the group. The energy consumption of the wireless devices decreases as their download and upload rate increases. Thus, groups with high permissible upload and download rate can be characterized as having lower energy consumption. The energy consumption of peers in a group increases as the maximum permissible upload and download rate of the group decreases. Peers join an energy group with lower energy consumption profile than their energy budget.

Additionally, the following restrictions are imposed on the communication pattern of peers. Peers in an energy group can only download the file from neighboring peers in the same group as themselves, using the traditional P2P file sharing protocol. It is assumed that peers in each energy group upload at the maximum permissible upload rate of the group. In P2P file sharing protocols, peers prefer to associate with neighboring peers having same or higher upload rates than them. As peers in the same group operate at similar upload rates, peer discover and associate with neighboring peers faster. Thus, allowing the network to converge to a stable state faster (103).

As the download rate a peer gets in P2P file sharing protocols is proportional to the its upload rate, no peer will upload more data to a neighboring peer than what it has downloaded from it. Thus, the average degree of proportionality between the upload and download rates of a peer is 1, assuming the contributions from seeds are insignificant when compared with those of neighboring peers. Consecutively, the average download rate a peer gets in an energy group is equivalent to the maximum

upload rate of the group. The peer can also get additional download rates from seeds. However, the download rate of the peer may still not be sufficient to download the file within its specified energy budget. To address this problem, the proposed mechanism allows peers in EC group to receive additional download rates from neighboring peers in ES group. It is assumed that the peers in ES have some residual energy or are not battery constrained, and can provide additional download rates to neighboring peers in EC groups in exchange of credits.

The use of credits for purchasing additional download rates from neighboring peers in ES group has the following advantages. 1) It prevents free riding. 2) In traditional P2P file sharing applications, a peer must contribute to the neighboring peers in ES group to increase its download rate. The use of credits allows peers to gain additional download rate at low energy cost as more energy is consumed when transmitting than receiving. 3) It provides long-term fairness. Peers can accumulate credits when downloading a file by providing additional download rates to neighboring peers in other groups. The credits can be used by battery constrained peers in future to enhance their download rate and lower their energy consumption when they join the network to download a file with low energy budget. Furthermore, both non battery-constrained and battery-constrained peers, can use the gained credits to purchase content in the network. The rate of exchange of credits and bandwidth is kept high for low energy group and it decreases from low to high energy. On the other hand, low energy group peers are given more preference when allocating additional download rates as they are operating at low and strict energy budgets. Such variable exchange rate promotes fairness as it compensates for the favourable treatment. Furthermore, it dissuades peers from joining energy group with high download rates (low energy groups) owing to their higher exchange rate. Figure 4.1 shows the interactions between peers in three energy groups, namely low, medium and high, and ES group, based on the proposed mechanism.

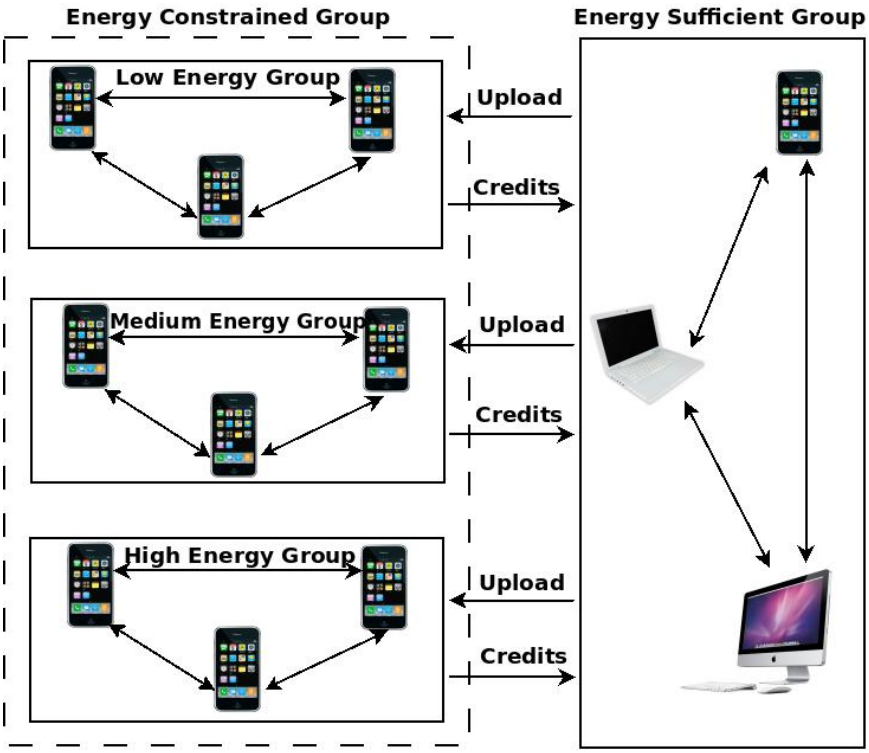

Figure 4.1. Interaction Between Peers in and Across Groups

4.4. FUNCTIONING OF THE PROPOSED MECHANISM

A peer must register with a central server before it can join the network. Each peer uses the assigned unique ID to authenticate itself whenever it joins the network. The central server keeps the track of the total credits available with each peer. Only authenticated peers are allowed to download files and are supplied with the set of neighboring peers. The proposed mechanism consists of 3 phases; 1) Initialization phase; 2) Bootstrapping phase; and 3) File download phase.

4.4.1. Initialization Phase. Peers must first communicate with the central server prior to joining the network. It is assumed the server maintains the information about peer's group membership and its ID. The peers on joining the network inform the server about their energy budget. Based on the received information, the server provides the energy constrained peers with the list of energy groups whose energy

consumption profile is lower than their energy budget, the corresponding bandwidth restrictions and the expected download rate the peers can get from neighboring peers in each energy group and the ES group. Based on the received information, the peer determines which group it should join to download the file based on available credits and informs the server. If none of the groups have their minimum energy consumption lower than the energy budget of the peer, the server may ask the peer to reevaluate its energy budget. On the other hand, energy sufficient peers on joining the network must choose if they wish to gain credits by provide additional download rates to peers in EC group. In the case, they wish to gain credits, they announce the amount of download rate they wish to provide to peers in EC group.

4.4.2. Bootstrapping Phase. On receiving a peers choice, the server provides it with two sets of neighboring peers. The first set contains the set of neighboring peers in the same group as itself and the second set contains the set of neighboring peers in ES group. Based on its group, the peer may provide additional download rates to peers in the second set or purchase additional download rate from them. For a peer in ES group, the server also informs it on how to distribute the additional download rates it wishes to offer among the neighboring peers in the second set. Essentially the server is also responsible for distributing the additional download rates being offered by peers in ES among neighboring peers in energy groups fairly. For peers in EC group, the second set consists of peers, which will be providing it with additional download rates.

4.4.3. File Download Phase. On receiving the set of peers, the peer downloads pieces of file from neighboring peers in the first set using traditional P2P file sharing protocol. If the peer chooses to purchase additional download rates, it contacts the neighboring peers in the second set for additional pieces of the file. The pieces for downloading are selected based on the rarest first strategy (34).

During the download if the peer feels continuing in the present group will not ensure the completion of download within the predefined energy budget, it may pause downloading the file in its present group. The peer can go through the initialization phases again to obtain a new group, which will allow it to download the rest of the file within the specified energy budget. On obtaining the new energy group, the peer joins it and resumes downloading the file in the new group as explained above.

4.5. CHALLENGES AND SOLUTIONS IN IMPLEMENTATION OF THE PROPOSED MECHANIMS

The proposed mechanism requires us to address the following challenges; 1) design and creation of energy groups; and 2) allocation of the additional download rates to peers in energy groups based on demands.

4.5.1. Design and Creation of Energy Groups. Let E be the set of energy groups, where $|E| = M$. For each energy group $E_i \in E$, let U_{E_i} and D_{E_i} be the maximum upload and download rate of the group. Let the set of energy groups E be ordered by their maximum upload rate in descending order, i.e. $U_{E_1} > U_{E_2} > \ldots > U_{E_M}$. Since peers can get additional download rates from seeds as well as purchase from neighboring peers in the ES group, the maximum download rate they can receive in an energy group is kept higher than the maximum upload rate of that group, i.e. $D_{E_i} > U_{E_i}$. To keep the energy consumption profile of each group unique, the maximum download rate of each group is kept as $D_{E_1} > D_{E_2} > \ldots D_{E_M}$. Ideally, the minimum energy with which a peer can download a file in an energy group is when the peer is downloading the file at maximum upload and download rate of the group. Since, $(U_{E_1} + D_{E_1}) > (U_{E_2} + D_{E_2}) > \ldots > (U_{E_M} + D_{E_M})$, the minimum energy consumption in the energy groups can be given as $E_1 < E_2 < \ldots < E_M$. Thus, group E_1 has the minimum energy consumption and requires peers to dedicate the highest amount of bandwidth for downloading the file. The minimum energy consumption

of the groups increases as the maximum permissible upload rate of the peers in the group decreases.

A survey of P2P file sharing applications was carried out to study the feasibility of creating the energy groups and restricting the bandwidth usage of peers in the group. It was found that mobile applications (e.g., Transmission P2P App for Nokia N900 and MobileMule) of some popular P2P file sharing protocols (e.g., BitTorrent (34) and eMule (5) respectively) allow users to restrict the upload and download rates that can be used by the application. Thus, establishing the feasibility of limiting the bandwidth usage of peers for creation of energy groups.

4.5.2. Distribution of Additional Download Rate to Peers. On joining ES group, each peer announces the amount of additional upload rate it wishes to offer. Similarly, on joining an energy group each peer announces the amount of additional download rate it wishes to purchase from neighboring peers in ES group. The central server must inform peers in ES group on how to distribute the additional download rates they are offering among peers in the energy group.

A cooperative game theory based model is used in this work to achieve an efficient and fair distribution of additional download rates among peers in the energy groups. The model considers the total additional download rate being offered by peers in the ES group as the utility that needs to be distributed among the requesting peers. Specifically, the model uses Shapely value based solution (100) to achieve the desired distribution.

Definition 4.5.1 *$G = (N, v)$ is a cooperative game with transferable utility, where N is a nonempty and finite set of players and $v : 2^N \rightarrow R$ is the characteristic function or worth of the coalition defined over the power set of N, given $v(\emptyset) = 0$. Every player represents a energy group and a nonempty subset $S \in 2^N$ represents the coalition formed between energy groups.*

Definition 4.5.2 *The characteristic function $v(S)$ or worth of a coalition S is defined as the weighted maximum amount of download rate the peers in the energy groups can purchase or want to purchase.*

As defined above, N is the set of players in the game, where each player is represents an energy group, i.e., $N = E = \{E_1, E_2, \ldots E_M\}$. Let O_{ES} be the total additional download rate being offered by peers in the ES group. v is a real valued function, which represents the worth of a coalition. The worth of a coalition is a real valued characteristic function defined over the power set 2^M, the set of all possible coalitions of energy groups. A coalition is a subset of energy groups $S \in 2^M$, which collaborate to maximize the download rate they can receive. v represents the the total additional download rate peers belonging to energy groups in coalition S can or wish to purchase, and is given by Equation 4.1.

$$v(S) = \left(\sum\nolimits_{E_i \in S} w_{E_i} \times min(R_{E_i}, x_{E_i} \times BW_{E_i})\right) - c(S) \text{ and } v(\emptyset) = 0 \tag{4.1}$$

x_{E_i} is the total number of peers and R_{E_i} is the total download rate that the peers in group E_i wish to purchase. w_i is the weight assigned to group E_i, such that $w_{E_1} > w_{E_2} > \ldots > w_{E_M}$. This is done to ensure that peers with low energy budget get higher preference when allocating download rates to them. BW_{E_i} is the total average download rate required by peers in group E_i to make their download rate equal to maximum download rate (D_{E_i}) of the group E_i, as given by Equation 4.2.

$$BW_{E_i} = \begin{cases} 0 & \text{if } D_{E_i} \leq \bar{\delta}_{E_i} \\ (D_{E_i} - \bar{\delta}_{E_i}) & \text{if } D_{E_i} > \bar{\delta}_{E_i} \end{cases} \tag{4.2}$$

In Equation 4.2, δ_{E_i} represents the average download rate of the peers in group E_i. The equation computes the additional download rate required to make the total

average download rate of peers in group E_i equal to the maximum download rate of the group. $c(S)$ is used to discourage peers from operating at a lower upload rate in the group than their maximum upload rate. It ensures that peer do not reduce the contributions within their own group and request more data rates from peers in ES group. However, the download rate of a peer also depends on network conditions, like, the number of seeds, available bandwidth, and the number of peers in each group. The function allows energy groups to balance their deficits with the surplus rates available in the other groups. It is calculated as shown in Equation 4.3.

$$c(S) = \sum\nolimits_{E_i \in S} (U_{E_i} - \bar{\delta}_{E_i}) \times x_{E_i} \tag{4.3}$$

The Shapely value based solution distributes the total additional download rate or the utility among the players fairly. To achieve the desired distribution, the solution considers the average marginal contribution of the group when distributing the total additional download rate among them. The average marginal contribution of an energy group in a coalition is measured as the average demand of additional download rate the group can demand in all possible combinations with which coalitions can be created. Let $\phi(E_i)$ be the function representing the distribution of the utility among the energy groups. $\phi(E_i)$ is given by Shapely value and is calculated as shown in Equation 4.4. Given the Shapely value $\phi(E_i)$ of the group, the total available download rate O_{ES} is distributed among the groups in the ratio of their Shapely values.

$$\phi(E_i) = \sum\nolimits_{s=1}^{M} \frac{|S|!(n-1-|S|!)}{n!} \sum\nolimits_{S \subseteq N, E_i \notin S} (v(S \cup \{E_i\}) - v(S)) \tag{4.4}$$

However, it still needs to be shown that the allocated additional download rate to groups using Shapely value will result in a grand coalition of all energy groups.

No group will have an incentive to deviate from the grand coalition and form smaller coalitions. Thus, prove that the solution is stable. This is achieved by proving that the solution obtained using Shapely values always lies in the core of the game. The core of a game is a set of feasible payoff profiles or the distribution of the utility among the players such that no player or coalition gets smaller utility than its worth. To prove this, the proposed model is first shown to be a convex game. The Shapely value of a convex game always lies in the core.

Theorem 4.5.3 *The game $G = (N, v)$ is a convex game.*

Proof A game G = (N, V) is convex if $v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$, $\forall S, T \in N$. The $\cap$ operator represents the event when two groups are combined to form a single group. In such a case, the maximum upload and download rate of the new group will be equal to maximum upload and download rate of group in the coalition with minimum energy consumption. This ensures that peers in lower energy group are able to complete the download. Thus, the maximum permissible upload and download rate of the peers in all the groups except the minimum energy group is increased. As a result, peers will get higher download rates from neighboring peers in their group than their old energy group. Hence, their demands will be reduced and $v(S)+v(T) > v(S \cap T)$. If two groups collaborate, they can reduce the impact of $c(S)$ function. Deficits due to lower download rates of one group can be compensated by the higher download rates available in the other group, thus, $v(S \cup T) > v(S) + v(T)$. In the rest of the cases, the $v(S \cup T)$, the two coalitions will either have no deficits or both will have deficits. In these two cases, the utility of the coalition will be the linear sum of their individual utilities as no coalition can benefit from one another, i.e., $v(S \cup T) = v(S) + v(T)$. Thus, $v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$.

Corollary 4.5.4 *The game $G = (N, V)$ has a non-empty core and the Shapely value for the game lies in the core.*

Proof The game $G = (N; V)$ is a convex game. Thus, G has an non-empty core as the Shapely value of a convex game always belongs to the core (100).

4.6. IMPLEMENTATION AND SIMULATION OF THE PROPOSED MECHANISM

To evaluate the proposed mechanism, its implementation in the context of BitTorrent protocol (34) is discussed first, and then analyzed using a comprehensive set of simulations. Having established the feasibility of the proposed mechanism, insights on how to extend it to other popular P2P file sharing applications (e.g., eMule and Kademlia) are provided.

4.6.1. Modifications to the BitTorrent Protocol. The BitTorrent (34) protocol employs a centralized tracker mechanism for peer discovery. A peer (P) intending to download a file (F), downloads a ".torrent" file from the host web-server. The ".torrent" file contains information about the tracker and the file F. The tracker maintains a list of peers currently downloading the file F. Upon contacting the tracker, P receives a list of neighboring peers it may contact for exchanging chunks of the file F. The peer P individually contacts the neighboring peers and requests them for chunks of file F. On mutual consent, the peers exchange chunks of the file F of mutual interest.

The tracker, in the BitTorrent protocol, assumes the function of central server. The tracker maintains the group membership information of each peer and their choice of action, i.e. whether they wants to provide additional download rate or purchase additional download rate, along with their identity. Each peer is provided with two sets of neighboring peers. The first set consists of neighboring peers in the same energy group as the peer. The peers can download the file from neighboring peers in the first set using the BitTorrent protocol. For peers in the energy groups, the second set contains neighboring peers in ES from which they can purchase additional

download rates. For peers in the ES group, the second set contains neighboring peers in energy groups to which they should provide additional download rates. The tracker runs the allocation algorithm for allocating additional download rates and informs the peers in ES group how to distribute it among the neighboring peers in the second set.

4.6.2. Simulation and Results. Simulations were carried out to evaluate the performance of the proposed mechanism. Simulations were conducted in ns2 (7) using a BitTorrent module (42). The EC group was assumed to have 3 energy groups, $E = \{low, medium, high\}$, representing peers with low, medium and high energy budget respectively. The maximum upload rate of low, medium and high energy group was considered as 192, 128 and 64 *kbps* respectively and the maximum download rate as 256, 192 and 128, *kbps* respectively. The upload rate of the peers in ES group was considered as 256 kbps. The size of the file was assumed to be 700 MB. Based on analysis of traces from live BitTorrent P2P network (136), the peer arrival rate for each energy group was kept as 0.0045 $peers/sec$. It was also assumed that a peer becomes a seed with probability 0.5 and all peers in a group operate at the maximum upload rate of the group. The weights assigned to each group was kept as $w_{low} : w_{medium} : w_{high} = 3 : 2 : 1$ and the simulation time was kept constant as 4.5 hrs. Each data point represents the average value computed over 10 iterations of each simulation.

In the first set of experiments, the average download rate of peers in each group with varying availability of additional download rate from neighboring peers in ES group was observed. The additional download rate offered by peers in ES group was varied as the ratio of their maximum upload rate, as shown in Figure 4.2. The allocation algorithm gives higher preference to low energy group and the peers in the group achieve maximum download rate faster. The rate at which peers in an energy group approach the maximum download rate of the group is dependent on the weights assigned to each group. When peers in low energy group achieve the maximum

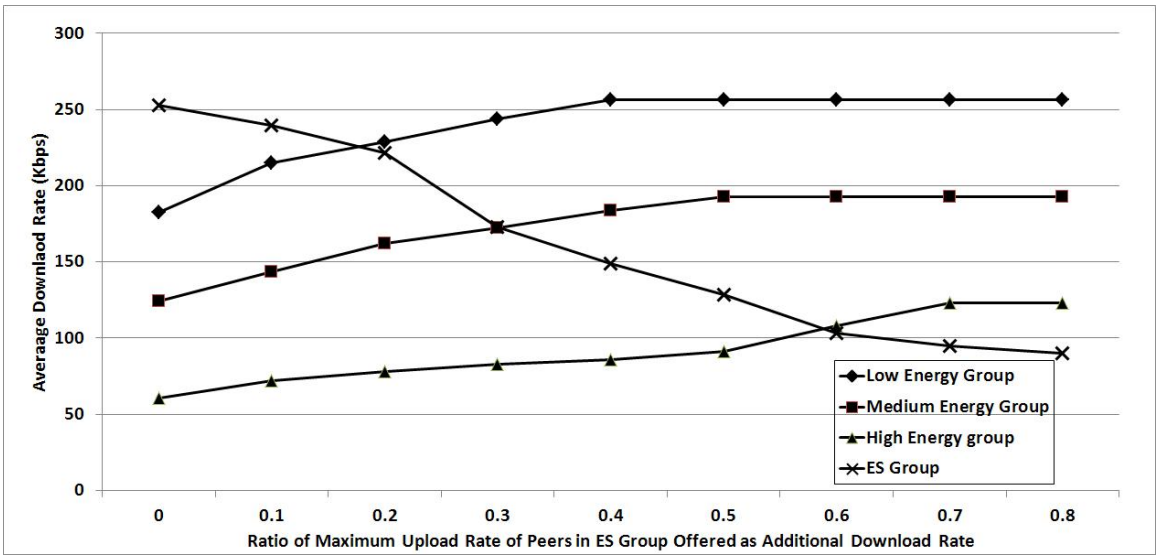

Figure 4.2. Average Download Rate of Peers with Varying Additional Download Rate

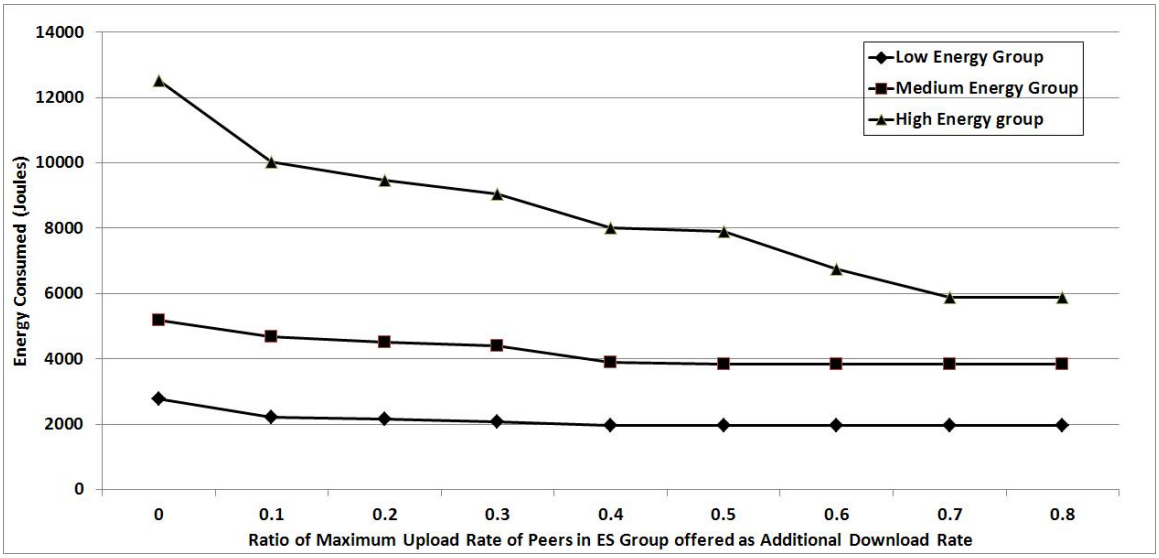

Figure 4.3. Average Energy Consumption of Peers with Varying Additional Download Rate

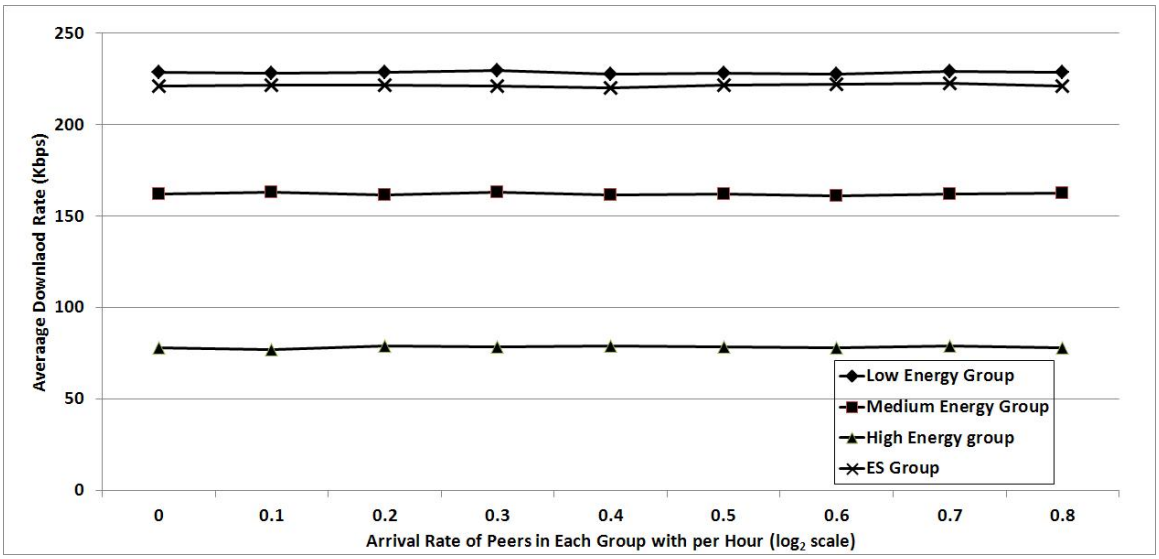

Figure 4.4. Average Download Rate of Peers with Varying Arrival Rate

download rate of 256 kbps, the peers in medium and high energy group are allocated the remaining additional download rate in the ratio of their demands. Hence, the rate at which peers in medium and high energy group approach the maximum download rate increases. As the peers in the ES group provide more additional download rate, they reduce their contributions to neighboring peers in their group. Since, BitTorrent works on tit for tat principle, peers in ES group receive less download rates from neighboring peers in their group. They rely more on the the download rate being provided by the seeds. Thus, their download rate reduces. Furthermore, using the application traces collected in the first experiment, traffic was generated to (from) Google Nexus One smartphone and the energy consumption for downloading the file was measured, as shown in Figure 4.3. The results establish the claim that peers in low energy groups are able to download the file using less energy than peers in high energy group. The energy consumption increases from medium to high energy group. Thus, establishing the feasibility of using energy groups to let peers download the file within their specified energy budgets.

In the second set of experiments, a study of the effect of arrival rate of peers on the average download rate of each energy group was carried out, as shown in Figure 4.4. The fraction of maximum upload rate offered by peers in ES group as additional download rate was kept constant at 0.5. The download rate a peer gets in a BitTorrent system is independent of the peer arrival rate (103). The implementation of the proposed mechanism does not require changes to the inherent characteristics of the BitTorrent protocol. It only requires us to put constraints on their bandwidth usage. Thus, the proposed modifications inherit the properties of the BitTorrent protocol. The BitTorrent protocol along with the additional upload rate allocation model ensures that the system remains robust to the varying peer arrival rates.

4.6.3. Other P2P File Sharing Protocols. In this section, the implementation of the proposed framework in other popular P2P file-sharing protocols, like, eMule (5) and BitTorrent Distributed Hash Table (DHT) (80) based on Kademlia (83) is discussed. eMule protocol, is a partially centralized P2P protocol and operates similar to BitTorrent. Peers are serviced based on their ranks, which is a measure of their contributions in the network. An eMule client uses a preemptive queue based system to service neighboring peers based on their ranks. In the proposed mechanism, multiple queues can be used to queue neighboring peers from different energy groups and provide service to them based on their ranks. The segregation allows allocation of additional download rates to neighboring peers in different energy groups through the choice of appropriate queue scheduling algorithm. Further, to assign ranks to peers, eMule used a credit based mechanism to determine the contributions of a peer, which can be easily extended to incorporate the required features of the proposed mechanism. BitTorrent DHT is distributed tracker system where each node acts as a peer and tracker. The lookup for neighboring peers is performed using the DHT protocol, while the rest of the functionalities are similar to BitTorrent. Hence, in the proposed mechanism, the DHT ring maintains a separate list for peers in each

energy group, which are downloading the same file. Thus, the DHT ring can be used to obtain the list of neighboring peers from different energy groups while the rest of the modifications to BitTorrent DHT remains the same as partially decentralized BitTorrent.

4.7. SUMMARY

In this chapter, an energy aware P2P file sharing framework was proposed for efficient distribution of disaster preparedness content among users facing a disaster. The proposed framework allows the users to define an energy budget for downloading the file. Based on the defined energy budget, the framework effectively allows low energy clients to "borrow" energy from clients with sufficient energy. The energy based adaptation increases the probability of users downloading the file within the restriction of their energy budget and maximizes the lifetime of their devices. A detailed description of challenges and associated solution in the implementation of the proposed framework in the context of BitTorrent protocol was provided. An exhaustive set of simulations were used to analyze the proposed framework, and establish its feasibility. Having established the feasibility of energy-based adaptation in P2P file sharing applications, the proposed framework can be used as a solution for adapting P2P applications for mobile devices.

5. E-DARWIN: ENERGY AWARE DISASTER RELIEF AND RECOVERY NETWORK

In the event of a disaster, immediate influx of health, environmental and infrastructure data is essential for assessing the conditions in the affected region, coordinating rescue efforts, and addressing the needs of the affected community. However, the availability of network infrastructure is not guaranteed in post-disaster scenarios. As the affected regions may not always be accessible, a temporary ad hoc network must be built utilizing all available resources for aiding in relief and recovery operations.

Smartphones and tablets, due to their proliferation, can be assumed to be widely available among the people in the affected region. These wireless devices can act as valuable resources in rapid establishment of a disaster recovery network for data collection and analysis followed by important decision making. For example, after the Haiti earthquake in 2010, there were approximately 2.8 million active mobile subscribers out of 10 million inhabitants contributing data for tracking the movement of population in the affected region (26). A network infrastructure based on such a large number of pre-existing devices could address the coverage and connectivity needs of the disaster recovery networks.

This work proposes a novel architecture called *Energy Aware Disaster Recovery Network Using WiFi Tethering* (E-DARWIN) for creating the required network infrastructure in the disaster affected region using wireless devices. The proposed architecture uses WiFi Tethering technology, ubiquitously available on wireless devices, to create the wireless ad hoc network. The devices join the network autonomously and participate in forwarding data to a remote emergency response command center with minimum delay. Additionally, wireless devices are increasingly being equipped with multi-modal sensors, such as temperature, accelerometer, pressure, GPS, microphone

and camera and hence, can act as rich sources of sensory information in disaster scenarios (46; 114). To utilize these rich capabilities, a distributed coalition formation game is designed and implemented, which aims at distributing the data requirements of the network among wireless devices based on their capabilities, available energy and network participation for higher energy efficiency. The performance of the proposed solution was then evaluated using an implemented prototype on Android platform and large-scale simulations. The research work presented in this chapter has been published in (104) and a journal version is ready to be submitted.

5.1. POST-DISASTER NETWORK SCENARIO

The network infrastructure is assumed to have been destroyed in the disaster affected region. It needs to be set up primarily using local resources as the affected region may not always be accessible. The central component of a disaster recovery network is a remote Emergency Command Center (ECC), which becomes operational soon after the occurrence of the disaster. The ECC receives data from the affected region, analyzes them and coordinates relief and rescue efforts accordingly, as shown in Figure 5.1.

Connectivity to the ECC from the affected region can be achieved by using WiFi access points (APs) in conjunction with satellite gateways. Each satellite gateway is composed of a very-small-aperture terminal (VSAT) dish antenna and a satellite modem, which can be easily assembled and disassembled for portability (1; 65). The satellite gateways with support of data rates up to 50 Mbps (9) can be assumed to satisfy the backhaul capacity requirement of the network. The APs can be deployed at various locations in the affected region by emergency crews. It is assumed that when road connectivity is available, the WiFi APs with satellite gateways are carried to the affected region by emergency vehicles. If no road connectivity is available to

the affected region, then these equipment can be carried or airdropped in the affected region by emergency crews (65).

The network is primarily composed of wireless devices available with people in the affected region. Based on their density and communication range, the devices are organized into multiple connected components, wherein devices in a component capture, store and forward data to a remote Emergency Command Center (ECC). It was assumed that the WiFi APs are deployed in each component. The wireless devices have high storage capacity, ranging from a few gigabytes to 512 GB, and can be assumed to have enough storage capacity available to store data. It was also assumed that all devices have the prototype of the proposed solution installed. After a disaster, it has been commonly observed that people wait for evacuation. They may go to the nearest relief center and return home. After returning from the relief center, they stay within their neighborhood (125). In this work, it has been assumed that the initial exodus of people to the relief center has already occurred, and there is no significant mobility of the devices.

5.2. RELATED WORKS

Using sensors for constructing disaster recovery network and capturing data has been extensively explored in literature (22; 45; 50). However, making use of wireless devices, like, smartphones and tablets, widely available among people in the affected region is a more practical approach (46; 114). Even though the solutions discussed in literature utilize the ad hoc mode of operation in wireless devices to construct disaster recovery network (32; 47; 81; 107), implementing it on current wireless devices requires modifications of the transceiver driver firmware, root access to kernel or jailbroken devices (2; 121). Evidently, this is undesirable, as these solutions are not seamlessly supported across all devices and are illegal in many countries. This

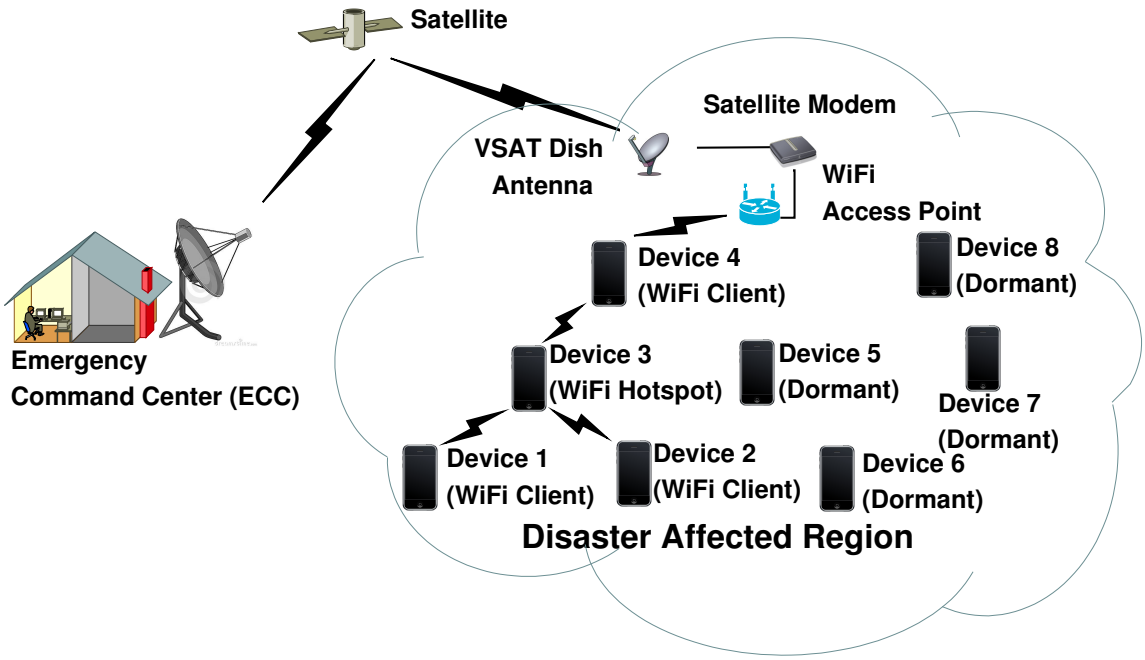

Figure 5.1. Network Architecture of E-DARWIN

motivated us to explore the use of technologies available on the wireless devices, i.e. WiFi Direct and WiFi Tethering, for constructing the disaster recovery network.

WiFi Direct or WiFi Peer-to-Peer (P2P) is a standard, which allows devices to communicate with one another without the need of a wireless access point (14). Wi-Fi Direct works by embedding a limited wireless access point in the devices, and using the Wi-Fi Protected Setup system to negotiate a link. Setup generally consists of bringing two Wi-Fi Direct devices together and then triggering a "pairing" between them, using a button on the devices or Near Field Communication (NFC). The feasibility of using WiFi Direct to create an ad hoc network in disaster scenarios has been shown in (47); however, the technology has its limitations. When using WiFi Direct technology, a device is connected to another device acting as a limited AP. However, not all devices have the capability to connect to multiple hotspots or APs at the same time, i.e., can pair with multiple devices simultaneously. This limits the usability of the technology in supporting a robust and connected network. Furthermore, the technology is not

available on all devices and is more energy-exhaustive than WiFi Tethering (123). Therefore, this work uses WiFi tethering technology to construct the ad hoc network.

WiFi Tethering technology is ubiquitously supported by all major device manufacturers and is available across all major OS platforms, such as iOS 4.3+, Android 2.2+ and Windows 7.5+. It allows a device to act as a WiFi hotspot, while allowing other devices in its communication range to connect to it. Using the hotspot, the connected devices can communicate with one another as well as use its data connection to access external networks. A characterization of the energy consumption of the device when using WiFi Tethering technology has been presented in (53). Based on the observed behavior, the authors proposed an energy efficient mechanism to further improve the usability of WiFi Tethering technology on mobile devices. Furthermore, WiFi Tethering has been used in the context of opportunistic networks (122) and creation of a computing cluster grid (30), wherein the devices utilize it to communicate with other devices in their communication range. Additionally, the authors in (123) and (129) adopt a role based approach to construct an ad hoc and mobile ad hoc network respectively using WiFi Tethering. The authors in (129), propose the use of network virtualization to let devices simultaneously assume the roles of hotspot and client. However, virtualization of the network interfaces is not supported in current mobile devices. Furthermore, the authors in (123) and (129), do not address issues related to network formation, routing and maintenance. To the best of our knowledge, no other work in literature investigates the construction of an ad hoc network for disaster scenarios using WiFi Tethering and routing of data in it.

5.3. OVERVIEW OF THE PROPOSED SOLUTION

The E-DARWIN architecture follows a selective-connected networking (15) paradigm, wherein the extent of connectivity of the devices in the network is restricted by the role taken by their neighbors, i.e. devices in their communication range. When

using WiFi Tethering, a device can either act as a WiFi hotspot or client. Only when a device acts as a hotspot, its neighbors can communicate with one another through it. The proposed architecture allows the devices behave autonomously, wherein they randomly take up the role of WiFi hotspot. To facilitate forwarding of data in the network, the devices autonomously discover their neighbors and synchronize with them so that they can schedule themselves to act as WiFi client only when their neighbors act as hotspot. When connected to a hotspot, the hotspot and its clients select one of them as the relay device and offload data to it. The relay device stores the data until one of its neighbors acts as a hotspot wherein the data is offloaded to another device selected as the relay device. The relay devices store and forward the data, until they can deliver it to a WiFi AP. To deliver data to the ECC with minimum delay, the devices select a relay device at each step, which can forward data to the WiFi AP earliest based on the schedule of the devices. Furthermore, given the uneven distribution of the devices in the affected region, significant redundancy may exist in the data captured by nearby devices. A unique aspect of this work is to design a distributed coalition formation game, which allows devices to cooperate and form coalitions to minimize the redundant data being captured in the network and distribute the data capturing task among themselves based on their available energy and network participation for higher network lifetime.

5.4. E-DARWIN - NETWORK ARCHITECTURE

The E-DARWIN network architecture is composed of ECC, WiFi APs with satellite gateway and wireless devices as shown in Figure 5.1. The proposed architecture seamlessly integrates wireless devices in a robust ad hoc network irrespective of the fact whether they can or cannot connect to multiple hotspots. To make the architecture flexible and scalable, it lets the devices behave autonomously, wherein

they discover and synchronize with one another and forward data with minimum delay. Each device in the network can be in only one of the three states at any given instant of time, namely WiFi hotspot, WiFi client and dormant, as described below.

Dormant - A device by default stays in the dormant state to conserve energy, until it is scheduled to act either as a WiFi hotspot or a client. In this state, all the network interfaces of the device are disabled and the entire device is in a low power mode with the CPU sleeping. Note that this state is different from the dormant state in the medium access layer, wherein an active data connection exists and only the wireless interface is periodically switched off to conserve energy.

WiFi Hotspot - Each device in the network becomes a WiFi hotspot consecutively after a random time interval. In this state, the device is responsible for selecting the relay node and facilitates the offloading of data to it. Devices act as hotspot for a predetermined time interval and then enter the dormant or client state. If the time when a device is scheduled to be in hotspot state overlaps with another state, the device will always enter or continue to be in the hotspot state, as its role as a hotspot is essential for facilitating communication between its neighbors and offloading the data.

WiFi Client - Using the mechanism discussed in Section 5.5, each device in the network discovers its neighbors and synchronizes with their schedules to act as a WiFi client only when one of its neighbors acts as a hotspot. In this state, the device periodically scans the wireless channel for advertisements from the hotspot and associates with it. On successful association, the client devices connected to the hotspot participate in the selection of the relay device and offloading of data to it. A device stays in the client state as long as the hotspot is active or until it is scheduled to act as a hotspot, whichever is earlier, and accordingly enter the dormant or hotspot state.

A device can capture data while it is in the hotspot or client state. If a device is

scheduled to capture data in the dormant state, it will wake up to do so and become dormant again.

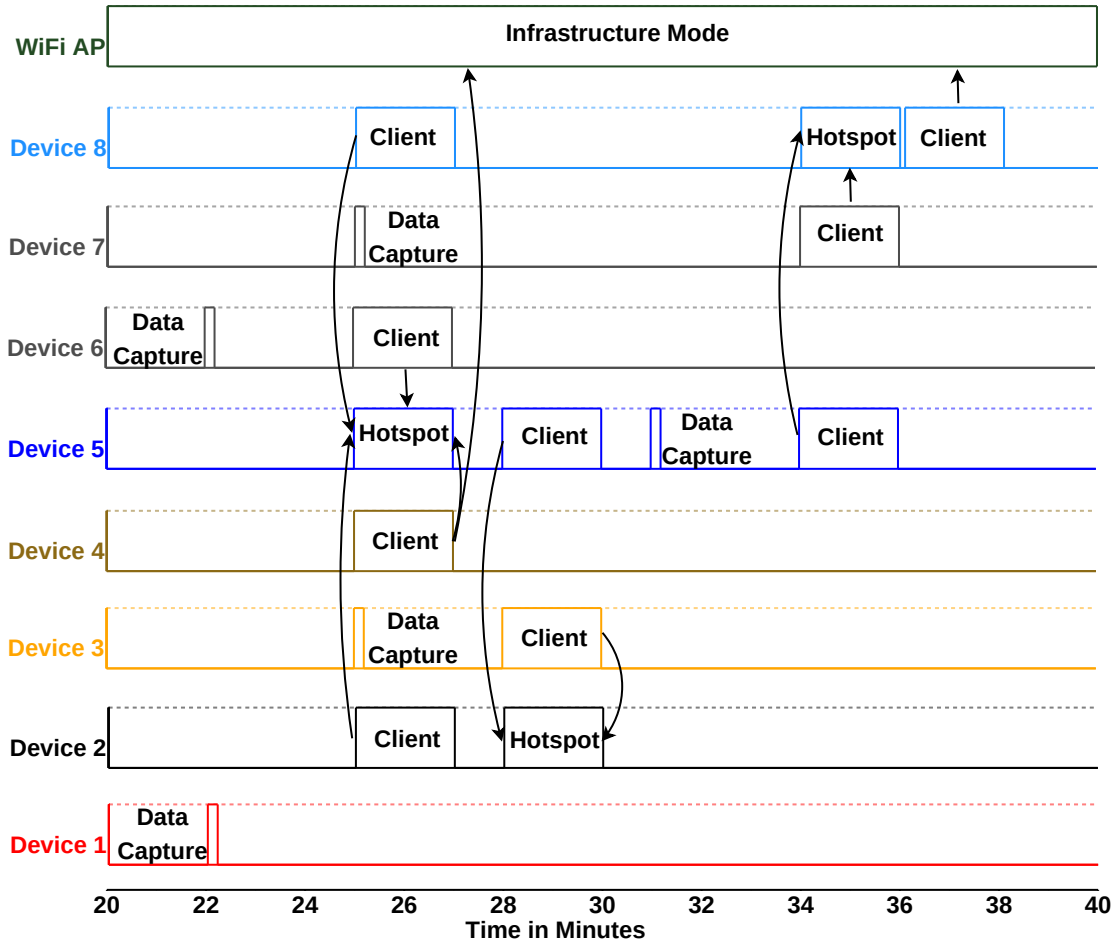

Figure 5.2. Functioning of E-DARWIN

5.4.1. Functioning of E-DARWIN. Figure 5.2 depicts the functioning of the E-DARWIN architecture corresponding to the network deployment shown in Figure 5.1. Each Device i, $\forall i \in \{1, 2, \ldots, 8\}$ in Figure 5.2 is denoted as D_i. It is assumed that the devices have synchronized with their neighbors and have scheduled themselves to act as WiFi client whenever one of their neighbors acts as WiFi hotspot. All devices connected to the hotspot interact directly with it, and indirectly with one another through the hotspot. Let us assume D_3 is scheduled to act as a WiFi hotspot at $t = 22$ mins. Devices D_1, D_2 and D_4, which are in the communication range of D_3 wake up as WiFi client at $t = 22$ mins and associate with it. Assuming D_4 is

connected to both D_3 and WiFi AP simultaneously as it is in the communication range of both the devices, and can connect to multiple hotspots at the same time. Hence, D_4 can forward data received from D_1, D_2 and D_3 to the WiFi AP as soon as it receives them, and is selected as the relay node. If D_1, D_2 and D_3 have any stored data, they forward the data to D_4. Assuming, the devices were previously configured to act as a WiFi hotspot for 2 mins, the devices stay in their respective state till $t = 24$ mins, and then become dormant.

Similarly at $t = 25$ mins, D_2, D_4, D_5, D_6 and D_8 are scheduled to wake up from the dormant state, with D_5 acting as WiFi hotspot and the rest as WiFi clients. Again, since D_4 can connect to both WiFi AP and hotspot (D_5) at the same time, it is chosen as the relay node. Now D_8 is also in the communication range of the WiFi AP and hotspot. However, assuming the capabilities of D_8 is limited, i.e., it can connect to only one hotspot at a time, it will only associate with D_5 at $t = 25$ mins. D_8 can collect data from other devices and when D_5 ceases to be a hotspot at $t = 27mins$, it can continue as WiFi client and connect with the WiFi AP to deliver the data. Thus, D_4, which can deliver data to the WiFi AP as soon as it receives them, is chosen as relay node and not D_8.

At the next time instant, i.e., $t = 28$ mins, D_2, D_3 and D_5 wake up with D_2 acting as a WiFi hotspot. Assuming D_5, which is also in the communication range of D_8, knows the schedule of D_8. Hence, it can advertise that since D_8 is supposed to act as WiFi hotspot at $t = 34$ mins, data can be forwarded to the WiFi AP through D_8 at $t = 36$ mins. Since, no other device has the opportunity to forward the data to the WiFi AP before D_5, the latter is selected as the relay device. If devices D_2 and D_3 have any captured data, they forward them to D_5. At $t = 34$ mins, when D_5, D_7 and D_8 wake up with D_8 acting as a WiFi hotspot. D_5 which had previously received data from other devices, forwards the stored data to D_8, which is now the relay node. At the end of the hotspot state, D_8 delivers the data to the WiFi AP by

changing its state to WiFi client and associating with the WiFi AP. D_3, D_5 and D_6 wake up from dormant state whenever they are scheduled to capture data and then go back to the dormant state.

5.5. NETWORK INITIALIZATION AND DATA FORWARDING

In the event of a disaster, the E-DARWIN application is activated. On activation, the devices discover other devices in their communication range and learn their schedule. After initialization, the devices forward data to the WiFi AP using relay devices such that they are delivered with minimum delay.

5.5.1. Activation of E-DARWIN Application. The first step towards network initialization is the activation of the E-DARWIn application on the devices. The application can be manually activated by the user. However, users may not always be available as they may be incapacitated or confused. As the network infrastructure is destroyed, no external stimulus can be given to the wireless devices for activating the application. Additionally, the loss of network itself is not an accurate indication of emergency as signal loss can occur due to signal attenuation inside buildings.

On the other hand, sirens are synonymous with emergencies. They are being used in mass notification systems, indoor alarm systems, and emergencies vehicles (e.g, fire truck and ambulances), to inform people about emergencies. Thus, after losing network connectivity, wireless devices can listen for sirens to further validate the occurrence of a disaster and activate the E-DARWIN application. Furthermore, as devices always have their microphone turned on, detecting the sirens will only utilize the device's CPU.

Sirens are high amplitude sound emitted by the emergency vehicles and devices. 'Wail' and 'Yelp' are two of the most commonly used sirens. Table 5.1 gives the specifications of the two sirens. Each cycle in a siren consists of a frequency sweep

Type of Siren	Minimum Fundamental Frequency (Hz)	Maximum Fundamental Frequency (Hz)	Cycle Rate
Wail	600 Hz	1200 Hz	12 cycles/min
Yelp	600 Hz	1200 Hz	180 cycles/min

Table 5.1. Specifications of Sirens

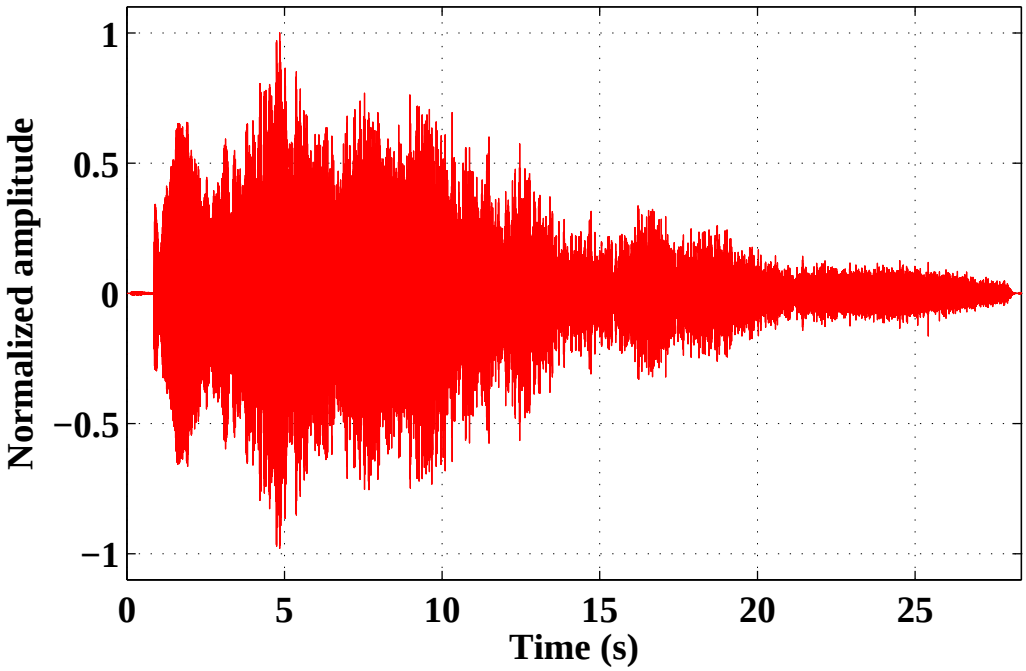

Figure 5.3. Waveform of Captured Wail Siren

from the minimum to maximum fundamental frequency. To detect these sirens, a low power system is designed fin this work or wireless devices.

Let T_{siren} be the duration of a single cycle of a siren, which is divided into multiple time slots of size t_{slots}. Samples of environmental noises are collected for each time slot and analyzed. Short Time Fourier Transform of the collected sample for each time slot is computed to convert them into frequency domain. The samples in frequency domain are then passed through a Hanning window to remove spectral leakage. From the samples of each time slot, the frequency with highest amplitude

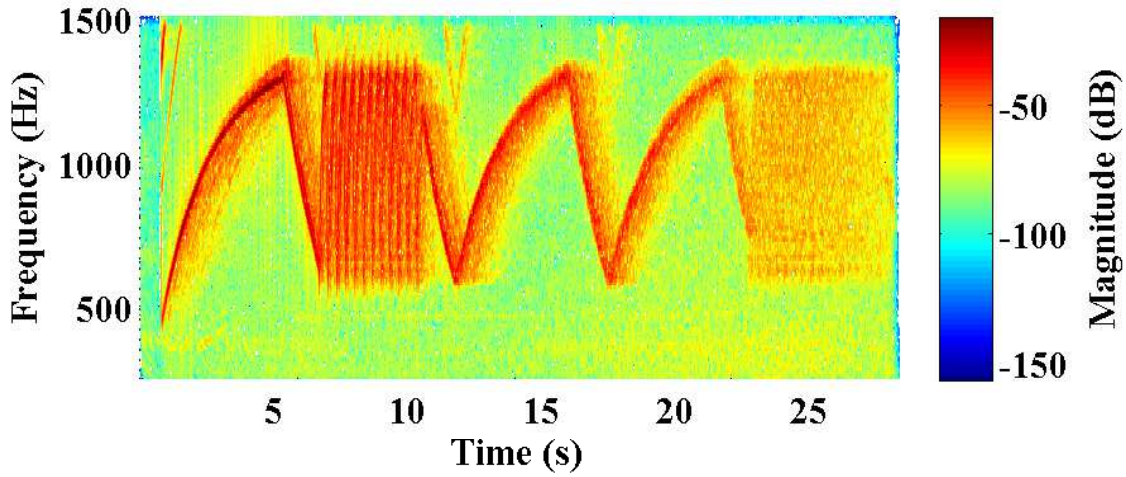

Figure 5.4. Spectrogram of Captured Wail Siren

signal is selected as the dominating frequency of the sample. After the dominating frequency of $\frac{T_{siren}}{t_{slots}}$ samples are computed, they are analyzed. To detect the siren, a scan of observed dominating frequencies is done to determine if it constitutes a frequency sweep and how many samples are out of sequence. The percentage of samples out of sequence is calculated and compared with a threshold to determine if a siren was present or not. The algorithm was implemented on Android platform, and was used to detect a sequence of wail and yelp sirens being emitted by a moving ambulance on a road intersection. Figures 5.3 shows the waveform, and Figure 5.4 shows the spectrogram of the captured sound data. The device was configured to capture the sound at 44.1k samples/sec and t_{slot} was selected as 11.61ms with a threshold of 10%. As shown in Figure 5.4, the wail siren was detected between time 0 to 5 seconds. On detection of the siren, the E-DARWIN application is activated and it enters the neighbor discovery and synchronization phase.

5.5.2. Neighbor Discovery and Synchronization. When the E-DARWIN application is activated, the devices need to discover their neighbors and learn their neighbor's schedule. Existing solutions in literature for neighbor discovery and synchronization rely on coordination between the devices to ensure that their radios are

Algorithm 1 E-DARWIN - Neighbor Discovery and Synchronization

Require: Initialize the set of neighbors $N = \emptyset$
Require: Configure the device to act as WiFi hotspot for $T_{initial}$ time
Require: H_{next} = Generate uniform random no.$(0, H_{max})$
Require: Set the device to be WiFi hotspot after $(T_{initial} + H_{next})$ time
Require: C_{next} = Generate uniform random no. $(0, C_{discover})$
Require: Set the device to be WiFi client after C_{next} time

```
while In initialization phase do
    if Device enters client state then
        Set the device to be WiFi hotspot again after I_discover time
        C_next = Generate uniform random no. (0, C_discover)
        Set the device to be WiFi client after (I_discover + C_next) time
    end if
    for Each new neighbor n do
        N = N ∪ {n}
        Inform the device will be hotspot after (T_initial + H_next) time
        Receive from n, when it will be hotspot again (H^n_next)
        Set device to be in client state after H^n_next time
    end for
end while
```

simultaneously turned on for discovering one another (44). However, to allow two devices using WiFi Tethering to discover one another, they must not only have their radios turned on but be in different states, i.e., they should act as WiFi hotspot and WiFi client respectively.

To address this problem, the devices are restricted to stay in the hotspot state during initialization for $T_{initial}$ time, as shown in Algorithm 1. The devices successively take up the role of WiFi client after random time intervals (C_{next}), that is uniformly distributed between $(0, C_{discover})$. After initialization, the devices by default stay in the dormant state, and successively take up the role of WiFi hotspot after random

time intervals (H_{next}), that is uniformly distributed between $(0, H_{max})$. The random role assumption allows devices to be in different state after random time intervals, and discover one another.

During initialization, a device in the hotspot state waits for one of its neighbors to become a client and associate with it. When the device becomes a client, it stays in that state for ($I_{discover}$) time. The client device periodically scans the wireless channel (T_{scan}) for advertisements from one of its neighbors, which may be in hotspot state. On receiving the advertisement from the hotspot, the client device associates with it. On successful association, the devices add one another to the neighbor list (N) and exchange one another's schedule, i.e., when they will become hotspot again ($T_{initial} + H_{next}$) after the initialization phase. It allows devices to be in dormant state after initialization and wake up only when one of their neighbors become hotspot.

To analyze the impact of algorithm parameters on the discovery mechanism, two neighboring devices are assumed to be in initialization phase. The devices will discover one another whenever they are in opposite states for the first time. Assuming the two devices enter the initialization phase together, each device become a client at a given time $t \in \{0, C_{discover}\}$ with probability $\frac{1}{C_{discover}}$. Then the probability that the two devices will become client at the same time and not discover one another within the $C_{discover}$ time interval can be computed as shown in Equation 5.1.

$$P(\text{ Two devices will become client at the same time in } C_{discover} \text{ time}) = \int_0^{C_{discover}} \frac{1}{(C_{discover})^2} dt = \frac{1}{C_{discover}} \quad (5.1)$$

Similarly, the probability that the two devices will be in different state in the $C_{discover}$ interval and discover one another is given by Equation 5.2.

$$P(\text{Two devices will be in different state in } C_{discover} \text{ time}) = \int_0^{C_{discover}} \frac{1}{C_{discover}} \left(1 - \frac{1}{C_{discover}}\right) dt = \left(1 - \frac{1}{C_{discover}}\right) \quad (5.2)$$

The time after which a device chooses to become client is independent of when it was client previously. Thus, the probability that the two devices will become client $(n-1)$ times before discovering one another is given by geometric distribution, as shown in Equation 5.3.

$$P(\text{ Two devices will discover one another in the } n^{th} \text{ instance they are client}) = \left(\frac{1}{C_{discover}}\right)^{n-1} \left(1 - \frac{1}{C_{discover}}\right) \quad (5.3)$$

Assuming the devices become client n times during initialization phase, the probability that they will discover one another can be calculated using Equation 5.4.

$$P(\text{ Two devices in initialization phase will discover one another }) = 1 - \left(\frac{1}{C_{discover}}\right)^{n} \quad (5.4)$$

The neighbor discovery protocol was analyzed for two neighboring devices in initialization phase trying to discover one another with varying $C_{discover}$ and varying time in initialization phase, as shown in Figure 5.5. As $C_{discover}$ increases, the devices have a larger time window to choose from to become a client. Thus, the probability that two devices will become client at the same time and not discover one another decreases with increasing $C_{discover}$. Furthermore, by keeping $T_{initial} \gg C_{discover}$, the

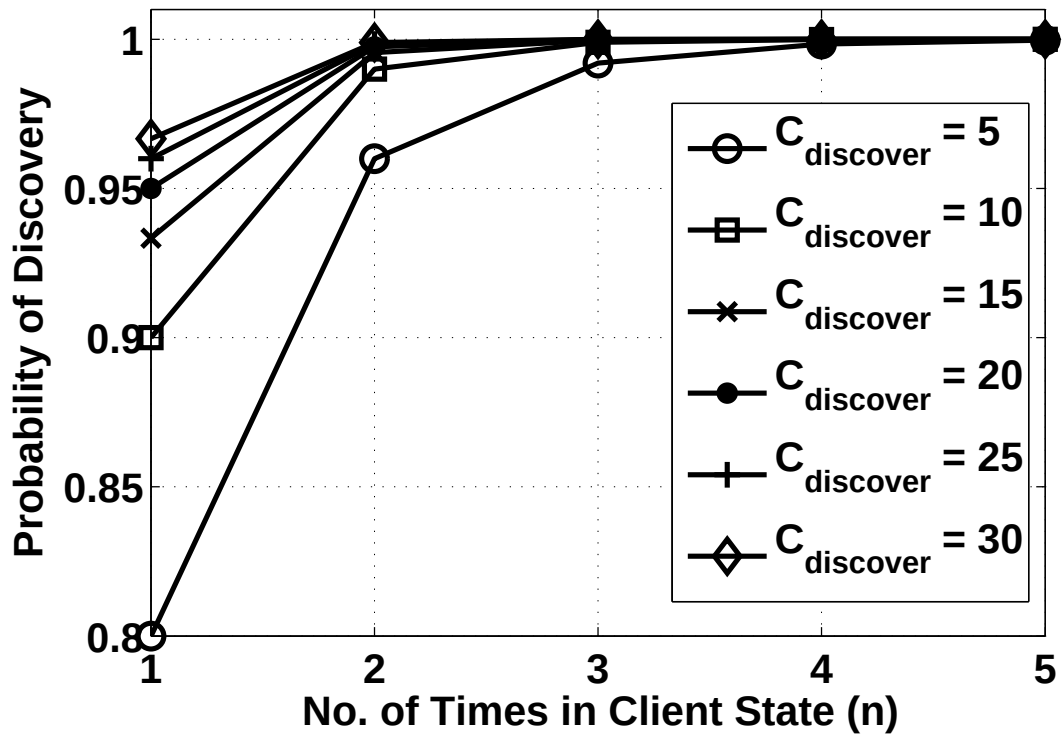

Figure 5.5. Probability of Two Devices in Initialization Phase Discovering One Another

devices can become client multiple times within the initialization phase (n) and increase the probability that they will discover one another as $(\frac{1}{C_{discover}})^n \rightarrow 0$ with increasing n. Based on analytical results, it can be concluded that devices will discover one another if they become client at least 4 times in the initialization phase, i.e., if $T_{initial} \geq 4 \times C_{discover}$.

Additionally, devices in initialization phase must also discover other neighboring devices, which are already initialized. Let us assume, $D_{neighbor}$ is a neighbor of the device in initialization phase, which is already initialized. Both devices will discover one another whenever they will enter opposite states together for the first time. The probability that the initializing device will enter client state and $D_{neighbor}$ will become hotspot respectively at time $t \in \{0, C_{discover}\}$ is given by Equation 5.5.

$$P(\text{Initializing device is client and Initialized device is hotspot at time } t) = \frac{1}{C_{discover}} \times \frac{1}{H_{max}} \quad (5.5)$$

Using Equation 5.5, the probability that the two devices will discover one another in $C_{discover}$ time interval can be calculated as shown in Equation 5.6.

$$P(\text{Initializing device is client and Initialized device is hotspot in } C_{discover} \text{ time}) = \int_0^{C_{discover}} \frac{1}{C_{discover}} \times \frac{1}{H_{max}} dt = \frac{1}{H_{max}} \quad (5.6)$$

Similarly, the probability that the initializing devices will become client and not discover $D_{neighbor}$ in $C_{discover}$ time interval is given by Equation 5.7.

$$P(\text{Initializing device is client and Initialized device is not hotspot in } C_{discover} \text{ time}) = \int_0^{C_{discover}} \frac{1}{C_{discover}} \times \left(1 - \frac{1}{H_{max}}\right) dt = \left(1 - \frac{1}{H_{max}}\right) \quad (5.7)$$

After initialization, devices become hotspot independent of the time when they were hotspot last. Thus, given the initializing device becomes client $(n-1)$ times during initialization phase before discovering $D_{neighbor}$ is given by geometric distribution and can be computed as shown in Equation 5.8.

$$P(\text{Initializing and Initialized device will discover one another in the } n^{th} \text{ instant the initializing device becomes client}) = \left(1 - \frac{1}{H_{max}}\right)^{n-1} \frac{1}{H_{max}} \quad (5.8)$$

Assuming the initializing device becomes client n times during initialization phase and will discover $D_{neighbor}$ is computed as given by Equation 5.9.

$$P(\text{Initializing and Initialized device will discover one another as client and hotspot respectively in initialization phase}) = 1 - \left(1 - \frac{1}{H_{max}}\right)^{n} \quad (5.9)$$

The two devices will also discover one another when the initializing device is hotspot and the $D_{neighbor}$ becomes client. Assuming $D_{neighbor}$ has m neighbors, which are already initialized. $D_{neighbor}$ will become a client whenever at least one of its m neighbor becomes a hotspot and it is not scheduled to become hotspot. Given each device operates independent of one another, the probability that $D_{neighbor}$ will become client at a given time t is given by Equation 5.10.

$$P(\text{Initialized device becomes client at time } t) = \left(1 - \frac{1}{H_{max}}\right)\left(1 - \left(1 - \frac{1}{H_{max}}\right)^{m}\right) \quad (5.10)$$

Let us assume devices become hotspot for I_H time interval after initialization phase. I_H is chosen greater than $I_{discover}$ as the devices must not only allow other devices to connect to it, but also select the relay node and facilitate offloading of data to it. Thus, if $D_{neighbor}$ and the initializing device become client at the same time, the

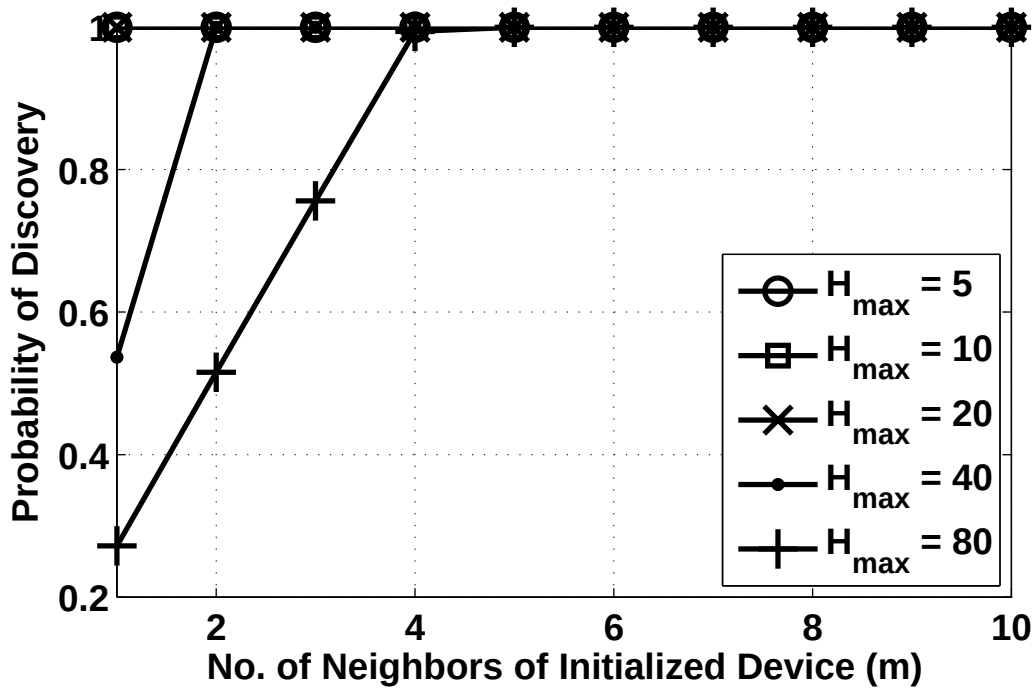

Figure 5.6. Probability of a Device in Initialize Phase discovering an Initialized Device having m neighbors with $n = 2$ and $T_{initial} = 4 \times C_{Discover} = 20$ mins

initializing device will exit client phase first and become hotspot for the two devices to discover one another. Hence, the devices will discover one another whenever $D_{neighbor}$ will become client first, as shown in Equation 5.11.

$$\begin{aligned} P(\text{Initializing and Initialized device will discover one another as hotspot and} \\ \text{client respectively within time } t) &= \int_0^t \left(1 - \frac{1}{H_{max}}\right)\left(1 - \left(1 - \frac{1}{H_{max}}\right)^m\right) dt \\ &= t\left(1 - \frac{1}{H_{max}}\right)\left(1 - \left(1 - \frac{1}{H_{max}}\right)^m\right) \end{aligned} \tag{5.11}$$

Figure 5.6 and Figure 5.7, show the probability of a device in initialization phase and an initialized with m neighbors discovering one another. The devices are more likely to discover one another when the initializing device is in hotspot state and

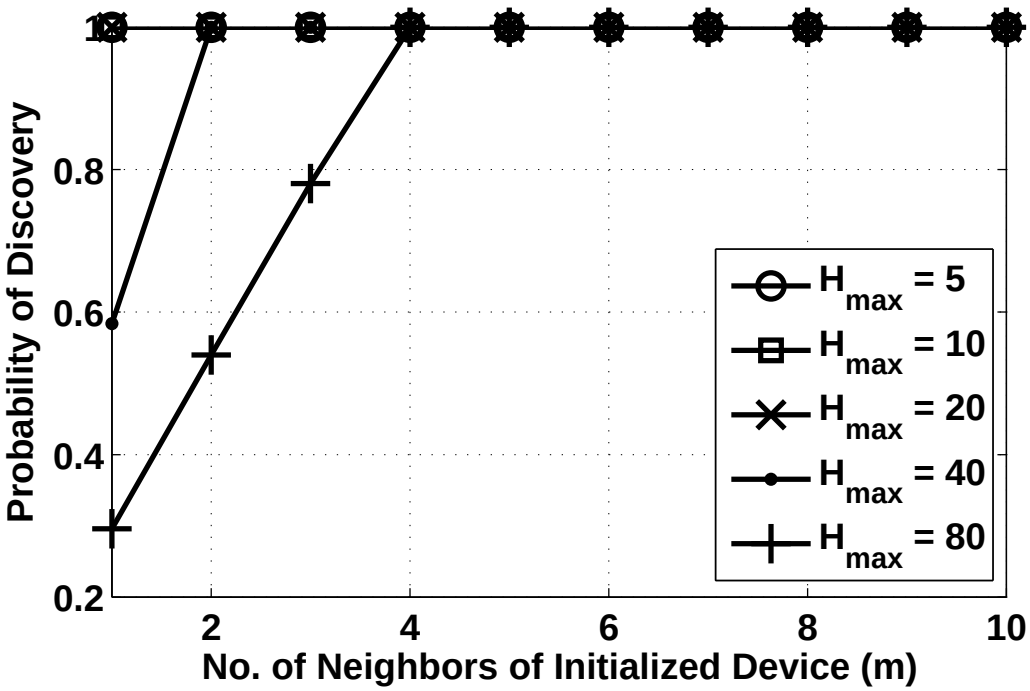

Figure 5.7. Probability of a Device in Initialize Phase discovering an Initialized Device having m neighbors with $n = 4$ and $T_{initial} = 4 \times C_{Discover} = 20$ mins

the initialized device becomes a client. As the number of neighbors of the initialized device increases, it enters client state more frequently. Hence, the two devices discover one another faster. With increasing H_{max}, initialized devices become hotspot less frequently. This reduces the frequency with which the neighbors of an initialized device become hotspot, and hence, the frequency with which the initialized device becomes client. Thus, reducing the probability that the two devices will discover one another. Keeping $T_{initial} > 4 \times C_{Discover}$ will provide initializing device with more opportunities to discover the initialized devices. To find the initialization time from Equation 5.11 for which the two devices will discover one another, the minimum number of neighbors are assumed, i.e., $m = 1$, and the equation is solved for t. Thus, the two devices will always discover one another, irrespective of the number of neighbors the initialized device has, if $T_{initial} = H_{max} - 1$.

Algorithm 2 WiFi Client

Ensure: Device is configured to act as WiFi client for I_H time
Ensure: Device is associated with at least one WiFi hotspot
Require: $i \leftarrow$ Device id
Require: $D_H \rightarrow$ Set of associated WiFi hotspot devices
Initialize the estimated delivery time using the device as a relay node as $t_i = \infty$
if {WiFi AP} $\in N$ **then**
    if {WiFi AP} $\in D_H$ **then**
        Data can be delivered immediately, i.e. $t_i = 0$
    else
        Data can be delivered after $t_i = I_H$ time
    end if
else
    Data can be delivered after $t_i = min(M)$ time
end if
Inform to all hotspots $d \in D_H$ that if i is chosen as relay node then data will be delivered to WiFi AP after t_i time
Ensure: Receive from hotspot the relay device id (r)
if client device (i) is not the relay device (r) **then**
    Offload data to relay device (r) through the hotspot
end if
Ensure: Receive message from each hotspot $d \in D_H$ of the time interval H^d_{next} after which it will act as hotspot again and the estimated data delivery time $T^d_{delivery}$ using them
for all $d \in D_H$ **do**
    Set the device to act as WiFi client after H^d_{next} time
    $M(d) = T^d_{delivery}$
end for

5.5.3. Forwarding Data in E-DARWIN. The devices start capturing and forwarding data to the WiFi AP after the initialization phase. The problem of routing data in ad hoc networks has been extensively studied in literature (16), wherein the devices select a neighbor to forward data based on a given route selection algorithm. In E-DARWIN, devices offload data to another device selected as the relay node until they are delivered to the WiFi AP. However, the relay device may not be a neighbor of the device offloading the data as they can be connected through a hotspot and not be aware of one another. Additionally, the relay device must be selected such that the

Algorithm 3 WiFi Hotspot

Ensure: Device is configured to act as hotspot for I_H time
Require: $D_C \rightarrow$ Set of associated WiFi client devices
Ensure: Receive delivery estimate t_c from each device $c \in D_C$

```
Define a function F
for all c ∈ D_C do
    F(c) = t_c
end for
if {WiFi AP} ∈ N then
    F(hotspot) = I_H
else
    F(hotspot) = min(M)
end if
r = argmin(F)
Broadcast relay device is r
H_next = Generate uniform random no. (0, H_max)
Set the device to act as WiFi hotspot after H_next time
Initialize the estimated delivery time when the device acts as hotspot next, i.e.
after H_next time, as T_delivery = ∞
for all c ∈ D_C do
    if t_c == 0 then
        Data can be delivered immediately to WiFi AP using a relay device con-
nected simultaneously to both the hotspot and WiFi AP, i.e., T_delivery = H_next
    else if t_c == I_H then
        Data can be delivered to the WiFi AP using a relay device connected to
the hotspot immediately after the device ceases to be hotspot, i.e., T_delivery =
H_next + I_H
    else t_c > H_next and T_delivery > t_c
        Data can be delivered to the WiFi AP using a relay device connected to
the hotspot after T_delivery = t_c time
    end if
end for
Broadcast device will be hotspot again after H_next time and data can be delivered
using it in T_delivery time
```

data can be delivered to the WiFi AP earliest based on the schedule of the devices. To enable this, each device maintains a function M, which stores the information on how soon the data can be delivered to the WiFi AP using a relay device connected to one of its neighbors acting as hotspot. Algorithm 2 and Algorithm 3 describe how the mapping function is updated and used to forward data to the WiFi AP in E-DARWIN in client and hotspot state respectively.

On being configured as WiFi client, each device scans the wireless channel for advertisements from its neighbors scheduled to act as hotspot then. After discovering and associating with the hotspot, each client device estimates when data can be delivered to the WiFi AP using it as the relay device and informs the hotspot of its estimated delivery time (STEPS 1-11 of Algorithm 2). Let us assume, the client device has the WiFi AP as its neighbor and is capable of connecting to multiple hotspots at the same time. Thus, the device can also connect to the WiFi AP while it is connected to the hotspot. In this scenario, the client device can forward data to the WiFi AP as soon as it receives them and will estimate its delivery time as 0 (STEP 4). On the other hand, the client device may have the WiFi AP as its neighbor but may not have the capability to connect to multiple hotspots at the same time. The client device can deliver data to the WiFi AP after I_H time when it disconnects from the hotspot and associates with WiFi AP. Hence, it estimates its delivery time as I_H (STEP 6). Finally, if the WiFi AP is not in the communication range of the client device, it must forward data to the WiFi AP using one of its neighboring devices. Each device maintains a function M, which stores the information that when can the data be delivered when one of its neighbor acts as hotspot. The client device chooses its estimated delivery time as the minimum of the estimated delivery time of its neighbors using the function M (STEP 9). The hotspot on receiving all the delivery estimates from its clients, determines which device has the lowest delivery estimate and announces it as the relay node (STEPS 1-11 in Algorithm 3). The client

devices on receiving the information, offload all the stored data to the relay device (STEPS 12-14 in Algorithm 2).

Meanwhile, the hotspot determines when it will act as hotspot again T_{next} and schedules itself to act as hotspot next (STEPS 12-13 of Algorithm 3). Additionally, the mapping function of the clients need to be updated to reflect the estimated delivery time when the device acts as hotspot again (STEPS 14-24 of Algorithm 3). If one of the clients is connected to WiFi AP, then data can again be forwarded to the WiFi AP as soon as the device acts as hotspot again. Hence, the delivery time is estimated as the time when the device will become hotspot again, i.e. H_{next} (STEP 17). If the client cannot connect to more than one hotspot at a time, it can deliver the data to the WiFi AP after it disassociates from the hotspot next, i.e. $H_{next} + I_H$ (STEP 19). If none of its clients are in the communication range of the WiFi AP, then the hotspot must forward data to the ECC through one of its clients acting as a relay device. Therefore, the hotspot determines which of its clients will deliver the data earliest after H_{next} time, i.e. when it will act as hotspot again and announces it as its estimated delivery time ($T_{delivery}$) to all its clients. The clients on receiving the information, update their mapping function accordingly and set themselves to be client again after the specified time interval (STEPS 15-18 of Algorithm 2). The clients in turn propagate the delivery estimates in the network through other neighbors acting as hotspot.

5.6. ENERGY AWARE DISTRIBUTED COALITION FORMATION

Wireless devices are increasingly being equipped with a wide array of sensors, which can also be used to capture data in post-disaster scenarios. The problem of data capturing and gathering has been extensively studied in literature, especially in the context of wireless sensor networks. The solutions discussed in literature primarily perform in-network data aggregation or compression to reduce the communication cost in the network or use model driven approaches to reduce the amount of data being

forwarded in the network by suppressing redundant information (18). However, these approaches require frequent interaction between the nodes and flow of information in the network, like, parameters for predicting the behavior of the random process and error estimates. In the E-DARWIN architecture, the devices must wait for one of their neighbors to become hotspot to forward data. Thus, the flow of information in the network is restricted by the frequency with which the devices become hotspot, which makes these traditional approaches infeasible when the frequency is kept low to conserve energy. To address this problem, a distributed coalition formation game is designed to suppress the data capturing task at source by exploiting the spatial correlation between the devices.

The data requirements of disaster recovery network can vary from collecting spatial data, like, GPS, for tracking the affected population (26), temperature and humidity readings from bluetooth based acquisition systems for environmental assessment, text based inputs for nutritional and health assessment (114; 131), audio samples captured from the environment for detecting trapped victims (45) as well as capturing video and images to build a spatial view of the affected region for infrastructure assessment (114). Each data requirement represents a real time process that the ECC wants to monitor. Let the data requirements of ECC be represented by the set $X = \{x_1, x_2, \ldots, x_n\}$. For each process $x_i \in X$, the ECC defines points of interest in the affected region, and wants devices in their vicinity to collect samples of data and send them with frequency f_i. The ECC collects samples of each data over a decision interval (τ) and then analyzes them to detect events and make decisions accordingly. The ECC informs each device on initialization the set of physical processes X it wants to monitor and the frequency f_i with which it wants each data source in the network to capture samples of data for each process $x_i \in X$. The ECC broadcasts the data requirements to all WiFi APs through the satellite link, which in turn broadcast them to all the devices connected to them. The devices in turn

propagate the information in the network by broadcasting it to their clients when they act as hotspot or through a neighboring device acting as hotspot.

On learning about the data requirements, each device determines the subset of data requirements it can capture and starts capturing and sending samples of them to the ECC with the desired frequency. However, not all devices are required to participate in the data capturing process, as samples from sources with high spatial correlation are redundant. This provides a set of highly correlated devices with the incentive to collaborate with one another and act as a single data source, i.e., collectively contribute data with frequency f_i, for reduced energy consumption. In (126), the authors define a distortion function $D_{x_i}(S)$ to measure the difficulty in estimation of events in a physical process x_i from samples captured by spatially correlated sources S and is used in this work to represent the redundancy among the samples. Thus, for a given process $x_i \in X$, the ECC can accurately detect events from samples as long as the level of distortion $D_{x_i}(S)$ or redundancy in the samples from the set of devices S is below a threshold $D^*_{x_i}$. Hence, a set of highly correlated devices whose samples are distorted above the threshold $D^*_{x_i}$, can collaborate and form coalitions to share the data capturing task among them for reduced energy consumption.

However, for the devices to form coalitions or join them, they must be able to evaluate and compare different coalitions and decide which coalition to join. To this end, utility of a coalition is defined. In coalition game theory, the utility of a coalition represents the worth of a coalition and is distributed among the devices in the coalition based on a solution concept. By relating the utility to the frequency with which a device captures data, the devices can evaluate, compare and decide which coalition to join. In this work, the utility is defined based on the following premise: the allocation of data capturing task should prefer devices with high available energy and low network participation. It is assumed that a device j allocates a fraction $E_j^{x_i}$ of its available energy (E_j) for capturing samples of the data $x_i \in X$, such that

$\sum_{x_i \in X} E_j^{x_i} = E_j$. However, the device's available energy will be used for network operations.

When acting as hotspot, the amount of energy a device consumes increases with the number of neighbors because more neighbors mean a higher volume of traffic flowing through it to the relay node. Additionally, the amount of time a device spends as client depends on the number of neighbors it has and the frequency with which its neighbors become hotspot. Given that all devices become hotspot with the same frequency, a device with more neighbors will switch to client state more frequently and spend more energy in network operations. Therefore, devices with more neighbors should be allocated a lesser burden of capturing data for higher network lifetime. Hence, when determining the contributions of a device j in a coalition, the total energy allocated by the device for capturing samples of a process is divided by the number of neighbors N_j it has. The total utility of a coalition represents the total weighted available energy of the devices in a coalition for capturing data and is given by Equation 5.12. Therefore, the distribution of the total utility among the devices in a coalition can be used to determine how much data capturing task should be allocated to each device in the coalition. To distribute the utility among the devices in the coalition, a payoff vector is defined ($\phi^v = \{\phi_1^v, \ldots, \phi_{|S|}^v\}$) that represents the distribution of the utility $v(S)$ among the set of devices S. A simple and strict method is then proposed for distributing the total utility among the devices, i.e. Proportional fairness, wherein the total utility is distributed among the devices in the ratio of their contributions, as shown in Equation 5.13, where $\sum_{j \in S} w_j = 1$ and within the coalition $\frac{w_i}{w_j} = \frac{v(\{i\})}{v(\{j\})}$. Hence, the ratio of utility that each device $j \in S$ in a coalition receives is representative of the ratio of data it should capture, i.e. $w_j f_i$. Furthermore, by keeping the allocated energy for capturing samples of each data independent of one another, the algorithm can be executed independently for each data requirement

$x_i \in X$.

$$v(S) = \begin{cases} \sum_{j \in S} \frac{E_j^{x_i}}{N_j} & \text{if } D_{x_i}(S) > D_{x_i}^{\star} \\ 0 & \text{Otherwise} \end{cases} \tag{5.12}$$

$$\phi_j^v = w_j v(S), \forall j \in S \tag{5.13}$$

5.6.1. Formulation of Distributed Coalition Formation Game. In this section, a transferable utility (TU) coalition formation game is defined using which devices in the network can form coalitions. Let us define a coalition game $G = (D, v)$, where D is the set of players representing each device and v is the utility function given by Equation 5.12. Coalitions in the network can be formed with a centralized approach; however determining the optimal coalitions is NP-hard (20), as it requires iterating over all possible coalitions of D devices. Thus, a distributed coalition formation approach is more desirable. In (20), the authors provide a generic framework for forming coalition games among players using two simple merge and split rules, which can be applied in a distributed manner. However, before the framework can be adapted and applied to the problem, the following concepts need to be defined.

Definition 5.6.1 *A collection of coalitions in the grand coalition D, denoted as K, is defined as the set $K = \{K_1, K_2, \ldots K_l\}$ of mutually disjoint coalitions K_i of D. In other words, a collection is a random group of disjoint coalitions K_i of D not necessarily including all players of D. If the collection includes all player of D, i.e. $\cup_{i=1}^{l} K_i = D$, the collection is referred to as partition of D.*

Definition 5.6.2 *A preference operator or comparison relation $\triangleright$ is defined for comparing two collections $O = O_1, O_2, \ldots, O_r\}$ and $P = \{P_1, P_2, \ldots, P_s\}$ with the same set of players A (i.e. $A = \cup_{i=1}^{r} O_m = \cup_{j=1}^{s} P_j \subseteq D$). Thus, $O \triangleright P$, implies that the way O partitions A is preferred over the way P partitions A.*

The comparison relation ($\triangleright$) provides players with a way to compare coalitions before joining or splitting from them. Among several well known comparison relations

discussed in literature (20), the Pareto order correctly captures the properties of the game. The Pareto order is defined as shown in Equation 5.14, with at least one strict inequality for a player $k \in O, P$. This means that players in coalitions in P have the incentive to deviate and form coalitions given by O as at least one player can improve its payoff without reducing that of others. Thus, by changing the coalition structures from P to O or vice versa, at least one player can get a smaller allocation of frequency of data capture resulting in lower energy consumption rate without increasing that of others. Intuitively, this means that all devices in coalitions have incentive to leave, join or form new coalitions if at least one device can get a smaller allocation of frequency of capturing data without increasing that of others, resulting in lower energy consumption for the device.

$$O \triangleright P \Leftrightarrow \{\phi_j^v(O) > \phi_j^v(P), \forall j \in O, P \tag{5.14}$$

Based on the above formulations, the rules for merging and splitting of the coalitions are defined as follows.

- Merge Rule: Merge the coalitions $\{C_1, \ldots, C_l\}$ if $\{\cup_{z=1}^{l} C_z\} \triangleright \{C_1, \ldots, C_l\}$
- Split Rule: Split the coalition $\{\cup_{z=1}^{l} C_z\}$ if $\{C_1, \ldots, C_l\} \triangleright \{\cup_{z=1}^{l} C_z\}$

The merge rule specifies that if at least one device in the coalitions $\{C_1, \ldots, C_l\}$ can achieve a lower frequency of capturing data by merging them, the devices will merge and form a single coalition $\{\cup_{z=1}^{l} C_z\}$. Similarly, the split rule specifies that if one of the devices in the coalition $\{\cup_{z=1}^{l} C_z\}$ can achieve a lower frequency of capturing data by splitting the coalitions into smaller coalitions, the devices will split and form smaller coalitions $\{C_1, \ldots, C_l\}$.

Theorem 5.6.3 *In the game G, random iterations of the merge and split rule will always terminate in finite steps from any arbitrary starting point to form stable coalitions.*

Proof The $\triangleright$ operator presents a Pareto optimal solution. As a result, the iterations of merge and split rule will always terminate (20; 110).

5.6.2. Implementation of the Coalition Game. On receiving the data requirements, each device creates a coalition with only itself in the coalition. Whenever a device acts as hotspot, all its neighbors send their coalition information including payoff of each device in the coalition. On receiving the information, the hotspot determines if it should split its coalition or merge it with its neighbors, i.e., if it would get a smaller allocation of frequency by merging or splitting its coalition. If it decides to merge or split the coalition, it first determines the validity of the new coalitions, then computes the new data capturing frequency allocations and informs its neighbors affected by its decision. The client devices in turn propagate the information to their neighbors in the coalition when they act as hotspot or client again. The proposed algorithm is repeated periodically, enabling devices to autonomously adapt to changes in available energy and mobility, if any, of the devices.

5.7. PERFORMANCE EVALUATION

The proposed architecture was first implemented on Android platform and deployed on a Google Nexus One device running Android OS 2.3.6. The device was used to characterize the energy consumed by the application in the initialization phase, and provide guidelines for selection of various algorithm parameters based on it. Based on the provided guidelines, algorithm parameters were selected for the data forwarding phase, and large-scale simulations were conducted to evaluate it. Each data point represents the average value observed over 25 iterations of each experiment.

5.7.1. Prototype-based Evaluation. The PowerTutor energy profiler was used to measure the energy consumption of the E-DARWIN application (8). In the

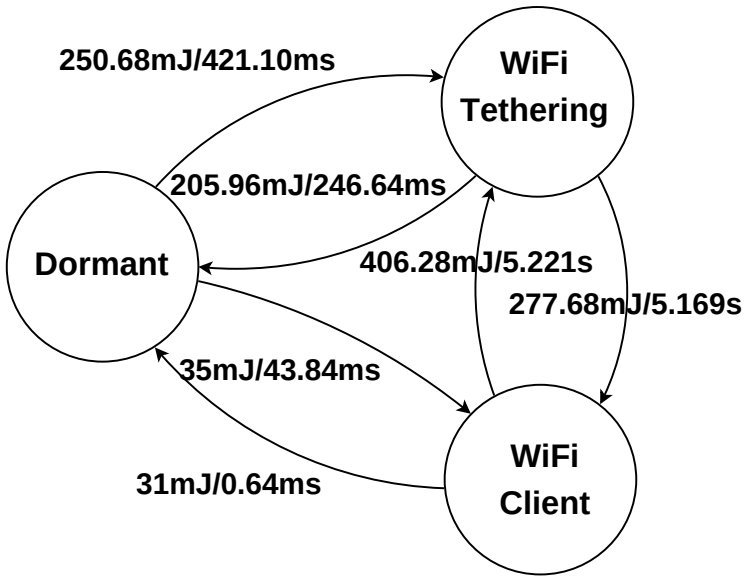

Figure 5.8. Energy and Time Consumed in State Transitions in Initialization Phase

first set of experiments, the energy and time consumed in transitioning between different states were measured, as shown in Figure 5.8. More energy and time is consumed in transitioning between dormant and WiFi Tethering state than WiFi client state. The device not only needs to enable or disable the wireless interface but also store or retrieve the hotspot configuration, and create or destroy the wireless network with the given configuration. Maximum energy and time is consumed transitioning from the WiFi client state to the WiFi Tethering state as the application needs to store the client state, disassociate with the hotspot, disable the wireless interface, retrieve the hotspot configuration and then create the hotspot with the given configuration.

In the initialization phase, the frequency with which a device becomes client depends on $C_{discover}$. A large value of $C_{discover}$ will ensure less energy is consumed due to lesser number of transitions to or from the WiFi Tethering state. However, this implies that the devices will spend more time in hotspot state. In client state, the devices scan the network for advertisements from hotspot. Thus, the client devices must scan the network at least once within $I_{discover}$ to discover the hotspot, i.e $T_{scan} \leq$

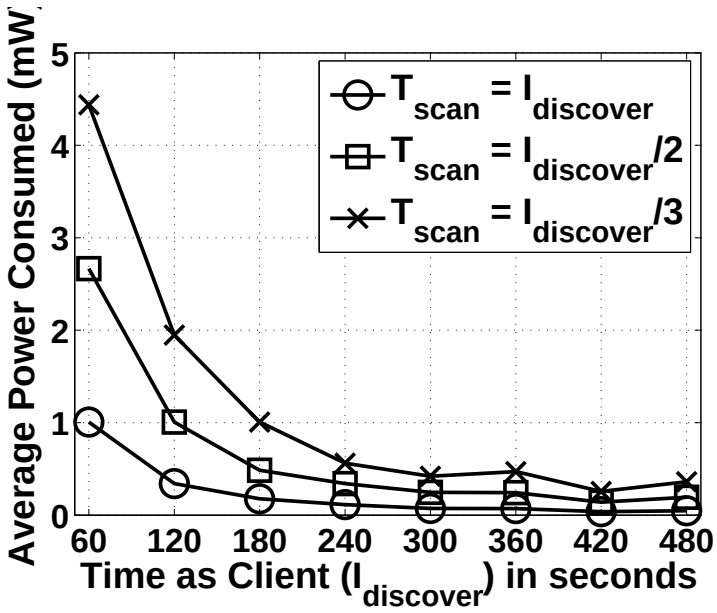

Figure 5.9. Average Power Consumed in Client State in Initialization Phase

$I_{discover}$. To study the impact of $I_{discover}$ and T_{scan} on the power consumption of the device in client state, experiments were conducted with $T_{initial} = 40$ mins and varying $I_{discover}$ and T_{scan}. The results are shown in Figure 5.9. As the frequency with which the devices scan the network increases, the sleep cycle of the WiFi interface is interrupted more frequently, which results in less sleep and higher power consumption. Thus, keeping $T_{scan} = I_{discover}$ ensures that the client device scans the network at least once to receive advertisements from the hotspot and minimum power is consumed. However, the advertisements may be lost due to variations in wireless channel or interference. Hence, T_{scan} is kept as $\frac{I_{discover}}{3}$ to ensure that the client devices have more opportunities to discover a hotspot. Furthermore, the power consumed by the device decreases with increasing $I_{discover}$ as the scanning frequency comparatively decreases. Thus, keeping $I_{discover}$ high, results in lower rate of energy consumption for the client devices. But it implies that the devices stay longer in hotspot state, which reduces the likelihood of the devices discovering another initialized device.

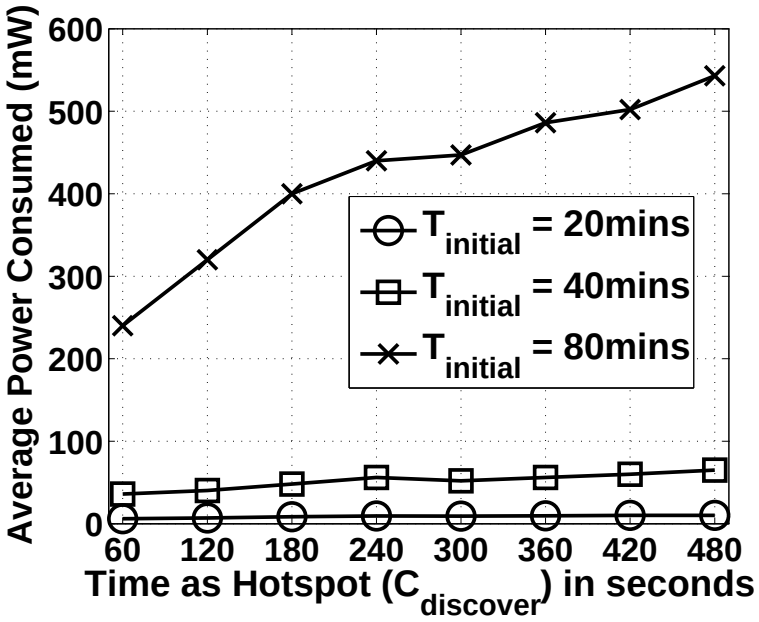

Figure 5.10. Average Power Consumed in Hotspot State in Initialization Phase

The impact of $T_{initial}$ and $C_{discover}$ on the energy consumption of the device was then studied, as shown in Figure 5.10. $I_{discover}$ was kept constant as 1 min. The power consumed by the application in hotspot state increases with increase in $C_{discover}$ as the wireless interface needs to be active for the complete duration. Thus, while higher $C_{discover}$ implies lower power consumption due to reduced state transitions, it comes at a cost of higher rate of energy consumption for the devices in hotspot state. As $T_{initial}$ increases, the total activity time of the application increases, which results in more battery consumption.

5.7.2. Simulations. The proposed solution was implemented using MiXiM-INET framework in OMNET++ simulator. The deployment area was assumed to be 1 sq km. The number of devices were varied from $300 - 700$ in steps of 100 to represent varying populations densities. The devices were distributed uniformly at random in the deployment area based on the observed distribution of population count in real-world data sets (111). The time a device stays in initialization phase

Table 5.2. Simulation Parameters for E-DARWIN Simulation

Carrier Frequency	2.4 GHz
Max. Transmission Power	110.11mW
Radio Propagation Model	Two Ray Path Loss Model
Thermal Noise	-110dBm
Modulation Scheme	Phase Shift Keying (PSK)
Receiver Sensitivity	-95.0dBm
Data Rate	11, 54 Mbps
Packet Size	40 bytes
MAC Protocol	IEEE 802.11
Simulation Time	24hrs
Antenna Model	Omni-directional
Battery Model	Li-Ion
Sampling Rate	1 sample/min
Battery Capacity	11.78Wh

was selected as $T_{initial} = H_{max} - 1$, with $C_{discover} = 6$ mins, $I_{discover} = 1$ min, $I_H = 2$ min, $T_{scan} = \frac{I_{discover}}{3}$ during initialization and $T_{scan} = \frac{I_H}{3}$ thereafter. The simulations were carried out for one component in the network consisting of one WiFi AP with satellite connectivity located at its center. It was also assumed that the ECC requires devices to capture sensory data at the rate of 1 sample/min. The battery capacity and maximum transmission power of each device was selected as $11.78Wh$ and 110.11mW respectively, which is representative of current smartphones. The available energy of each device was chosen uniformly at random between 0 and maximum capacity. The data observed from experiments in Section 5.7.1 on the time and energy consumed in transitioning between various states was input into the battery model of the simulator to make the results as close to reality as possible. The wireless channel was modeled as two ray path loss model with the devices operating at carrier frequency of 2.4 GHz. The simulation parameters are shown in Table 5.2.

In the first set of experiments, the impact of H_{max} on the delay in delivering data to the WiFi AP was studied, as shown in Figure 5.11. As H_{max} is increased, the time interval between a device acting as hotspot increases. As a result, there are

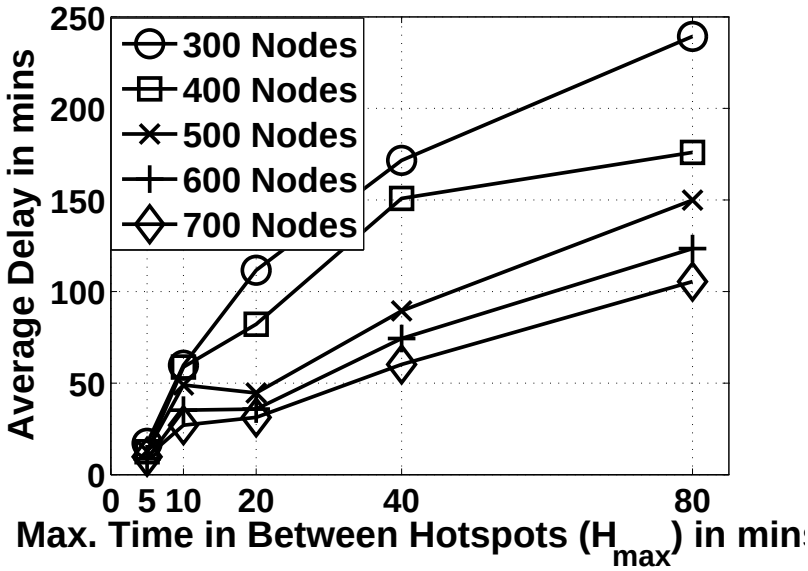

Figure 5.11. Average Delay in Delivering Data to the AP in Data Forwarding Phase

longer delays in delivering data to the WiFi AP as the devices must wait for one of their neighbors to become hotspot to offload data.

For a given H_{max}, as the density of the devices in network increases, the devices spend more time in client state and less in dormant state as they have more neighbors. This results in higher rate of energy consumption for the devices as shown in Figure 5.12. However, the devices have more opportunities to offload data to the relay device, which reduces the delay in delivering data to the WiFi AP. When H_{max} = 5 mins and H_{max} = 10 mins, the devices enter the hotspot state very frequently as a result of which they are always in client or hotspot state but never in dormant state. Thus, the power consumption is highest in both the cases. As H_{max} increases, the devices enter hotspot and consecutively client state less frequently and stay dormant longer, resulting in lower power consumption, as shown in Figure 5.12.

However, with increasing H_{max} more data gets stored in each device, which results in higher volume of traffic being offloaded at each step. Thus, there is more contention among the devices to forward data to the relay device through the hotspot

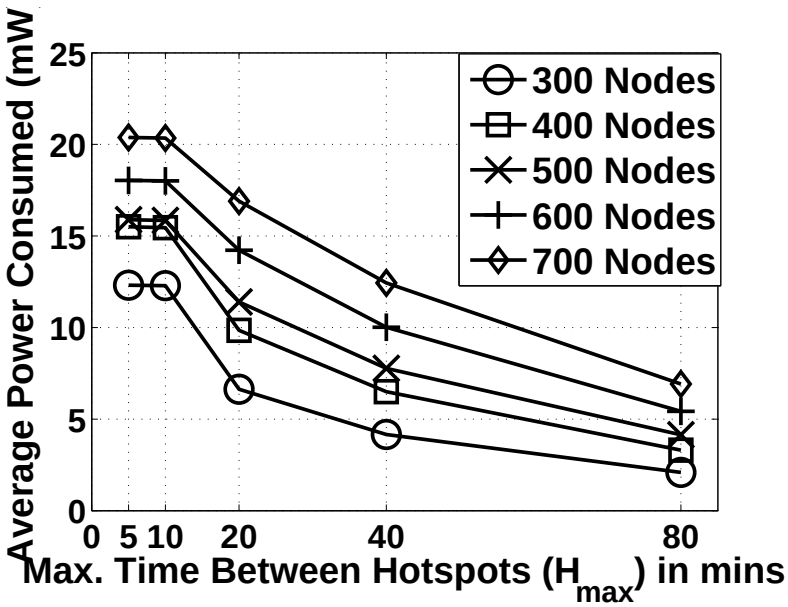

Figure 5.12. Average Power Consumed in Data Forwarding Phase

at each step. The contention results in higher data loss due to congestion and interference at the hotspot, as shown in Figure 5.13. Hence, even though higher H_{max} results in lower rate of energy consumption for the devices, it comes at the cost of higher delay and lower data delivery.

In the second set of experiments, the coalition formation game was evaluated. Among various models used in literature to measure spatial correlation between the devices, the power exponential model is the most popular and is used here (126). The power exponential model is given by $e^{(\frac{-d}{\theta_1})^{\theta_2}}$, with $\theta_1 > 0, \theta_2 \in (0, 2]$, where d is the distance between the devices, θ_1 is the range parameter, which controls how fast the spatial correlation between the devices decay with the distance between them and θ_2 is the smoothness parameter, which controls the geometrical properties of the random field. The model is substituted in the distortion function given by (126) as covariance function to measure the distortion in samples of the devices.

First, it is shown that the proposed coalition formation game converges to a stable solution in small finite steps. To this end, T_{max} is kept as 40 mins, and the

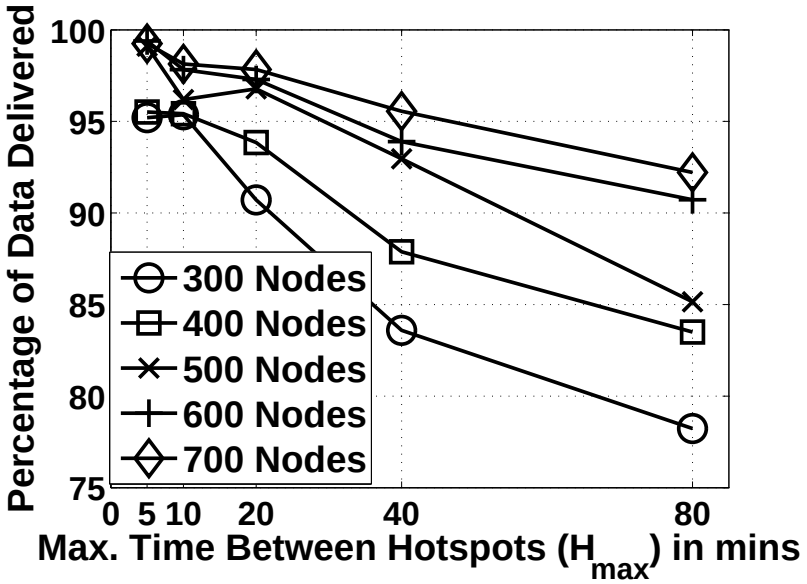

Figure 5.13. Percentage of Data Delivered in Data Forwarding Phase

number of devices in the network is varied as 300, 500 and 700 with $\theta_1 = 100$ to depict different population densities as well as θ_1 as 10, 100, 1000, 10000 with 500 nodes to represent varying degree of correlation between the devices. Figure 5.14 and Figure 5.15 show that the devices converge to a constant number of stable coalitions in the network. With increasing device density, the devices find more devices to collaborate with from their set of neighbors. Hence, the devices form coalitions faster and converge faster. Furthermore, increase in θ_1 implies increased spatial correlation between the devices. Thus, the devices form larger coalitions, which increases the convergence time as the devices need to wait for one of their neighboring devices to become a hotspot before they can propagate their coalition information. At $\theta_1 = 10$, the spatial correlation between the devices is low. As a result, no valid coalition involving two or more devices can be formed and no coalition formation activity is seen.

To evaluate the benefits of the proposed energy aware coalition formation game, the following values are assumed for $\theta_1 = 100$, $\theta_2 = 1$ and $D^{\star}_{x_i} = 2$. The results

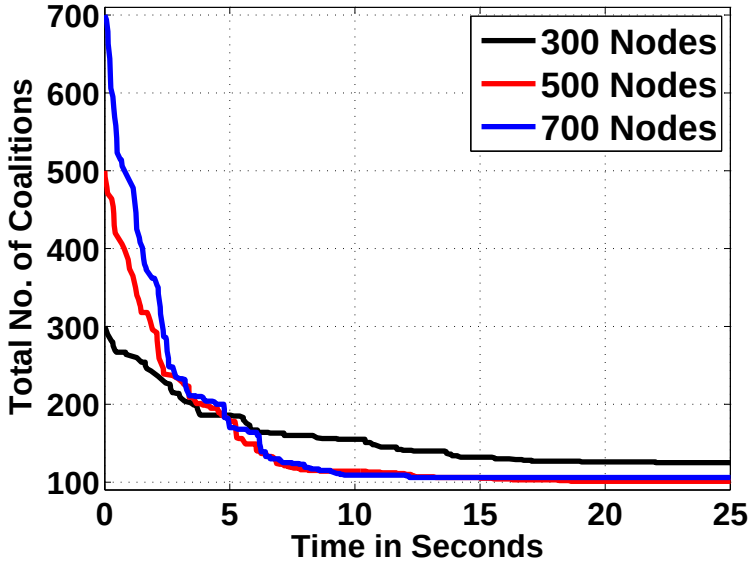

Figure 5.14. Convergence of the Coalition Formation Game with Varying Number of Devices and $\theta_1 = 100$

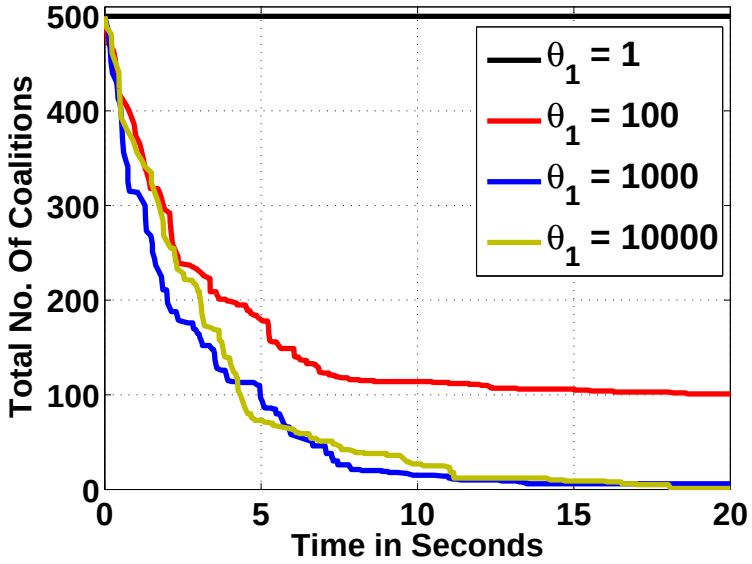

Figure 5.15. Convergence of the Coalition Formation Game with Varying θ_1

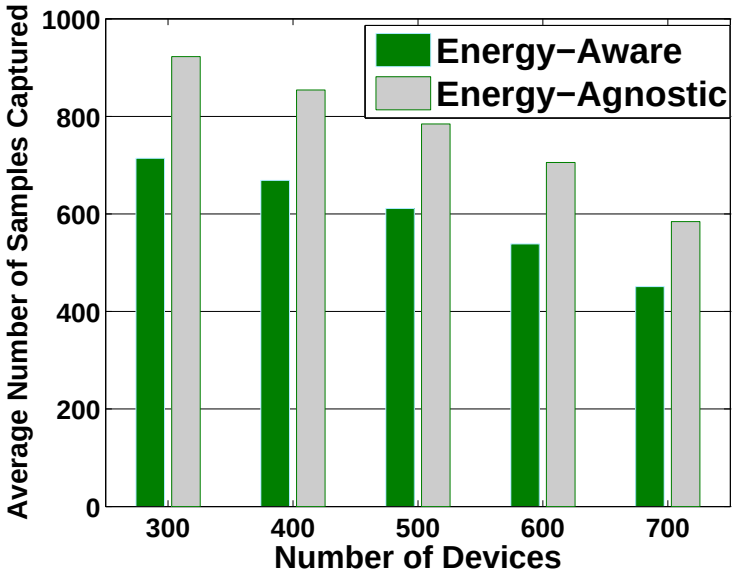

Figure 5.16. Comparison of Energy Aware and Agnostic Approach

are compared with an energy agnostic approach, where the coalitions are formed as per the algorithm discussed in Section 5.6 but without considering the available energy of the device in Equation 5.12. The energy consumed in capturing samples of data, depends on the type of data being captured and varies with it. Therefore, to perform a comparison independent of the cost of the capturing data samples, the average number of samples captured per device using the two approaches are compared, as shown in Figure 5.16. In the proposed mechanism, the devices are allocated the frequency of capturing data based on their available energy and network participation. The process results in higher energy efficiency for the devices, which results in higher degree of cooperation among the devices than energy-agnostic approach.

5.8. UTILIZING RESCUE WORKERS TO COLLECT DATA IN THE E-DARWIN ARCHITECTURE

Rescue workers are the primary mobile entities in the post-disaster scenarios (125). They move back and forth between the relief centers to the affected region

providing supplies, services and aid. The relief camps are assumed to be connected to the ECC through satellite gateways. These workers carry communication devices, which are capable of collecting data from other devices present in the affected region. Thus, they can act as data mules, collecting data from the affected region and delivering it to the ECC through the relief center. This can further reduce the delay in data collection.

5.8.1. Interaction with Rescue Worker Devices. The rescue worker's device must discover other devices in the affected region to collect data. Discovering neighboring devices in a mobile ad hoc network is a major challenge. The wireless interface is one of the major consumer of energy on the devices. Owing to its high energy cost, it is not feasible to keep the wireless interface always on as it may completely drain the battery. To address this problem, a passive low power device discovery mechanism is designed.

The proposed mechanism makes use of near-ultrasonic frequency range to communicate with the devices and make them aware of the presence of the rescue device in their vicinity. The wireless devices are capable of capturing sound waves up to 22.5kHz. Humans can ideally hear sounds in the range of 10Hz to 20 KHz. However, in practice humans can perceive sounds only up to 14kHz. Thus, the frequency range above 14KHz can be used for communication between the devices. In this work, frequency range between 17kHz-18kHz is selected for use. Higher frequency ranges are avoided here as there is noticeable roll off in the accuracy of the speaker above 19kHz.

Rescue workers are assumed to be carrying the wireless devices with portable bluetooth speaker. The speakers are used to amplify the sound waves being emitted by the device with the rescue workers. To broadcast itse presence, the rescue worker's device emits a signal, which is a frequency sweep between the 17kHz-18kHz signal at 120 cycles/second. The wireless devices periodically scan the network for the frequency sweep, using the mechanism discussed in Section 5.5.1. On detection of the

signal, the devices become WiFi clients and connect to the rescue worker's device, which is always in hotspot state. On successfully connecting with the device, the devices determine if the rescue worker can deliver data to the ECC faster using Algorithm 2. It is assumed the rescue workers are aware of their inter-arrival time at the relief centers. If the rescue workers can deliver data to the ECC faster, the devices offload all the stored data to the rescue worker's device after which they go back to their respective states.

To establish the feasibility of the proposed mechanism, it was implemented on the Android platform. The periodic scan interval for the device was selected as 2 seconds after which the device listens to the network for 17ms. The listening interval is chosen as the duration of two cycles to satisfy Nyquist sampling criteria. No power is consumed in activation of any of the device's interface as the microphone is always on. The only energy consumption is due to the usage of CPU for running the signal detection algorithm, which was observed to be very low at 73mW. The energy consumption of the regular wireless interfaces are of a much higher magnitude (123). Thus, the proposed mechanism presents a very low cost passive solution for the wireless devices in the affected region to be aware of the presence of the rescue worker's devices and offload data to them.

5.9. SUMMARY

In this chapter, a novel architecture called E-DARWIN for creating the network infrastructure in disaster affected areas using WiFi Tethering technology was proposed. Specifically, the proposed architecture lets the devices autonomously discover their neighbors and forward data to the WiFi AP with minimum delay. Additionally, to utilize the rich multi-modal sensory capability of the wireless devices, a distributed coalition formation game was designed that allows spatially correlated

devices to share the data capturing tasks among them based on their available energy and network participation. This results in higher network lifetime of the devices, which is crucial in disaster scenarios. These properties were demonstrated through an exhaustive set of experiments and the tradeoff between various network parameters was studied. Finally, a low power passive mechanism was proposed to exploit the presence of mobile entities, like, rescue workers, to reduce the delay in delivering data to the ECC. The proposed mechanism utilizes near-ultrasonic frequencies to let the rescue workers broadcast their presence and allow devices to offload data to them. The data is then carried by the mobile elements, and delivered to the ECC for analysis and decision making. Additionally, having implemented a prototype of the proposed solution on Android platform, the feasibility of the solution has been established. As the proposed architecture offers end-to-end solution for creation and functioning of an ad hoc network using wireless devices for relief and recovery operations, the architecture can be adapted and applied to a wide range of disaster scenarios.

6. CONCLUSION

This work focuses on efficient utilization of resources (e.g. wireless sensors and wireless devices) during various phases of disaster management, namely, prevention, preparedness, relief and recovery. A state-of-the-art survey was conducted on how these resources are being used to address various technological needs of the four phases disaster management. Disaster prevention is an important phase of disaster management. Wireless sensor networks (WSNs) have been playing an important role in timely detection and prevention of disasters. However, the operating environment of the sensors can be hostile. Novel solutions were presented in this work to preserve the location privacy of data sources for secure functioning of WSNs in hostile environments and preventing disasters.

The use of wireless devices in aiding disaster preparedness, relief and recovery operations was investigated next. An energy-aware P2P file sharing framework was proposed for efficient distribution of disaster preparedness content among the users. Finally, an energy-aware architecture was proposed for creation of an ad hoc network infrastructure using wireless devices for disaster relief and recovery operations. Novel mechanisms were designed for utilizing the multi-modal sensory capabilities of wireless devices to build a temporal-spatial view of the affected region.

However, there are a number of open research issues still left to investigate for further improving disaster management. As mentioned earlier, in real-disaster scenarios the network may be composed of both mobile and static users. This work provides a mechanism for the mobile users to discover other devices in the network and collect data from them. Naturally, before the data is offloaded it needs to be decided whether the user is capable of delivering data to the ECC with minimum delay. This is a major challenge as the movement of users in post-disaster scenarios

can not always be predicted. Having collected the data, solutions need to be designed to optimize the quality of relief and recovery operations based on it. The data may contain environmental and infrastructural information about the affected region as well as social network information of the affected people. How do we effectively utilize such diverse information to improve the quality of relief and recovery operations is still an open challenge.

BIBLIOGRAPHY

[1] Avl 9066 - vsat flyaway solutions. [Online]. Available: http://www.groundcontrol.com/AVL/AVL_9066_SNG_Brochure_Specifications.pdf [Accessed: 11th November, 2014]

[2] "B.a.t.m.a.n. - better approach to mobile ad-hoc networking." [Online]. Available: http://www.open-mesh.org/projects/open-mesh/wiki [Accessed: 11th November, 2014]

[3] "Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html [Accessed: 11th November, 2014]

[4] "disasterready.org, a cornerstone ondemand foundation initiative." [Online]. Available: www.disasterready.org [Accessed: 11th November, 2014]

[5] emule project. [Online]. Available: http://www.emule-project.net/ [Accessed: 11th November, 2014]

[6] "International federation of red cross and red crescent societies." [Online]. Available: http://www.ifrc.org/en [Accessed: 11th November, 2014]

[7] "Network simulator (ns2)," http://isi.edu/nsnam/ns/.

[8] "Powertutor." [Online]. Available: http://powertutor.org/index.html [Accessed: 11th November, 2014]

[9] "Surfbeam 2 broadband system," ViaSat Inc. [Online]. Available: http://www.viasat.com/files/assets/surfbeam2_Overview_018_web.pdf [Accessed: 11th November, 2014]

[10] D. E. M. Agency, "Ready, prepare, plan and stay informed." [Online]. Available: http://www.ready.gov [Accessed: 11th November, 2014]

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[12] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257 – 279, 2005.

[13] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[14] W.-F. Alliance, "Wi-fi direct." [Online]. Available: http://www.wi-fi.org/discover-and-learn/wi-fi-direct [Accessed: 11th November, 2014]

[15] M. Allman, K. Christensen, B. Nordman, and V. Paxson, "Enabling an energy-efficient future internet through selectively connected end systems," in *6th Workshop on Hot Topics in Networks*, 2007, pp. 1–6.

[16] E. Alotaibi and B. Mukherjee, "Survey paper: A survey on routing algorithms for wireless ad-hoc and mesh networks," *Comput. Netw.*, vol. 56, no. 2, pp. 940–965, Feb. 2012.

[17] G. Anastasi, M. Conti, I. Giannetti, and A. Passarella, "Design and evaluation of a bittorrent proxy for energy saving," in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, 2009, pp. 116 –121.

[18] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537 – 568, 2009.

[19] G. Anastasi, I. Giannetti, and A. Passarella, "A bittorrent proxy for green internet file sharing: Design and experimental evaluation," *Computer Communications*, vol. 33, no. 7, pp. 794 – 802, 2010.

[20] K. R. Apt and A. Witzel, "A generic approach to coalition formation," *Int'l Game Theory Review*, vol. 11, no. 03, pp. 347–367, 2009.

[21] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, vol. 46, no. 5, pp. 605–634, Dec. 2004.

[22] N. Aziz and K. Aziz, "Managing disaster with wireless sensor networks," in *Proc. of ICACT*, 2011, pp. 202–207.

[23] F. Bai and A. Helmy, "A survey of mobility models," *Wireless Adhoc Networks. University of Southern California, USA*, vol. 206, 2004.

[24] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 185–196.

[25] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Commun. ACM*, vol. 46, no. 2, pp. 43–48, Feb. 2003.

[26] L. Bengtsson, X. Lu, A. Thorson, R. Garfield, and J. von Schreeb, "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: A post-earthquake geospatial study in haiti," *PLoS medicine*, vol. 8, no. 8, 2011.

[27] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wirel. Netw.*, vol. 10, no. 5, pp. 555–567, Sep. 2004.

[28] J. Blackburn and K. Christensen, "A simulation study of a new green bittorrent," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, 2009, pp. 1 –6.

[29] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '98. New York, NY, USA: ACM, 1998, pp. 85–97.

[30] F. Busching, S. Schildt, and L. Wolf, "Droidcluster: Towards smartphone cluster computing – the streets are paved with potential computer clusters," in *Proc. of the 2012 32Nd Int'l Conf. on Distributed Computing Systems Workshops*, 2012, pp. 114–117.

[31] M. Castillo-Effer, D. Quintela, W. Moreno, R. Jordan, and W. Westhoff, "Wireless sensor networks for flash-flood alerting," in *Proc. of the 5th IEEE Int'l Caracas Conference on Devices, Circuits and Systems*, vol. 1, 2004, pp. 142–146.

[32] S.-H. Cheong, K.-I. Lee, Y.-W. Si, and L. H. U, "Lifeline: Emergency ad hoc network," in *Proc. of the 7th Int'l Conf. on Computational Intelligence and Security*, 2011, pp. 283–289.

[33] C.-Y. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug 2003.

[34] B. Cohen, "Incentives build robustness in bittorrent," in *Proc. First Workshop on Economics of Peer-to-peer Systems*, Berkely, 2003.

[35] D. P. Coppola, *Introduction to international disaster management.* Butterworth-Heinemann, 2006.

[36] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[37] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks*, ser. Q2SWinet '05. New York, NY, USA: ACM, 2005, pp. 16–23.

[38] G. Ding and B. Bhargava, "Peer-to-peer file-sharing over mobile ad hoc networks," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, ser. PERCOMW '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 104–.

[39] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," in *Smart Structures and Materials*. International Society for Optics and Photonics, 2005, pp. 477–484.

[40] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, May 2005.

[41] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 71–84.

[42] H. T. Eger, Kolja, A. Binzenhöfer, and G. Kunzmann, "Efficient simulation of large-scale p2p networks: packet-level vs. flow-level simulations," in *Proc. of the second workshop on Use of P2P, GRID and agents for the development of content networks*, 2007, pp. 9–16.

[43] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*, ser. SOSP '99. New York, NY, USA: ACM, 1999, pp. 48–63.

[44] V. Galluzzi and T. Herman, "Survey: Discovery in wireless sensor networks," *International Journal of Distributed Sensor Networks,*, vol. 2012, 2012.

[45] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.

[46] P. W. Gething and A. J. Tatem, "Can mobile phone data improve emergency response to natural disasters?" *PLoS Medicine*, vol. 8, 2011.

[47] M. Gielen, "Ad hoc networking using wi-fi during natural disasters: Overview and improvements," in *17th Twente Student Conference on IT*, vol. 17, June 2012.

[48] E. Gilbert, B. Kaliaperumal, and E. B. Rajsingh, "Research issues in wireless sensor network applications: a survey," *International Journal of information and electronics engineering*, vol. 2, no. 5, pp. 702–706, 2012.

[49] J. J. Gonzalez, O.-C. Granmo, B. E. Munkvold, F. Y. Li, J. Dugdale *et al.*, "Multidisciplinary challenges in an integrated emergency management approach," in *ISCRAM 2012-9th International Conference on Information Systems for Crisis Response and Management*, 2012.

[50] N. Gopalakrishnan Nair, P. Morrow, and G. Parr, "Design considerations for a self-managed wireless sensor cloud for emergency response scenario," in *Proc. of 12th Annual PostGraduate Symposium on the Convergence of Telecomm., Netw. and Broadcasting*, 2011.

[51] D. Guha-Sapir, P. Hoyois, and R. Below, "Annual disaster statistical review 2013: The numbers and trends," August 2014.

[52] S. Gurun, P. Nagpurkar, and B. Y. Zhao, "Energy consumption and conservation in mobile peer-to-peer systems," in *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, 2006, pp. 18–23.

[53] H. Han, Y. Liu, G. Shen, Y. Zhang, and Q. Li, "Dozyap: Power-efficient wi-fi tethering," in *Proc. of the 10th Int'l Conf. on Mobile Systems, Applications, and Services*, 2012, pp. 421–434.

[54] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "Firewxnet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '06. New York, NY, USA: ACM, 2006, pp. 28–41.

[55] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 174–185.

[56] T. Herman, S. Pemmaraju, L. Pilard, and M. Mjelde, "Temporal partition in sensor networks," in *Stabilization, Safety, and Security of Distributed Systems*. Springer, 2007, pp. 325–339.

[57] V. Hristidis, S.-C. Chen, T. Li, S. Luis, and Y. Deng, "Survey of data management and analysis in disaster situations," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1701–1714, Oct. 2010.

[58] M. Jimeno and K. Christensen, "A prototype power management proxy for gnutella peer-to-peer file sharing," in *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, 2007, pp. 210–212.

[59] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, June 2005, pp. 599 –608.

[60] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 350–361.

[61] K. Kant, "Supply and demand coordination in energy adaptive computing," in *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2010, pp. 1–6.

[62] ——, "Distributed energy adaptive computing," in *IEEE International Conference on Communications (ICC)*, May 2010, pp. 1–5.

[63] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 23, pp. 293 – 315, 2003, sensor Network Protocols and Applications.

[64] O. Kassinen, E. Harjula, J. Korhonen, and M. Ylianttila, "Battery life of mobile peers with umts and wlan in a kademlia-based p2p overlay," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, 2009, pp. 662 –665.

[65] H. Kazerooni, "The berkeley lower extremity exoskeleton project," in *Experimental Robotics IX*, ser. Springer Tracts in Advanced Robotics, 2006, vol. 21, pp. 291–301.

[66] I. Kel andnyi, A. Ludanyi, and J. Nurminen, "Bittorrent on mobile phones - energy efficiency of a distributed proxy solution," in *Green Computing Conference, 2010 International*, 2010, pp. 451 –458.

[67] I. Kelényi and J. K. Nurminen, "Cloudtorrent - energy-efficient bittorrent content sharing for mobile devices via cloud services," in *Proceedings of the 7th IEEE conference on Consumer communications and networking conference*, 2010, pp. 646–647.

[68] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Springer-Verlag, 1976.

[69] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 254–263.

[70] A. Klemm, C. Lindemann, and O. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *IEEE 58th Vehicular Technology Conference, 2003*, vol. 4, Oct 2003, pp. 2758–2763 Vol.4.

[71] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and T. D. Hämäläinen, "Energy-efficient neighbor discovery protocol for mobile wireless sensor networks," *Ad Hoc Netw.*, vol. 7, no. 1, pp. 24–41, Jan. 2009.

[72] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wirel. Netw.*, vol. 6, no. 4, pp. 263–277, Jul. 2000.

[73] P. Krishnamoorthy and M. Wright, "Towards modeling the behavior of physical intruders in a region monitored by a wireless sensor network," in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, ser. AISec '10. New York, NY, USA: ACM, 2010, pp. 1–7.

[74] F. Legendre, T. Hossmann, F. Sutton, and B. Plattner, "30 years of wireless ad hoc networking research: What about humanitarian and disaster relief solutions? what are we still missing?" in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief (ACWR)*, 2011, pp. 217–217.

[75] J. Li, W. Jia, L. Huang, M. Xiao, and J. Wang, "An efficient source peer selection algorithm in hybrid p2p file sharing systems," in *Proceedings of the 7th international conference on Algorithms and architectures for parallel processing*, 2007, pp. 380–390.

[76] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, "Privacy preservation in wireless sensor networks: A state-of-the-art survey," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1501–1514, 2009.

[77] N. Li, M. Raj, D. Liu, M. Wright, and S. K. Das, "Using data mules to preserve source location privacy in wireless sensor networks," in *Proceedings of the 13th International Conference on Distributed Computing and Networking*, ser. ICDCN'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 309–324. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25959-3_23

[78] Y. Li and J. Ren, "Mixing ring-based source-location privacy in wireless sensor networks," in *Proceedings of ICCCN*, August 2009, pp. 1–6.

[79] ——, "Source-location privacy through dynamic routing in wireless sensor networks," in *Proceedings of INFOCOM*, 2010.

[80] A. Loewenstern. (2008) Bittorrent dht protocol. [Online]. Available: http://www.bittorrent.org/beps/bep0005.html [Accessed: 11th November, 2014]

[81] W. Lu, W. K. G. Seah, E. W. C. Peh, and Y. Ge, "Communications support for disaster recovery operations using hybrid mobile ad-hoc networks," in *Proc. of LCN*, 2007, pp. 763–770.

[82] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 88–97.

[83] P. Maymounkov and D. Mazires, "Kademlia: A peer-to-peer information system based on the xor metric," 2002, vol. 2429, pp. 53–65.

[84] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 137–145.

[85] M. Mecella, M. Angelaccio, A. Krek, T. Catarci, B. Buttarazzi, and S. Dustdar, "Workpad: An adaptive peer-to-peer software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios," in *Proceedings of the International Symposium on Collaborative Technologies and Systems*, ser. CTS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 173–180.

[86] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proceedings of ICNP*, October 2007, pp. 314–323.

[87] A. Meissner, T. Luckenbach, T. Risse, T. Kirste, and H. Kirchner, "Design challenges for an integrated disaster management communication and information system," in *The First IEEE Workshop on Disaster Recovery Networks (DIREN 2002)*, vol. 24, 2002.

[88] G. o. I. Ministry of Home Affairs, National Disaster Management Division, "Ict for disaster risk reduction the indian experience," 2005. [Online]. Available: http://www.ndmindia.nic.in/wcdrdocs/ictfordisasterriskreduction.pdf [Accessed: 11th November, 2014]

[89] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *Wireless Commun.*, vol. 11, no. 1, pp. 48–60, Feb. 2004.

[90] P. Mohapatra, C. Gui, and J. Li, "Group communications in mobile ad hoc networks," *Computer*, vol. 37, no. 2, pp. 52–59, Feb. 2004.

[91] J. Nurminen and J. Noyranen, "Energy-consumption in mobile peer-to-peer - quantitative results from file sharing," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, 2008, pp. 729 –733.

[92] I. F. of Red Cross and R. C. Societies, "World disaster report," 2013. [Online]. Available: http://worlddisastersreport.org/en/ [Accessed: 11th November, 2014]

[93] L. B. Oliveira, I. G. Siqueira, D. F. Macedo, A. A. F. Loureiro, H. C. Wong, and J. M. Nogueira, "Evaluation of peer-to-peer network content discovery techniques over mobile ad hoc networks," in *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*. IEEE, 2005, pp. 51–56.

[94] L. M. Oliveira and J. J. Rodrigues, "Wireless sensor networks: a survey on environmental monitoring," *Journal of communications*, vol. 6, no. 2, pp. 143–151, 2011.

[95] Y. Ouyang, X. Le, G. Chen, J. Ford, and F. Makedon, "Entrapping adversaries for source protection in sensor networks," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006.

[96] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings. 2nd ACM workshop on Security of ad hoc and sensor networks, SASN*, 2004.

[97] D. Panigrahi, S. Dey, R. Rao, K. Lahiri, C. Chiasserini, and A. Raghunathan, "Battery life estimation of mobile embedded systems," in *VLSID '01: Proceedings of the The 14th International Conference on VLSI Design (VLSID '01)*. Washington, DC, USA: IEEE Computer Society, 2001, p. 57.

[98] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[99] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[100] H. Peters, *Game Theory - A multi-level Approach*. Springer, 2008.

[101] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Wireless Sensor Networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 2004, pp. 399–423.

[102] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," in *Peer-to-Peer Systems IV*. Springer, 2005, pp. 205–216.

[103] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 367–378.

[104] M. Raj, K. Kant, and S. Das, "E-darwin: Energy aware disaster recovery network using wifi tethering," in *23rd International Conference on Computer Communication and Networks (ICCCN)*, Aug 2014, pp. 1–8.

[105] M. Raj, K. Kant, and S. K. Das, "Energy adaptive mechanism for p2p file sharing protocols," in *Proceedings of the 18th International Conference on Parallel Processing Workshops*, ser. Euro-Par'12. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 89–99.

[106] M. Raj, N. Li, D. Liu, M. Wright, and S. K. Das, "Using data mules to preserve source location privacy in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 11, no. 0, pp. 244 – 260, 2014.

[107] N. K. Ray and A. K. Turuk, "A framework for disaster management using wireless ad hoc networks," in *Proc. of CCS*, 2011, pp. 138–141.

[108] C. Reinhardt, "Taxi cab geometry: History and applications!" *The Montana Mathematics Enthusiast*, vol. 2, no. 1, pp. 38–65, April 2005.

[109] S. M. Ross, *Introductory Statistics*. Academic Press Title, 2010.

[110] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "Coalition formation games for collaborative spectrum sensing," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 276–297, Jan 2011.

[111] M. Salvatore et al., "Mapping global urban and rural population distributions," *Food and Agriculture Organization of the United Nations*, 2005.

[112] J. P. Scott and P. J. Carrington, *The SAGE Handbook of Social Network Analysis*. Sage Publications Ltd., 2011.

[113] C. R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[114] W. Shan, J. Feng, J. Chang, F. Yang, and Z. Li, "Collecting earthquake disaster area information using smart phone," in *IEEE Int'l Conf. on System Science and Engineering (ICSSE)*, 2012, pp. 310–314.

[115] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding, "Cool-tether: Energy efficient on-the-fly wifi hot-spots using mobile phones," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 109–120.

[116] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '02. New York, NY, USA: ACM, 2002, pp. 160–171.

[117] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures," in *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42. New York, NY, USA: ACM, 2009, pp. 168–178.

[118] N. Soreide, C. Woody, and S. Holt, "Overview of ocean based buoys and drifters: Present applications and future needs," in *16th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, vol. 4. IEEE, 2001, pp. 2470–2472.

[119] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 34–40, Jun. 2004.

[120] R. Tan, G. Xing, J. Chen, W.-Z. Song, and R. Huang, "Quality-driven volcanic earthquake detection using wireless sensor networks," in *Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*, ser. RTSS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 271–280.

[121] J. Thomas, J. Robble, and N. Modly, "Off grid communications with android meshing the mobile world," in *IEEE Conf. on Technologies for Homeland Security (HST)*, November 2012, pp. 401 –405.

[122] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "Wifi-opp: Ad-hocless opportunistic networking," in *Proc. of the 6th ACM Workshop on Challenged Networks*, 2011, pp. 37–42.

[123] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel, "Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching," in *Proc. of the 8th ACM MobiCom Workshop on Challenged Networks*, 2013, pp. 31–36.

[124] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," *Comput. Netw.*, vol. 43, no. 3, pp. 317–337, Oct. 2003.

[125] M. Y. S. Uddin, D. M. Nicol, T. F. Abdelzaher, and R. H. Kravets, "A post-disaster mobility model for delay tolerant networking," in *Winter Simulation Conference*, ser. WSC '09. Winter Simulation Conference, 2009, pp. 2785–2796.

[126] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, pp. 245–259, June 2004.

[127] Y. Wang, X. Wang, B. Xie, D. Wang, and D. P. Agrawal, "Intrusion detection in homogeneous and heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 698–711, Jun. 2008.

[128] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, Mar. 2006.

[129] H. Wirtz, T. Heer, R. Backhaus, and K. Wehrle, "Establishing mobile ad-hoc networks in 802.11 infrastructure mode," in *Proc. of the 6th ACM Workshop on Challenged Networks*, 2011, pp. 49–52.

[130] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 13–24.

[131] C. Yang, J. Yang, X. Luo, and P. Gong, "Use of mobile phones in an emergency reporting system for infectious disease surveillance after the Sichuan earthquake in China," *Bulletin of the World Health Organization*, vol. 87, pp. 619 – 623, 2009.

[132] W. Yang and W. Zhu, "Source location privacy in wireless sensor networks with data aggregation," in *Proceedings of UIC*, August 2010, pp. 1–6.

[133] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proceedings of WiSec*, 2008.

[134] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Tenth International Conference on Computer Communications and Networks*. IEEE, 2001, pp. 304–309.

[135] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[136] B. Zhang, A. Iosop, J. Pouwelse, and P. Garbacki, "The peer-to-peer trace archive: Design and comparative trace analysis," Delft University of Technology, Tech. Rep. PDS-2010-003,, 2010.

[137] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, pp. 61–72, March 2002.

[138] G. Zussman and A. Segall, "Energy efficient routing in ad hoc disaster recovery networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 405 – 421, 2003.

VITA

Mayank Raj was born in Patna, India in 1982. He earned his Bachelor's degree in Electronics and Communication Engineering from Dayananda Sagar College of Engineering at Bangalore in 2005. He completed is Master's degree from International Institute of Information Technology, Bangalore in 2007. After his graduation, he worked as a Research Analyst in Applied Research Group at Satyam Computing Services Ltd. and then as a Research Fellow at International Institute of Information Technology, Bangalore. Mayank joined University of Texas at Arlington as a PhD candidate in 2009. In 2013, he transferred to Missouri University fo Science and Technology where he earned his Doctorate of Philosophy in Computer Science, in December, 2014. As a PhD student, Mayank worked under Prof. Sajal K. Das , focusing on energy adapting computing, disaster management, bio-inspired networking, and cloud computing.